Jun Wang Zhang Yi
Jacek M. Zurada Bao-Liang Lu
Hujun Yin (Eds.)

Advances in Neural Networks – ISNN 2006

LNCS 3973

Third International Symposium on Neural Networks
Chengdu, China, May/June 2006
Proceedings, Part III

Springer

Lecture Notes in Computer Science 3973

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

Editorial Board

Jun Wang Zhang Yi Jacek M. Zurada
Bao-Liang Lu Hujun Yin (Eds.)

Advances in Neural Networks – ISNN 2006

Third International Symposium on Neural Networks
Chengdu, China, May 28 - June 1, 2006
Proceedings, Part III

Springer

Volume Editors

Jun Wang
The Chinese University of Hong Kong
Dept. of Automation and Computer-Aided Engineering
Shatin, New Territories, Hong Kong
E-mail: jwang@acae.cuhk.edu.hk

Zhang Yi
University of Electronic Science and Technology of China
School of Computer Science and Engineering
Chengdu, Sichuan, China
E-mail: zhangyi@uestc.edu.cn

Jacek M. Zurada
University of Louisville, Dept. of Electrical and Computer Engineering
Louisville, Kentucky, USA
E-mail: jacek.zurada@louisville.edu

Bao-Liang Lu
Shanghai Jiao Tong University, Dept. of Computer Science and Engineering
Shanghai, China
E-mail: blu@cs.sjtu.edu.cn

Hujun Yin
University of Manchester, School of Electrical and Electronic Engineering
Manchester M60 IQD, UK
E-mail: h.yin@manchester.ac.uk

Library of Congress Control Number: 2006925897

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

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

ISSN 0302-9743
ISBN-10 3-540-34482-9 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-34482-7 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11760191 06/3142 5 4 3 2 1 0

Preface

This book and its sister volumes constitute the Proceedings of the Third International Symposium on Neural Networks (ISNN 2006) held in Chengdu in southwestern China during May 28–31, 2006. After a successful ISNN 2004 in Dalian and ISNN 2005 in Chongqing, ISNN became a well-established series of conferences on neural computation in the region with growing popularity and improving quality. ISNN 2006 received 2472 submissions from authors in 43 countries and regions (mainland China, Hong Kong, Macao, Taiwan, South Korea, Japan, Singapore, Thailand, Malaysia, India, Pakistan, Iran, Qatar, Turkey, Greece, Romania, Lithuania, Slovakia, Poland, Finland, Norway, Sweden, Demark, Germany, France, Spain, Portugal, Belgium, Netherlands, UK, Ireland, Canada, USA, Mexico, Cuba, Venezuela, Brazil, Chile, Australia, New Zealand, South Africa, Nigeria, and Tunisia) across six continents (Asia, Europe, North America, South America, Africa, and Oceania). Based on rigorous reviews, 616 high-quality papers were selected for publication in the proceedings with the acceptance rate being less than 25%. The papers are organized in 27 cohesive sections covering all major topics of neural network research and development. In addition to the numerous contributed papers, ten distinguished scholars gave plenary speeches (Robert J. Marks II, Erkki Oja, Marios M. Polycarpou, Donald C. Wunsch II, Zongben Xu, and Bo Zhang) and tutorials (Walter J. Freeman, Derong Liu, Paul J. Werbos, and Jacek M. Zurada). ISNN 2006 provided an academic forum for the participants to disseminate their new research findings and discuss emerging areas of research. It also created a stimulating environment for the participants to interact and exchange information on future challenges and opportunities of neural network research.

Many volunteers and organizations made great contributions to ISNN 2006. The organizers are grateful to the University of Electronic Science and Technology of China and the Chinese University of Hong Kong for their sponsorship; to the National Natural Science Foundation of China and K.C. Wong Education Foundation of Hong Kong for their financial supports; and to the Asia Pacific Neural Network Assembly, European Neural Network Society, IEEE Computational Intelligence Society, IEEE Circuits and Systems Society, and International Neural Network Society for their technical cosponsorship. The organizers would like to thank the members of the Advisory Committee for their supports, the members of the International Program Committee for reviewing the papers and members of the Publications Committee for checking the accepted papers in a short period of time. Particularly, the organizers would like to thank the publisher, Springer, for publishing the proceedings in the prestigious series of

Lecture Notes in Computer Science. Last but not least, the organizers would like to thank all the speakers and authors for their active participation at ISNN 2006, which is essential for the success of the symposium.

May 2006

Jun Wang
Zhang Yi
Jacek M. Zurada
Bao-Liang Lu
Hujun Yin

ISNN 2006 Organization

ISNN 2006 was organized and sponsored by the University of Electronic Science and Technology of China and the Chinese University of Hong Kong. It was technically cosponsored by the Asia Pacific Neural Network Assembly, European Neural Network Society, IEEE Circuits and Systems Society, IEEE Computational Intelligence Society, and International Neural Network Society. It was financially supported by the National Natural Science Foundation of China and K.C. Wong Education Foundation of Hong Kong.

Jun Wang, Hong Kong (General Chair)
Zhang Yi, Chengdu, China (General Co-chair)
Jacek M. Zurada, Louisville, USA (General Co-chair)

Advisory Committee

Shun-ichi Amari, Tokyo, Japan (Chair)
Hojjat Adeli, Columbus, USA
Guoliang Chen, Hefei, China
Chunbo Feng, Nanjing, China
Kunihiko Fukushima, Tokyo, Japan
Okyay Kaynak, Istanbul, Turkey
Yanda Li, Beijing, China
Erkki Oja, Helsinki, Finland
Marios M. Polycarpou, Nicosia, Cyprus
Shoujue Wang, Beijing, China
Youlun Xiong, Wuhan, China
Shuzi Yang, Wuhan, China
Siying Zhang, Qingdao, China
Shoubin Zou, Chengdu, China
Walter J. Freeman, Berkeley, USA (Co-chair)
Zheng Bao, X'ian, China
Ruwei Dai, Beijing, China
Toshio Fukuda, Nagoya, Japan
Zhenya He, Nanjing, China
Frank L. Lewis, Fort Worth, USA
Ruqian Lu, Beijing, China
Nikhil R. Pal, Calcutta, India
Tzyh-Jong Tarn, St. Louis, USA
Paul J. Werbos, Washington, USA
Lei Xu, Hong Kong
Bo Zhang, Beijing, China
Nanning Zheng, Xi'an, China

Steering Committee

Zongben Xu, Xi'an, China (Chair)
Tianping Chen, Shanghai, China
Wlodzislaw Duch, Torun, Poland
Xiaoxin Liao, Wuhan, China
Zhiyong Liu, Beijing, China
Zhengqi Sun, Beijing, China
Donald C. Wunsch II, Rolla, USA
Fuliang Yin, Dalian, China
Liming Zhang, Shanghai, Chinac
Mingtian Zhou, Chengdu, China
Houjun Wang, Chengdu, China (Co-chair)
Andrzej Cichocki, Tokyo, Japan
Anthony Kuh, Honolulu, USA
Derong Liu, Chicago, USA
Leszek Rutkowski, Czestochowa, Poland
DeLiang Wang, Columbus, USA
Gary G. Yen, Stillwater, USA
Juebang Yu, Chengdu, China
Chunguang Zhou, Changchun, China

Program Committee

Bao-Liang Lu, Shanghai, China (Chair)
Hujun Yin, Manchester, UK (Co-chair)
Shigeo Abe, Kobe, Japan
Ajith Abraham, Seoul, South Korea
Khurshid Ahmad, Surrey, UK
Sabri Arik, Istanbul, Turkey
A. Bouzerdoum, Wollongong, Australia
Jianting Cao, Saitama, Japan
Jinde Cao, Nanjing, China
Wenming Cao, Hangzhou, China
Matthew Casey, Surrey, UK
Liang Chen, Prince George, Canada
Luonan Chen, Osaka, Japan
Songcan Chen, Nanjing, China
Yen-Wei Chen, Kyoto, Japan
Xue-wen Chen, Kansas, USA
Yuehui Chen, Jinan, China
Xiaochun Cheng, Berkshire, UK
Yiu Ming Cheung, Hong Kong
Zheru Chi, Hong Kong
Sungzoon Cho, Seoul, Korea
Jin-Young Choi, Seoul, Korea
Emilio Corchado, Burgos, Spain
Chuanyin Dang, Hong Kong
Shuxue Ding, Fukushima, Japan
Tom Downs, Brisbane, Australia
Meng Joo Er, Singapore
Shumin Fei, Nanjing, China
Mauro Forti, Siena, Italy
Wai Keung Fung, Winnipeg, Canada
Marcus Gallagher, Brisbane, Australia
John Qiang Gan, Essex, UK
Chengling Gou, Beijing, China
Chengan Guo, Dalian, China
Lei Guo, Nanjing, China
Ping Guo, Beijing, China
Min Han, Dalian, China
Qing-Long Han, Rockhampton, Australia
Zhifeng Hao, Guangzhou, China
Daniel W.C. Ho, Hong Kong
Zengguang Hou, Beijing, China
Dewen Hu, Changsha, China
Jinglu Hu, Fukuoka, Japan
Sanqing Hu, Chicago, USA
Guangbin Huang, Singapore
Shunan Huang, Singapore
Marc van Hulle, Leuven, Belgium
Malik Magdon Ismail, Troy, USA
Danchi Jiang, Hobart, Australia
Joarder Kamruzzaman, Melbourne, Australia
Hoon Kang, Seoul, Korea
Samuel Kaski, Helsinki, Finland
Nikola Kasabov, Auckland, New Zealand
Tai-hoon Kim, Seoul, Korea
Tae Seon Kim, Seoul, Korea
Yean-Der Kuan, Taipei, Taiwan
Hon Keung Kwan, Windsor, Canada
James Lam, Hong Kong
James Kwok, Hong Kong
Xiaoli Li, Birmingham, UK
Shaowen Li, Chengdu, China
Yuanqing Li, Singapore
Yangmin Li, Macauo
Xun Liang, Beijing, China
Hualou Liang, Houston, USA
Lizhi Liao, Hong Kong
Yanchun Liang, Changchun, China
Fei Liu, Wuxi, China
Meng-Hiot Lim, Singapore
Ju Liu, Jinan, China
Guoping Liu, Treforest, UK
Hongtao Lu, Shanghai, China
Meiqin Liu, Hangzhou, China
Fa-Long Luo, San Jose, USA
Wenlian Lu, Leipzig, Germany
Jinwen Ma, Beijing, China
Zhiwei Luo, Nagoya, Japan
Stanislaw Osowski, Warsaw, Poland
Qing Ma, Kyoto, Japan
Ikuko Nishkawa, Kyoto, Japan
Zhiqing Meng, Hangzhou, China
Paul S. Pang, Auckland, New Zealand
Seiichi Ozawa, Kobe, Japan
Yi Shen, Wuhan, China
Jagath C. Rajapakse, Singapore
Michael Small, Hong Kong
Daming Shi, Singapore
Ponnuthurai N. Suganthan, Singapore
Jochen J. Steil, Bielefeld, Germany
Fuchun Sun, Beijing, China
Changyin Sun, Nanjing, China

Norikazu Takahashi, Fukuoka, Japan
Yu Tang, Mexico City, Mexico
Christos Tjortjis, Manchester, UK
Michel Verleysen, Louwain, Belgium
Dan Wang, Singapore
Si Wu, Brighton, UK
Cheng Xiang, Singapore
Simon X. Yang, Guelph, Canada
Yingjie Yang, Leicester, UK
Dingli Yu, Liverpool, UK
Gerson Zaverucha, Rio de Janeiro, Brazil
Huaguang Zhang, Shenyang, China
Liqing Zhang, Shanghai, China
Tao Zhang, Tianjin, China
Yanqing Zhang, Atlanta, USA
Jin Zhou, Shanghai, China
Pu Sun, Ann Arbor, USA
Ying Tan, Hefei, China
Peter Tino, Birmingham, UK
Dan Ventura, Provo, USA
Bing Wang, Hull, UK
Kesheng Wang, Trondheim, Norway
Wei Wu, Dalian, China
Daoyi Xu, Chengdu, China
Xiaosong Yang, Wuhan, China
Zhengrong Yang, Exeter, UK
Wen Yu, Mexico City, Mexico
Zhigang Zeng, Hefei, China
Jie Zhang, Newcastle, UK
Qingfu Zhang, Essex, UK
Ya Zhang, Kansas, USA
Yunong Zhang, Maynooth, Ireland

Organizing Committee

Yue Wu (Chair), Chengdu, China
Xiaofeng Liao (Co-chair), Chongqing, China

Publications Committee

Chuandong Li (Chair), Chongqing, China
Jianwei Zhang (Co-chair), Hamburg, Germany
Mao Ye (Co-chair), Chengdu, China

Publicity Committee

Bin Jiang (Chair), Chengdu, China
Zeng-Guang Hou (Co-chair), Beijing, China
Jennie Si (Co-chair), Tempe, USA

Registration Committee

Xiaorong Pu (Chair), Chengdu, China

Local Arrangements Committee

Hongli Zhang (Chair), Chengdu, China

Secretariats

Jiancheng Lv, Chengdu, China
Tao Xiang, Chongqing, China

Table of Contents – Part III

Transportation Systems

Communication Networks

Information Security

Fault Detection

Financial Analysis

Neuroinformatics

Bioinformatics

Biomedical Applications

Industrial Applications

Other Applications

Hardware Implementation

Traffic Volume Forecasting Based on Wavelet Transform and Neural Networks

Shuyan Chen[1,2] and Wei Wang[2]

[1] College of Transportation, Southeast University,
210096 Nanjing, China
wangwei@seu.edu.cn
[2]Department of Electronic information, Nanjing Normal University,
210097 Nanjing, China
chenshuyan@njnu.edu.cn

Abstract. This paper focuses on traffic volume forecasting that is an essential component of any responsive traffic control or route guidance system. A new approach for traffic volume prediction is proposed based on wavelet transform and neural networks. First, apply multi-resolution analysis to the original traffic volume time series to obtain a trend series and a hierarchy of detail series. Then apply neural networks to each obtained time series. Next sum all the forecasting values to get the final prediction of traffic volume**.** This hybrid method is implemented within a Matlab environment. The feasibility of the developed method as well as its validity to predict traffic volume has been demonstrated on real data gathered in Suzhou city. Moreover, a comparison between the hybrid method and conventional neural networks is conducted. The results show the proposed hybrid model outperformed the neural networks.

1 Introduction

Intelligent Transportation System (ITS) is a key technique to solve many traffic problems present on many cities. With the development of ITS, traffic volume forecasting serve for traffic control and vehicle route guidance has attracted the wavelet techniques. The use of wavelet transform (WT) as a new tool for time series analysis in traffic engineering domain has been recently reported [1-4], where wavelet techniques were utilized to de-noise traffic data or to convert the original traffic data into an approximation signal and several detail signals which are easy to model.

In this paper, we discuss a new hybrid approach based on wavelet transform and the back propagation neural networks (BPNNs) for traffic volume forecasting. The main motivation of using wavelet techniques is because the obtained time series are easy to model and predict. The reason to employ BPNNs is that it can learn a mapping of any complexity, which is its most important attribute. First, apply wavelet decomposition and reconstruction to manipulate the original traffic volume time series. Second, train neural networks to predict the reconstructed signals. Last, add up all the forecasting values to get the wanted forecasting results. From our experiment, it can be concluded that this new approach yields much better results, and it can enhance

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1 – 7, 2006.

prediction accuracy, and thus, wavelet analysis has shown potential in traffic volume forecasting.

The rest of the paper is organized as follows. The next section describes the principle of wavelet analysis. Section 3 provides a description of BPNNs. Section 4 describes the algorithm of our hybrid approach. In Section 5, the experimental results are presented to demonstrate the effectiveness of this method. Finally we summarize our contributions and mention some of the directions the authors are considering investigating.

2 Introduction of Wavelet Analysis

Wavelet analysis is a transformation method that transforms the original signal (usually in the time domain) into a different domain for analysis and processing [1-5]. If the signal analyzed is non-stationary, i.e. mean and autocorrelation of the signal are not constant over time, there are some serious restrictions using Fourier transform (FT). As we know, most of the time series of traffic data are non- stationary, this is why we applied discrete wavelet transform (DWT) to the traffic volume.

By Fourier transformation, the original signal is decomposed into a linear combination of sine and cosine harmonics whereas by discrete wavelet transform in wavelet multi-resolution, a signal is represented by a sum of a more flexible function called wavelet that is localized in both time and frequency. Suppose x(t) is the time signal, it can be expressed as follows [6],

$$x(t)=\sum_k c_{j,k}\varphi_{j,k}(t)+\sum_k d_{j,k}\psi_{j,k}(t)+\sum_k d_{j-1,k}\psi_{j-1,k}(t)+\cdots+\sum_k d_{1,k}\psi_{1,k}(t) \tag{1}$$

Where, $\varphi_{j,k}(t)$ and $\psi_{j,k}(t)$ are wavelet function, $c_{j,k}$, $d_{j,k}$, …, $d_{1,k}$ are amplitude coefficients of each basis function, j,k belong to integers set, j>0 is the resolution level, *k* is the shift parameter, k=1,2,…,N/2, N is the number of data recorded.

Theoretically, the coefficients of DWT $c_{j,k}$ and $d_{j,k}$ can be found by taking the inner products of the basis functions and the original signal as follows:

$$c_{j,k}=\left\langle x,\varphi_{j,k}\right\rangle=\sum_t x(t)\varphi_{j,k}(t) \tag{2}$$

$$d_{j,k}=\left\langle x,\psi_{j,k}\right\rangle=\sum_t x(t)\psi_{j,k}(t) \tag{3}$$

Take a reference at level j, there are two sorts of components, which we denoted by $A_j(t)$ and $D_i(t)$ (i=1,2,…,j), let

$$A_j(t)=\sum_k c_{j,k}\varphi_{j,k}(t) \tag{4}$$

$$D_i(t)=\sum_k d_{i,k}\psi_{i,k}(t) \tag{5}$$

Then, $A_j(t)$ correspond to coarser detail at level j which is called an approximation of the signal x(t), it contains the low frequency component of x(t), $D_i(t)$ (i=1,2,…,j)

correspond to the fine details. If we realize wavelet transform at levels j, the original signal can be reconstructed directly by approximation signal plus detail signals, i.e. ,

$$x(t) = A_j(t) + D_j(t) + D_{j-1}(t) + \ldots + D_1(t) . \quad (6)$$

3 Apply Neural Networks to Obtained Signals

Neural network has been considered to be an important forecasting tool used in a variety of areas, including traffic volume prediction [7, 8]. In this work, we have concentrated our attention on using BPNNs to predict every obtained signal. Although there are currently many different types of neural networks available, we choose this kinds of neural network architecture for they has the ability to implement arbitrary complex input/output mappings and have been widely used in practice.

The architecture of BPNNs normally consists of an input layer, one or more hidden layers and an output layer. The BPNNs operates in a feed-forward manner, i.e., processing signal flows only from the input to the output layers. When an input pattern is presented to the network, it will first be mapped by hidden layer weights, and then used to activate the hidden neurons. The outputs of hidden neurons serve as inputs to the next layer and repeat the same mapping and activation procedure. Training of a BPNN model is conducted by presenting a set of input-output pairs to the network, and adjusting hidden and output layer weights and biases according to the commonly used back-propagation rule. The key idea of this training method is that if the network parameters can be adjusted in the gradient descent direction of the output error surface, then the network output error can be minimized. The adjustments are made in a backward direction, where the error is first computed at the output layer and then back-propagated to the hidden layers. Users can control the magnitude of weight adjustment and the convergence speed by setting the learning and momentum rates [9, 10].

The BPNNs implemented for traffic volume forecasting consisted of one input layer, one hidden layer and one output layer. The input layer had four input neurons representing data at time t,t+1,t+2 and t+3, and one output neuron was used to indicate forecasting value at time t+4. The number of neurons in the hidden layer was set to 4. Suppose x(t) (t=1,2,...,n) is a time series, setting (x(t),x(t+1),x(t+2),x(t+3)) as the input pattern, and choosing x(t+4) as the known target, the network can be trained to predict the future data, Mathematically,

$$(x(t), x(t+1), x(t+2), x(t+3)) \rightarrow x(t+4), \mathrm{t} = 1,2,\ldots,\mathrm{n}-4 . \quad (7)$$

4 Algorithm of Our Hybrid Approach

Based on the wavelet transform, the traffic volume time series can be converted into an approximation signal and multi-scale detail signals. When detail signals are not considered, it may lose much information for prediction. Hence, it is expected that a

higher accuracy should be gained when predicting the detail signals and adding it to forecasting results directly from the approximation signal.

For the approximation signal and each detail signals, here we apply neural network model to achieve their forecasting values. Suppose the forecasting result of the detail signals is $\hat{D}_i(t)$ (i=1,2,...,j) and approximation signal is $\hat{A}_j(t)$, we would achieve the future value of the original traffic volume $\hat{x}(t)$ by summing the aforementioned predicted parts,

$$\hat{x}(t) = \hat{A}_j(t) + \hat{D}_j(t) + \hat{D}_{j-1}(t) + ... + \hat{D}_1(t). \tag{8}$$

5 An Illustrate Example with Real Traffic Data

5.1 Description of Data

To demonstrate the strength of our method, we applied this approach to forecast traffic volume. Traffic volume is the number of vehicles passing a certain position within interval. Data were gathered from an intersection in Suzhou city and were recorded in 15-minute intervals. Divide the collected traffic volume data into two sets, the first set included 288 point collected from November 27, 2002 through November 29, 2002 used as known data for modeling, and the second part included 69 data measured on November 30, 2002, during a period from 6:45 to 24:00, used as test data.

The decomposition and reconstruction of the time series has been achieved with sym4 function at level 1. Figure 1 shows the approximation signal A_1 and detail signal D_1 separated from the raw signal.

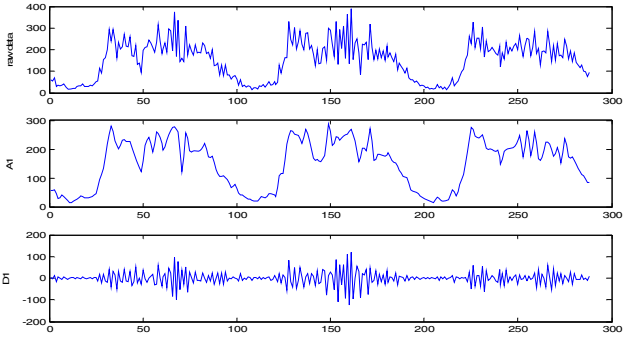


Fig. 1. The traffic volume and its approximation signal A_1 and detail signal D_1

5.2 Measures of Performance

Choose two criteria, i.e. mean absolute percentage error (MAPE) and root mean square error (RMSE) to compare the performance of the approach. MAPE and RMSE are defined as follow:

$$MAPE = \frac{1}{N}\sum_{t=1}^{N}\left|\frac{x(t)-\hat{x}(t)}{x(t)}\right| \tag{9}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(x(t)-\hat{x}(t))^2} \tag{10}$$

Where, x(t) denotes the traffic volume at time t and $\hat{x}(t)$ its prediction value, N is the number predicted values concerned.

5.3 Experiment Results

Train two BPNNs with the trend and details time series. During the training of the BPNN model, the initial weights were generated randomly. Pattern-by-pattern learning method was applied, i.e., weights were updated as soon as the output error of one training pattern was calculated, and sigmoid units were used in the hide layer and the output layer. The learning rate was set to 0.2, the momentum rate was fixed at 0.15, and the number of epochs to train through is set to 1500. The final prediction value is achieved by summing up the outputs of these two networks.

For the purpose of comparison, we trained another BPNN with the original traffic volume series and tested with the same test data. The structure of BPNN is the same as the above BPNN used in our method whereas its output represents the likely value of the predicted.

We train multiple BPNNs using the same data, but initializing each network with different random weights via setting different random seed in order to reduce biases in reporting performance measures. The comparison between our method and general BPNNs on criterion MAPE and RMSE over the test set were shown in table 1, and it concludes 5 times comparison and their averages.

Table 1. Comparison of performance between our hybrid approach and traditional BPNNs

		1	2	3	4	5	Average
Our approach	MAPE	11.78	11.45	11.35	11.57	10.71	11.37
	RMSE	27.87	29.90	30.65	29.45	28.40	29.25
Traditional BPNNs	MAPE	25.34	27.28	23.75	22.04	26.15	24.91
	RMSE	54.19	67.16	54.94	51.65	53.93	56.37

The testing results are very encouraging. As shown in Table 1, our method achieved the better MAPE of 11.37% and RMSE of 29.25 compare to normal BPNNs with the corresponding number of 24.91% and 56.37, respectively. It can be concluded that our method is superior to the traditional neural networks since it increases the prediction accuracy on the test set. It is also found that the approximation component and detail component are much easier to fit than the original traffic volume.

Figure 2 illustrated the real data in test set and its prediction value, yield by the hybrid method and general BPNNs approach with the best performance over the test data corresponding to No 4 in table 1. As this plot indicated, the hybrid method gave a better goodness of fit.

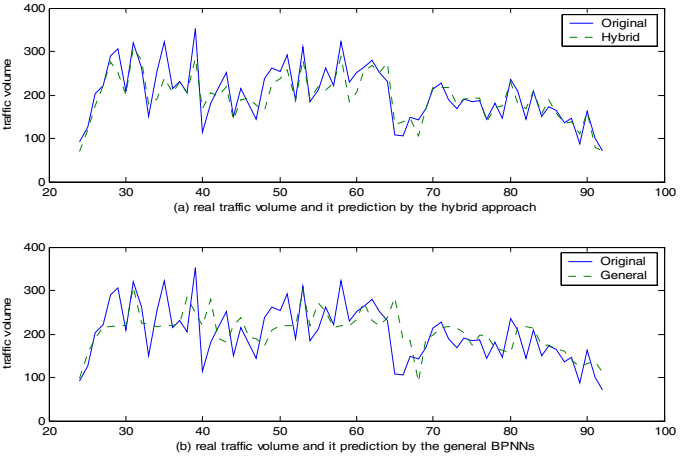


Fig. 2. The real traffic volume in test set and its prediction

6 Conclusions

The wavelet transform is a popular time frequency analysis tool in signal processing and multi-resolution analysis is a major property of wavelet transform, by which signal can be decomposed into high frequency components and low frequency components. In this study, we focused to use wavelet transform techniques to analyze traffic volume time series, and a composite methodology for traffic volume forecasting based on wavelet approach and BPNNs has been proposed. We applied neural network to approximation signal and the detail signal, then summed up the prediction of different components to get the final prediction.

The strength of the method is demonstrated with real traffic volume dataset. The real case study shows that such a hybrid method indeed provides a close match with the empirical traffic volume.

However, the proposed method suffers from two shortcomings; one is that its performance depends heavily on the selection of wavelet function. In our test, we alleviate this problem by trial and error. Much is still unknown about how to select a suitable wavelet function, and it is worthy of deep research to address this problem.

Another question is that at which level we should perform DWT when using this hybrid method. In our other experiments, 3-level DWT were used, yielding a test set MAPE one or two percent lower. Although the prediction accuracy varied only a small amount between these two experiments, the second experiment required significantly training time for it trained 4 neural networks instead of 2 neural networks for 1-level DWT. It is necessary to consider the tradeoff between the computing intension and predict accuracy.

Acknowledgments

This work is supported by National Natural Science Foundation of People's Republic of China, Grant No.50378016 and Natural Science Foundation of Education Commission of Jiangsu Province, Grant No. 05KJB520056. We would like to thank Prof. Dr. Luc De Raedt for his help.

References

1. Skander, S.: On the Use of the Wavelet Decomposition for Time Series Prediction. Neurocomputing 48(1-4) (2002) 267–277
2. Yao, S.J., Song, Y.H., Zhang, L.Z., Cheng, X.Y.: Wavelet Transform and Networks for Short-term Electrical Load Forecasting. Energy Conversion and Management 41(18) (2000) 1975-1988
3. Teng, H.L., Qi, Y.: Application of Wavelet Technique to Freeway Incident Detection. Transportation Research C 11(3-4) (2003) 289–308
4. Chen, S.Y., Wang, W., Qu, G.f.: Combining Wavelet Transform and Markov Model to Forecast Traffic Volume. the Third International Conference on Machine Learning and Cybernetics (ICMLC2004) 5(2004) 2815-2818
5. Dong, Z., Guo, X., Zheng, J., etc.: Calculation of Noise Resistance by Use of the Discrete Wavelets Transform. Electrochemistry Communications 3(2001)561-565
6. Aballe, M. Bethencourt, Botana, F.J.: Using Wavelets Transform in the Analysis of Electrochemical Noise Data. Electrochimica Acta 44 (1999) 4805–4816
7. Yin, H., Wong, S., Xu, J., Wong, C.: Urban Traffic Volume Prediction Using a Fuzzy-neural Approach. Transportation Research C 10 (2002) 85-98
8. Chen, H., Susan Grant-Muller. : Use of Sequential Learning for Short-term Traffic Volume Forecasting. Transportation Research Part C 9 (2001) 319-336
9. Tom, M. Mitchell.: Machine Learning. 1st edn. China Machine Press, Beijing (2003)
10. Srinivasan, D., Jin, X., Cheu, R.L.: Evaluation of Adaptive Neural Network Models for Freeway Incident Detection. IEEE Transactions on Intelligent Transportation Systems 5(1) (2004)1-11

Prediction of Railway Passenger Traffic Volume by means of LS-SVM

Zhen-Rui Peng[1], Fu Wu[2], and Zhao-Yuan Jiang[1]

[1] School of Mechatronics, Lanzhou Jiaotong University, Lanzhou, 730070, China
{pengzr, jzy}@mail.lzjtu.cn
[2] Key Laboratory of Opto-Electronic Technology and Intelligent Control of Ministry of Education, Lanzhou Jiaotong University, Lanzhou, 730070, China
wufu@mail.lzjtu.cn

Abstract. Based on Least Squares Support Vector Machine (LS-SVM), a method for the prediction of railway passenger traffic volume is proposed. The railway passenger traffic volume from 1985 to 2002, provided by National Bureau of Statistics of China, is employed as total data set. The normalized passenger volume from 1985 to1999 is used as training data set to establish LS-SVM model, while the normalized volume from 1999 to 2002 is utilized as testing data set to carry out prediction. LS-SVM is applied to establish prediction model. The prediction results by LS-SVM model are compared with those by BP neural network method. The results show that LS-SVM outperforms BP neural network in the prediction of railway passenger traffic volume.

1 Introduction

The prediction of railway passenger traffic volume is to analyze the trend of passenger traffic volume and perform quantitative calculation on the basis of qualitative analysis. It has attracted more and more attention because railway passenger traffic volume plays a vital role in economic evaluation of railway project, national resources allocation, and readjustment of investment structure within railway enterprises. Unfortunately, the factors affecting railway passenger traffic volume are complex. It is very difficult to describe the way that affects the passenger traffic volume precisely, which leads to the difficulty in establishing model for the prediction of passenger traffic volume [1]. So, a variety of models have been used in the prediction of passenger traffic volume up to now. Traditional prediction methods include multiple linear regression, stochastic time series, general exponential smoothing, expert system method, etc [2]. Recent research approaches include gray prediction model [3], neural network [4], [5], etc. Although traditional methods can predict approximately the trend of short-time passenger volume, many parameters should be known beforehand and how to modify these parameters under different circumstances should also be known. Essentially, gray prediction model is a kind of exponential increasing prediction methods, which must satisfy the condition that the primal time series is

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 8 – 14, 2006.

nonnegative and monotone. However, sometimes this condition can't be satisfied. The neural network method also causes some inconvenience. Its structure is complicated and many parameters must be estimated. The training of neural network is easy to get stuck into local minima. In addition, neural network uses Empirical Risk Minimization (ERM) principle, which only minimizes the training errors and usually gets poor generalization performance.

Support Vector Machine (SVM) been successfully used in many practical applications such as pattern recognition, function estimation and financial time series [6], [7], [8], [9]. SVM can achieve generalization performance by Structure Risk Minimum (SRM) principle. Least Squares Support Vector Machine (LS-SVM), as a modification of SVM, adopts the least squares linear system as its loss function and therefore solves a set of linear equations. LS-SVM also has good covergence and high precision. Hence, this method is easier to use than quadratic programming solvers in SVM method.

The objective of this work is to put forward a new method for the prediction of railway passenger traffic volume. LS-SVM is used to establish prediction model. The prediction results by LS-SVM model are compared with those by BP nerual network. The results show that LS-SVM outperforms BP neural network in the prediction of railway passenger traffic volume.

The rest of this paper is organized in the following manner. Section 2 describes regression LS-SVM. Section 3 performs the prediction task of railway passenger traffic volume. Section 4, the last part, concludes this work.

2 Regression LS-SVM

The basic idea of regression SVM is to nonlinearly map the training data via mapping function into a higher dimensional feature space, and then obtain a linear regression problem and solve it in this feature space [7]. Suykens [6] originally proposed a modification to the Vapnik's SVM regression formulation. In LS-SVM formulation, equality instead of inequality constraints and a sum squared error (SSE) cost function are used. This new version of SVM greatly simplifies the problem and converges quickly.

Considering regression problem, given l (in this work, l =10.) training data $(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_l, y_l)$, where the input $\mathbf{x}_i \in R^n$ and $y_i \in R$. $\mathbf{x}_i$ denotes the normalized values of passsenger traffic volume in the past five years. y_i denotes the normalized value of passsenger traffic volume in the 6th year. In primal weight space, the regression problem can be represented as the LS-SVM form [6], [9]:

$$\min J(\mathbf{w}, \boldsymbol{\xi}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \gamma\frac{1}{2}\sum_{i=1}^{l}\xi_i^2 \ . \tag{1}$$

$$y_i = \mathbf{w}^T\varphi(\mathbf{x}_i) + b + \xi_i, \quad i = 1, 2, 3, \cdots, l \ . \tag{2}$$

where $\varphi(\mathbf{x})$ represents a high dimensional feature space, which is nonlinearly mapped from the input space; γ is the regularization constant; $\xi = (\xi_1, \xi_2, .., \xi_l)$ is the error vector; $\mathbf{w}$ and b are weight vector and biasis term, respectively. Construct the Lagrangian:

$$L(\mathbf{w}, b, \xi, \alpha) = J(\mathbf{w}, b, \xi) - \sum_{i=1}^{l} \alpha_i \left\{ \mathbf{w}^T \varphi(\mathbf{x}_i) + b - y_i + \xi_i \right\}. \tag{3}$$

where $\alpha_i \in R$ are the Lagrange multipliers. The conditions for optimality lead to a set of linear equations:

$$\begin{bmatrix} 0 & \mathbf{l}_v^T \\ \mathbf{l}_v & \Omega + \frac{1}{\gamma}\mathbf{I} \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{y} \end{bmatrix}. \tag{4}$$

with $\mathbf{y} = [y_1, \cdots, y_l]$, $\mathbf{l}_v = [1, \cdots, 1]$, $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_l]$. $\Omega = K(\mathbf{x}_i, \mathbf{x}_j)$ is a kernel function satisfying Mercer's conditions. Three typical kernel functions are listed in Table 1. The LS-SVM regression formulation is then constructed:

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b \ . \tag{5}$$

Table 1. Typical kernel functions

Kernel function	Expression
Linear kernel	$\mathbf{x}_i^T \mathbf{x}$
Polynomial kernel	$(1 + \mathbf{x}_i^T \mathbf{x})^d$
RBF kernel	$\exp(-\left\|\mathbf{x} - \mathbf{x}_i\right\|^2 / \sigma^2)$

3 Prediction of Railway Passenger Traffic Volume

The prediction is conducted in the following way. Use the actual railway passenger traffic volume from 1985-2002 as the primal data set; normalize the total passenger traffic volume; divide the total normalized volume data into training data set and testing data set; by introducing five-fold Cross Validation (CV) technique, establish LS-SVM model for passenger traffic volume prediction according to the training data set; perform prediction using testing data set; compare the LS-SVM prediction results with those by BP neural network.

3.1 Data Preparation

The primal data of railway passenger traffic come from the operation and management departments of the national, local and joint-venture railways, not including the

railways for military purpose, industrial lines and special railway lines. The 18-years passenger data from 1985 to 2002, provided by National Bureau of Statistics of China and railway sub-bureaus, consist of total data set [10].

In order to process these data with LS-SVM conveniently, and compare the prediction results by LS-SVM method with those by BP neural network method expediently, normalize the passenger traffic volume [1]:

$$x^{'}(i) = \frac{x(i)}{x_{\max}(i)} \ . \tag{6}$$

where $x^{'}(0)$, $x^{'}(1)$, $x^{'}(2)$, $x^{'}(3)$, … , $x^{'}(17)$ denote the time series of the normalized values of passenger traffic volume from 1985 to 2002, respectively. $x(i)(i = 0,1,2,\ldots,17)$ is the *i*th passenger traffic volume. $x_{\max}$ denotes the largest passenger traffic volume from 1985 to 2002. In our work, $x_{\max}$ =122645, its corresponding normalized value is 0.9999.

3.2 Prediction Methodology and Model

Because many Chinese enterprises take "five year period" to make plans, in this work, the passenger traffic volume in the "past 5 years" is used to predict the passenger volume in 6th year. That is, the prediction model has five inputs and one output. The normalized values from 1985 to 1999 constitute the training data set to establish the LS-SVM model for the prediction of passenger traffic volume. The normalized values from 1999 to 2002 are used as testing data set to predict the railway passenger volume. The detailed data are listed in Table 2.

The prediction performance is evaluated using the following metric, namely, the Mean Square Error (MSE):

$$\mathrm{MSE} = \frac{1}{k-1}\sum_{i=1}^{k}\left|a_i - r_i\right|^2 \ . \tag{7}$$

where a_i denotes the actual passenger traffic volume. r_i denotes the predicted volume and k denotes the total number of passenger traffic volume in testing data set.

Owing to the good performance of radial basis kernel function in nonlinear approximation, it is selected as the kernel function of LS-SVM. The parameters of LS-SVM directly influence its generalization performance. Hence, they must be chosen elaborately. In this work, by using the procedure of grid-search and step-by-step refinement, the optimal parameters of LS-SVM can be determined. i.e., σ^2 =2 (kernel parameter), γ =7 (regularization constant), and MSE= 0.00541.

3.3 Prediction Results

The prediction results by using the LS-SVM prediction model (8) in the testing data set are listed in Table 3. From Table 3, the maximal relative prediction error is 5.482%.

Table 2. Training data set and testing data set [1]

Data set	Year	Input data					Actual data
		$x'(i)$	$x'(i+1)$	$x'(i+2)$	$x'(i+3)$	$x'(i+4)$	$x'(i+5)$
Training data set	1990	0.9141	0.8853	0.9171	0.9999	0.9279	0.7804
	1991	0.8853	0.9171	0.9999	0.9279	0.7804	0.7752
	1992	0.9171	0.9999	0.9279	0.7804	0.7752	0.8129
	1993	0.9999	0.9279	0.7804	0.7752	0.8128	0.8598
	1994	0.9279	0.7804	0.7752	0.8128	0.8598	0.8866
	1995	0.7804	0.7752	0.8128	0.8598	0.8866	0.8377
	1996	0.7752	0.8128	0.8598	0.8866	0.8377	0.7729
	1997	0.8128	0.8598	0.8866	0.8377	0.7729	0.7608
	1998	0.8598	0.8866	0.8377	0.7729	0.7608	0.7753
	1999	0.8866	0.8377	0.7729	0.7608	0.7753	0.8167
Testing data set	2000	0.8377	0.7729	0.7608	0.7753	0.8167	0.8567
	2001	0.7729	0.7608	0.7753	0.8167	0.8567	0.8574
	2002	0.7608	0.7753	0.8167	0.8567	0.8574	0.8611

In order to further validate the LS-SVM model, the prediction is also carried out with BP neural network (BP-NN) method, which has been widely used in many applications. MATLAB 6.5 provides a neural network toolbox named "NN Toolbox4.0.2", which has the convenient Graphics User Interface (GUI). Utilizing this friendly toolbox, we can easily realize all the tasks relative to the neural network prediction such as selection of transfer function, network structure, initialisation, training, and simulation. The prediction results of passenger traffic volume with BP-NN method are also listed in Table 3. From Table 3, the maximal prediction relative error is 6.753%, which is larger than the maximal prediction relative error (5.482%) with using LS-SVM model. The relative prediction errors of the other two years are also greater than those obtained with LS-SVM model.

For the training data set (owing to space limitation, concrete prediction data are omitted.), the prediction results using both LS-SVM model and BP-NN method are also given as follows. For LS-SVM model, the MSE is 0.00054 and its corresponding maximal relative error is 3.6%; for BP-NN method, the MSE is only 0.00013 and its corresponding maximal relative error is -2.3%. Fig.1 shows the prediction results of railway passenger traffic volume from 1990 to 2002 (both in the training set and testing set) by using these two methods. For both of the two methods, the maximal prediction errors happen in 2002. The difference between actual volume value and prediction volume value with BP-NN method is 713157, while the difference between actual volume value and prediction volume value with LS-SVM model is 578932. Meanwhile, the prediction results of the other two years, obtained by LS-SVM model, are also close to the actual volume values.

It is obvious that, in the training data set, the prediction errors using BP-NN method are less than those by using LS-SVM model. While, in the testing data set, the prediction errors using LS-SVM model are less than those by using BP–NN method.

[1] The passenger traffic volume values are the normalized ones.

Table 3. Comparison of passenger prediction results by LS-SVM and BP-NN in the testing data set [2]

Year	Actual volume	Prediction value		Relative error (%)		MSE	
		LS-SVM	BP-NN	LS-SVM	BP-NN	LS-SVM	BP-NN
2000	0.8567	0.8417	0.8804	1.800	-3.418		
2001	0.8574	0.8522	0.8838	0.605	-2.988	0.00053	0.0023
2002	0.8611	0.8139	0.8066	5.482	6.753		

The results are natural. Because the LS-SVM is based on SRM principle, which seeks to minimize an upper bound of the generalization error consisting of both the training error and a confidence interval, while neural network is based on ERM principle, which only minimizes the training error. Thus, the generalization performance of neural network is not as good as that of LS-SVM.

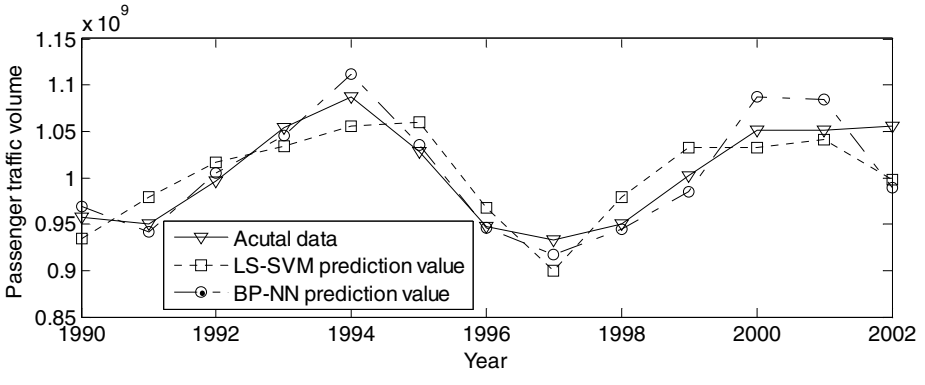


Fig. 1. Comparison of prediction results between LS-SVM and BP NN methods

4 Conclusions

A new method for the prediction of railway passenger traffic volume is put forward based on the LS-SVM modeling. LS-SVM is a kind of machine learning method based on SRM principle, which has better generalization performance for the regression problem of small samples. This method can effectively overcome the inherent drawbacks in neural network, such as the complex structure and getting stuck into local minima. The results show that LS-SVM prediction model is effective and LS-SVM outperforms BP neural network in railway passenger traffic prediction. It is needed to point out that present work is sheer data driving and quantitative. The future work is to combine qualitative analysis with quantitative calculation. This can make up the deficiency of current method to provide more valuable information for the decision-making of railway enterprise.

[2] The passenger traffic volume values are the normalized ones.

Acknowledgements

This paper is supported by"Qing Lan"Talent Engineering Funds by Lanzhou Jiaotong University. The authors would like to thank the referees for their helpful comments and suggestions.

References

1. Wang, Z., Wang, Y.H., Jia, L.M., Li, P.: The Application of Improved BP Neural Network in the Prediction of Railway Passenger Volume Time Serial. China Railway Science 26(2) (2005) 128-131
2. Moghram, L., Rahrnan, S.: Analysis and Evaluation of Five Short-Term Load Forecasting Techniques. IEEE Trans. Power Syst. 4(4) (1989) 1484-1491
3. Linwas, K.H., Liu, B.D.: A Gray System Modeling Approach to the Prediction of Calibration Intervals. IEEE Trans. Instrum. Meas. 54(1) (2005) 297-304
4. Hippert, H.S., Pedriera, C.E., Souza, R.C.: Neural Networks for Short-Term Load Forecasting: A Review and Evaluation. IEEE Trans. Power Syst. 16(1) (2001) 44-55
5. Khotanzad, A., Elragal, H., Lu, T.L.: Combination of Artificial Neural-Network Forecasters for Prediction of Natural Gas Consumption. IEEE Trans. Neural Networks 11(2) (2000) 464-473
6. Suykens, J.A.K., Vandewalle, J.: Least Squares Support Vector Machine Classifiers. Neural Process. Lett. 9(3) (1999) 290-301
7. Vapnik, V.N.: Statistical Learning Theory. Wiley, New York (1998)
8. Wu, C.H., Ho, J.M., Lee, D.T.: Travel-Time Prediction with Support Vector Regression. IEEE Trans. Intell. Transp. Syst. 5(4) (2004) 276-281
9. Suykens, J.A.K., Van Gestel, T., De Brabanter, J., De Moor, B., Vandewalle, J.: Least Squares Support Vector Machines. World Scientific, Singapore (2002)
10. National Bureau of Statistics of China: China Statistical Yearbook. China Statistics Press, Beijing (1983-2003)

Traffic Flow Modeling of Urban Expressway Using Artificial Neural Networks

Guo-Jiang Shen

The National Key Lab. of Industrial Control Technology,
Yuquan Campus, Zhejiang University,
Hangzhou, China
gjshen@iipc.zju.edu.cn

Abstract. From the viewpoint of macroscopic dynamic characteristics of urban expressway traffic flow, a commonly used macroscopic dynamic deterministic traffic flow model is analyzed, and the 1.5-layer feed-forward network modeling for the urban expressway traffic flow is established. Hangzhou urban expressway is simulated and the result demonstrates that the neural network model is able to reproduce traffic congestion built in realty with considerable accuracy, thus making it suitable for evaluating various control strategies and performing further modeling and simulation tasks.

1 Introduction

Urban expressway has great attraction to drivers. However, the notoriously increasing number of vehicles that use the provided urban expressway capacity has lead to severe problems in the form of recurrent and no recurrent congestion resulting in serious economic and environmental problems, as well as increased public frustration and discomfort. The reliable traffic models become important for traffic simulation, prediction and control [1-6]. We analyze a commonly used macroscopic dynamic deterministic traffic flow model of urban expressway and use 1.5-layer feed-forward network to estimate model parameters. The main goal of the paper is to describe the applying approach and modeling procedures, and to demonstrate the accuracy and usefulness of macroscopic model for urban expressway.

2 Modeling Approach

An expressway link m is divided into N_m segments with length of $L_{m,i}$ (Fig.1), each segment having uniform characteristics. The nodes are placed at locations where a major change in road geometry occurs, as well as at junctions, on-ramps, and off-ramps. For each segment i of each link m at each time instant $t= k\cdot T$, k=0,1,2,…,K, where K is the time horizon, the following macroscopic variables are defined. *Traffic density*: $\rho_{m,i}(k)$ (veh/km·lane) is the number of vehicles in segment i of link m at time

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 15 – 22, 2006.

$k \cdot T$ divided by the length of the segment $L_{m,i}$ and by the number of lanes λ_m .*Mean speed*: $v_{m,i}(k)$ (km/h) is the mean speed of the vehicles included in segment *i* of link *m* at time $k \cdot T$. *Traffic flow*: $q_{m,i}(k)$ (veh/h) is the number of vehicles leaving segment *i* of link *m* during the period $[k \cdot T, (k+1) \cdot T]$, divided by *T*.

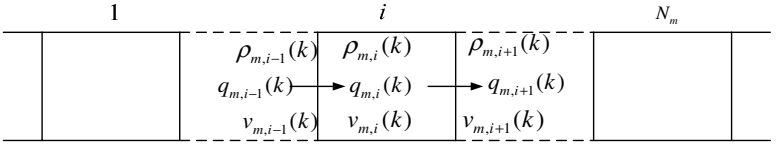


Fig. 1. Discretised expressway link

2.1 Link Model

Expressway links: the basic equations used to calculate the traffic variables for every segment i of expressway link m are as follow

$$q_{m,i}(k+1) = \varepsilon \lambda_m \rho_{m,i}(k) v_{m,i}(k) + (1-\varepsilon) \lambda_m \rho_{m,i+1}(k) v_{m,i+1}(k), \tag{1}$$

$$\rho_{m,i}(k+1) = \rho_{m,i}(k) + \frac{T}{L_{m,i} \lambda_m} \left[q_{m,i-1}(k) - q_{m,i}(k) \right], \tag{2}$$

$$\begin{aligned} v_{m,i}(k+1) = {} & v_{m,i}(k) + \frac{T}{\tau} \left\{ V\left[\rho_{m,i}(k)\right] - v_{m,i}(k) \right\} + \frac{T}{L_{m,i}} v_{m,i}(k) \left[v_{m,i-1}(k) - v_{m,i}(k) \right] \\ & - \frac{\mu_{m,i}(k) T}{\tau L_{m,i}} \frac{\rho_{m,i+1}(k) - \rho_{m,i}(k)}{\rho_{m,i}(k) + \delta} \end{aligned}, \tag{3}$$

$$V(\rho_{m,i}(k)) = v_{f,m} \cdot \exp\left[-\frac{1}{a_m} \left(\frac{\rho_{m,i}(k)}{\rho_{cr,m}} \right)^{a_m} \right], \tag{4}$$

$$\mu_{m,i}(k) = \begin{cases} \mu_1 \dfrac{\gamma}{\rho_{jam,m} - \rho_{m,i+1}(k) + \beta} & if\ \rho_{m,i+1}(k) > \rho_{m,i}(k) \\ \mu_2 & else \end{cases}, \tag{5}$$

where $v_{f,m}$ denotes the free-flow speed of link *m*, $\rho_{cr,m}$ denotes the critical density per lane of link *m* (the density where the maximum flow in the link occurs), $\rho_{jam,m}$ denotes the congested density per lane of link *m* (the density where the minimum speed in the link occurs), and a_m is a parameter of the fundamental diagram (eqn. (4)) of link *m*. furthermore, τ, a time constant, and ε, δ, μ_1, μ_2, γ, and β, are

constant parameters same for all network links. $v_{f,m}$, $\rho_{cr,m}$, $\rho_{jam,m}$ and a_m, are constant parameter which reflect particular characteristics of a given traffic system and depend on lane geometry, vehicle characteristics, driver's behavior etc.

Origin links: For origin links, a simple queue model is used. Origin links are characterized by their flow capacity and their queue length. The outflow $q_o(k)$ of an origin link *o* is given by

$$q_o(k) = r_o(k) \cdot \min\left\{ d_o(k) + \frac{\omega_o(k)}{T}, q_{\max,o}(k) \right\}, \tag{6}$$

where $d_0(k)$ is the demand flow at time period *k* at origin *o*, $\omega_o(k)$ is the length in vehicles of a possibly existing queue at time period *k*, $q_{\max,o}(k)$ is the flow capacity at the specific period, and $r_o(k) \in [r_{\min}, 1]$ is the metering rate for origin *o* link at period *k*. The flow capacity depends on the density of the primary downstream leaving link *m* in the following way

$$q_{\max,o}(k) = \begin{cases} Q_o & \text{if } \rho_{m,1}(k) < \rho_{cr,m} \\ Q_o \cdot \dfrac{\rho_{\max} - \rho_{m,1}(k)}{\rho_{\max} - \rho_{cr,m}} & \text{else} \end{cases}, \tag{7}$$

where Q_o is the constant flow capacity of the origin link and $\rho_{\max}$ is the maximum possible density in the expressway link.

The conservation equation for an origin link yields

$$\omega_o(k+1) = \omega_o(k) + T \cdot \left[d_o(k) - q_o(k) \right]. \tag{8}$$

2.2 Node Model

Let $Q_p(k)$ be the total traffic volume entering an expressway node *p* at time period *k*. Then the turning rate $\beta_{p,m}(k)$ is the portion of traffic volume $Q_p(k)$ which leaves node *p* at time period *k* through link $m \in O_p$, where O_p is the set of links leaving node *p*. Let I_p be the set of links entering node *p*. The following equations hold

$$Q_p(k) = \sum_{n \in I_p} q_n^{N_n}(k) \quad \forall p, \tag{9}$$

$$q_{m,0}(k) = \beta_{p,m}(k) \cdot Q_p(k) \quad \forall m \in O_p, \tag{10}$$

where $q_{m,0}(k)$ is the traffic volume that leaves node *p* via out link *m*. Equations (9) and (10) provide $q_{m,0}(k)$ needed in (2) for *i*=1.

When node *p* has more than one leaving link, the virtual downstream density $\rho_{m,N_m+1}(k)$ and speed of entering link *m* is given by

$$\rho_{m,N_m+1}(k) = \frac{\sum_{u \in O_p} \rho_{u,1}^2(k)}{\sum_{u \in O_p} \rho_{u,1}(k)}, \tag{11}$$

$$v_{m,N_n+1}(k) = \frac{\sum_{m \in O_p} v_{u,1}^2(k)}{\sum_{m \in O_p} v_{u,1}(k)}, \tag{12}$$

where $\rho_{m,N_m+1}(k)$ is the virtual density downstream of the entering link m to be used in eqn. (1) and (3) for $i = N_m$, and $\rho_{u,1}(k)$ is the density of the first segment of leaving link u. $v_{m,N_m+1}(k)$ is the virtual mean speed downstream of the entering link m to be used in eqn. (1) for $i = N_m$, and $v_{u,1}(k)$ is the mean speed of the first segment of leaving link u.

When node p has more than one entering link, the virtual upstream speed $v_{m,0}(k)$ of leaving link m is given by

$$v_{m,0}(k) = \frac{\sum_{u \in I_p} v_{u,N_u}(k) \cdot q_{u,N_u}(k)}{\sum_{u \in I_p} q_{u,N_u}(k)}, \tag{13}$$

where $v_{m,0}(k)$ is the virtual speed upstream of the leaving link m that is needed in eqn. (3) for i=1.

2.3 Model Summary

From the previous sections, a nonlinear dynamic model of the form

$$\boldsymbol{X}(k+1) = \boldsymbol{F}\left[\boldsymbol{X}(k), \boldsymbol{C}(k), \boldsymbol{D}(k)\right], \boldsymbol{X}(0) = \boldsymbol{X}_0, \tag{14}$$

can be obtained by substituting (1), (9), (10), (11), (12)into (2); (4), (5), (11), (13) into (3); and (6), (7) into (8), where $\boldsymbol{X}$ is the state vector consisting of the densities $\rho_{m,i}$ and mean speeds $v_{m,i}$ of every segment i of every link m, and the queues ω_o of every origin link o, $\boldsymbol{C}$ is the control vector including the ramp metering rates r_o of every on-ramp, and $\boldsymbol{D}$ is the disturbance vector consists of the demand d_o at every origin link o and the turning rates $\beta_{p,m}$ at every bifurcation node p.

3 Modeling by Neural Networks

Here, we focus on one of the most popular neural networks, called sigmoidal feed-forward networks. The output of the processing element is the nonlinear sigmoidal function of the sum of the inputs and a possible threshold.

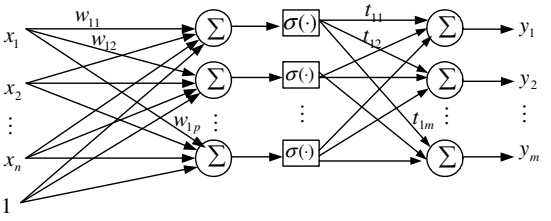


Fig. 2. A 1.5-layer feed-forward network

We will model the state variable **x** in the equation (14) by a 1.5-layer feed-forward network with p processing elements (in Fig.2.). Let $\boldsymbol{\theta} = \left[w_{11}, \cdots, w_{n+1,p}, t_{11}, \cdots, t_{pm} \right]^T$. The Output vector $\boldsymbol{Y}$ of the network is $\left[\cdots, \rho_{m,i}(k+1), v_{m,i}(k+1), w_o(k+1), \cdots \right]^T$, and it is just a weighted sum of the outputs of the hidden layer, and the input vector $\boldsymbol{X}$ is $\left[\cdots, \rho_{m,i}(k), v_{m,i}(k), w_o(k), r_o(k), d_o(k), \beta_{p,m}(k), \cdots \right]^T$. Then the input-output relationship of the neural network can be expressed as

$$\boldsymbol{Y} = \boldsymbol{N}(\boldsymbol{X}; \boldsymbol{\theta}) \ , \tag{15}$$

$$y_j = \sum_{l=1}^{p} t_{lj} \sigma \left(\sum_{i=1}^{n} w_{il} x_i + w_{n+1,l} \right), \ j = 1, \cdots, m \ , \tag{16}$$

where p is the number of processing elements of 1.5-layer feed-forward network and decided by real traffic conditions.

Without loss of generality, equation (14) can be described by

$$\boldsymbol{X}(k+1) = \boldsymbol{A}\boldsymbol{X}(k) + \boldsymbol{G}\left[\boldsymbol{X}(k) \right], \ \boldsymbol{X}(0) = \boldsymbol{X}_0 \ , \tag{17}$$

where $\boldsymbol{G}(\bullet)$ is an unknown nonlinear function, $\boldsymbol{A}$ is an arbitrary constant vector.

Let $\boldsymbol{N}_G(\bullet; \boldsymbol{\theta}_G)$ be the neural network that identifies the nonlinear function $\boldsymbol{G}(\bullet)$ that is used to generate the estimates $\hat{\boldsymbol{X}}(k+1)$ that are as close to measured $\boldsymbol{X}(k+1)$ as possible for $i = 1, \cdots, n$. Then the following model generates the state estimates

$$\hat{\boldsymbol{X}}(k+1) = \boldsymbol{A}\boldsymbol{X}(k) + \boldsymbol{N}_G(\bullet; \boldsymbol{\theta}_G) \ , \tag{18}$$

that correspond to the weights $\boldsymbol{\theta}_G(k)$ at time kT.

The error

$$\left\| \boldsymbol{e}(k) \right\| = \left\| \hat{\boldsymbol{X}}(k+1) - \boldsymbol{X}(k+1) \right\| = \left\| \boldsymbol{N}_G(\bullet; \boldsymbol{\theta}_G(k)) - \boldsymbol{G}(k) \right\| . \tag{19}$$

The adjustment rule for the weights $\boldsymbol{\theta}_G$ is chosen as [3]

$$\boldsymbol{\omega}(k+1) = \boldsymbol{\theta}_G(k) - \frac{\gamma_0}{\beta_0 + \left\|\xi_i(k)\right\|^2}\boldsymbol{\xi}(k)\boldsymbol{e}(k), \tag{20}$$

$$\boldsymbol{\theta}_G(k+1) = \begin{cases} \boldsymbol{\omega}(k+1) & if \ \left\|\boldsymbol{\omega}(k+1)\right\| \le \left\|\boldsymbol{M}_{\boldsymbol{\theta}_G}\right\| \\ \dfrac{\left\|\boldsymbol{M}_{\boldsymbol{\theta}_G}\right\|}{\left\|\boldsymbol{\omega}(k+1)\right\|}\boldsymbol{\omega}(k+1) & else \end{cases}, \tag{21}$$

$$\boldsymbol{\xi}(k) = \frac{\partial^T \boldsymbol{N}_G\left(\bullet;\boldsymbol{\theta}_G(k)\right)}{\partial \boldsymbol{\theta}_G(k)}, \tag{22}$$

where $0 < \gamma_0 < 2$, $\beta_0 > 0$ and $\boldsymbol{M}_{\boldsymbol{\theta}_G}$ are design parameters.

The performance of the learning of $\boldsymbol{N}_G(\bullet;\boldsymbol{\theta}_G)$ can be evaluated by

$$MSE(n) = \frac{1}{n}\sum_{k=1}^{n}\boldsymbol{e}(k). \tag{23}$$

It can be showed by [3] that if $\boldsymbol{G}(\bullet)$ is linear with respect to $\boldsymbol{\theta}_G$ and can parameterized to be of the same from as $\boldsymbol{N}_G(\bullet;\boldsymbol{\theta}_G)$ with a corresponding unknown $\boldsymbol{\theta}_G^*$ then (20) guarantees that $\boldsymbol{e}(k) \to 0$ as $k \to \infty$.

4 Computer Simulation

The objective of this section is to describe the model validation by simulation aiming at enabling the expressway model to represent traffic conditions with sufficient accuracy.

Shangtang expressway is main motorway form north to south in Hangzhou China, and contains nine main junctions. This expressway is subject to considerable recurrent congestion. Congestion is especially heavy form Desheng road to Xihu road at early and night peak time. The Shangtang expressway was modeled in both directions, the total length of the expressway is 16 km (both directions), and the total number of links that was used to model the expressway is 24. This number includes 6 expressway links, and 18 or origin links. The expressway links were divided into a total of 36 segments, the length of each segment ranges form 380 to 690 m.

We use a 1.5-layer feed-forward network of twelve basic processing elements (p=36) for N_G .For the 1.5-layer feed-forward network training procedure, data from loop detectors for four consecutive days (Feb. 5-9 2005) were available. These data consisted of one-minute measurements of flow and speed for the whole day. The density was calculated by the traffic and the mean speed in advance.

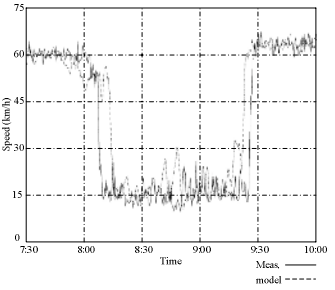


Fig. 3. Stretch 2, measured versus predictive speed

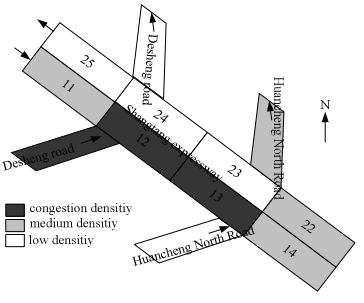


Fig. 4. Simulated traffic conditions in a part of expressway, 8:15 A.M. 2005

Based on the 1.5-layer feed-forward network that is trained and the real data from loop detectors for Feb. 15 2005, examples of the model output for single location can be seen in Fig. 3 and 4. Fig. 3 depicts the speed trajectory determined by the model and compared with the actual speed measurements. Fig. 3 is indicative of the model's ability to represent traffic conditions in each of the expressway stretches based on the 1.5-layer feed-forward network. Fig. 4 shows the model's prediction of traffic conditions at 8:15 A.M 2005. We find the model' prediction result is very close to the real condition. From this figure it can be seen that the model reproduces the previously described recurrent congestions sufficiently, thus making it a suitable tool for evaluating the impact of various traffic control measures on the traffic flow process.

5 Conclusion

This paper presented the developed model of the urban expressway based on METANET model, and adopted the 1.5-layer feed-forward network for modeling traffic flow on the macroscopic level trained by real data form loop detectors. The 1.5-layer feed-forward network consists of layers and their weights are adjusted by robust adaptive laws. The simulation result obtained form this approach against real data demonstrate it's effectivity and that the expressway model described by 1.5-layer feed-forward network is able to reproduce traffic congestion built in realty with considerable accuracy, thus making it suitable for evaluating various control strategies

and performing further modeling and simulation tasks. The off-line evaluation of expressway control measures ramp metering, motorway-to-motoway control and route guidance will be the subject of further work for urban expressway.

Acknowledgment

This work was supported in part by Natural Science Fund of China (NSFC-60203030), Advanced Research Plan between China and France (PRA SI01-04) etc, Science development Plan of Hangzhou (20051331B02). We would like to thank the teachers and classmates being with The National Key Lab. of Industrial Control Technology Zhejiang University in China for his/her help.

References

1. Papageorgiou, M., Blosseville, J.M., Hadj-Salem, H.: Modeling and Real-time Control of Traffic Flow on the Southern Part of Boulevard Peripherique in Paris, Part 1: Modeling. Transpn. Res-A 24 (5) (1990) 345-359
2. Papageorgiou, M., Blosseville, J.M., Hadj-Salem, H.: Modeling and Real-time Control of Traffic Flow on the Southern Part of Boulevard Peripherique in Paris, Part 2: Control. Transpn. Res-A 24 (5) (1990) 361-370
3. Ho, F.S., Ioannou, P.: Traffic **F**low Modeling and Control Using Artificial Neural Networks. IEEE Control Systems 16(5) (1996) 16-26
4. Shen. G.J., Sun Y.X.: Control-oriented Macroscopic Dynamic Model in Urban Traffic Network. Journal of Zhejiang University (Engineering Science) 39(10) (2005) 1485-1489
5. Huang, X.Y., Li, B.J.: Neural Controller for Highway Traffic. Information and Control 29(3) (2000) 280-284
6. Irwin, G., Warwick, K., Hunt, K.: Neural Networks for Control and System, Peter Peregrinus Ltd. (1992)

Radial Basis Function Network for Traffic Scene Classification in Single Image Mode

Qiao Huang, Jianming Hu, Jingyan Song, and Tianliang Gao

Department of Automation, Tsinghua University, Beijing, 100084, China
huangqiao@mails.tsinghua.edu.cn,
{hujm, jysong}@mail.tsinghua.edu.cn,
gtl03@mails.tsinghua.edu.cn

Abstract. In this paper, a radial basis function (RBF) network based method inspired by mature algorithms for face recognition is applied to classify traffic scenes in single image mode. Not to follow traditional ways of estimating traffic states through image segmentation and vehicle tracking, this method avoids complicated problems in digital image processing (DIP) and can operate on just one image, while the old ones rely on consecutive images. The proposed method adopts discrete cosine transform (DCT) for feature selection, then a supervised clustering algorithm is fulfilled to help design hidden layer of RBF network for which Gaussian function is chosen, finally linear least square (LLS) is used to solve the weights training problem. Experiments show that this method is valid and effective under the new application background.

1 Introduction

Nowadays, transportation is playing a very important role in the balanced running of social and economic systems with the rapid development of economy and society. However, urban traffic problems such as traffic accidents, traffic congestions, are becoming more and more serious in almost all large cities in the world.

Traffic problems arisen in the cities may bring into tremendous losses to lives and properties of the urban residents. Especially in developing countries, with the sharp increasing of vehicle ownership in large cities, traffic accidents and congestions occurred much more frequently. Since the traffic is floating and evolutive, the urban traffic problems may be explosive and chained, in which a large amount of persons will be involved.

In order to deal with the problem, Urban Traffic Surveillance System (UTSS), as a modernization symbol of a city, should be established as soon as possible. UTSS is composed of four indispensable steps, i.e., monitoring, prediction, alarm and decision-aided support. Monitoring is the first and the most important step in UTSS. If a traffic incident such as an accident or a traffic jam is monitored, detected and dealt with in time, the loss caused by will be considerably reduced. Actually, some urban traffic monitoring systems, called Closed Circuit Television (CCTV) have been implemented in many metropolitans and freeways all over the world. The video images

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 23 – 32, 2006.

are collected and transmitted to the traffic control center via optical fiber networks. The traffic manager can obtain the traffic video information in real-time, find out traffic incidents and issue traffic control instructions.

Traffic state recognition, one of the most fundamental works for traffic surveillance system, makes itself an attractive and interested research direction in recent years. In this paper, macroscopic traffic information such as congestion is extracted from training a radial basis function network. Traditionally, inductive loops buried in the roadway are used to detect vehicles and further to estimate traffic parameters, but these years the rising vision-based methods gradually replace the destructive old technique due to their easily-deployment and abundant functions.

In this context, various digital image processing and computer vision techniques are always referred and most researchers follow a general process of background update, image segmentation and vehicle tracking [1]. However, these approaches are heavily computative and sensitive to complicated environment, e.g. occlusion between vehicles or between vehicle and obstacles, the shadow variations under different lighting conditions and etc... Casting off such restriction, the entropy concept is introduced to design a novel system for traffic flow monitoring [2]. Hidden Markov Models (HMM) and Maximum Likelihood (ML) criterion are applied to detect congestions in MPEG-2 coded videos [3]. Nevertheless, the same drawback of these approaches is that they depend on consecutive frames, hence limit the application domain. Compared with these existed algorithms, the method proposed in this paper just needs information of a single image and neatly avoids complex DIP steps. Considering sometimes exact traffic parameters are not so necessary while macroscopic traffic state is already enough, this method really makes sense.

Based on this idea, a mature processing architecture [4] in face recognition field is used for reference and to some extent modified to perform traffic state classification. Classifier is built through supervised clustering and learning of a Gaussian RBF network. Before, DCT is chosen to fulfill feature selection.

The remainder of this paper is organized as follows. In section 2, details of RBF neural network are described including algorithms and adjustments in the whole building and training process. In section 3, experiments and corresponding results are presented. At last, a conclusion is drawn and future work is discussed in section 4.

2 Algorithm Description of Traffic Scene Classification Based on RBF Neural Network

Radial Bases Function (RBF) neural network, core of the classifier for traffic state recognition, is a feed forward neural network. Well known for its capability of universal approximation, RBF basically has a three-layer structure, as shown in **Fig.1**. In this paper, the first layer accepts processed image features as the inputs; in the hidden layer, neuron nodes whose centers correspond to the patterns of input feature space, construct a nonlinear mapping through computing a set of Gaussian radial functions on the input vectors; In the last layer, weighted summations of the mid results are computed and outputted, from which, traffic states are classified. Details are described in the next parts.

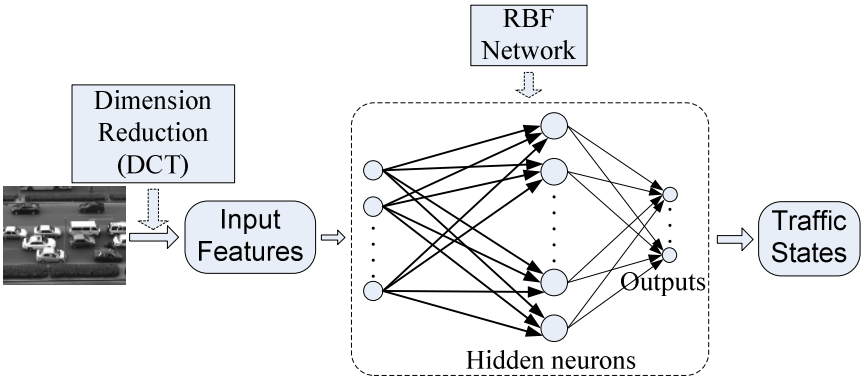


Fig. 1. The Flow Chart of Traffic State Classification Based on RBF

2.1 Input Vector: Feature Selection

Like face recognition problem, inputting raw pixel values of traffic scene image into a network for learning is very inefficient due to the huge dimensionality and high information redundancy. Principal component analysis (PCA) is widely used as the dimensionality reduction method. However, belonging to linear techniques, PCA isn't suitable for modeling nonlinear data such as information contained in traffic images [5]. Moreover, strong dependency on samples of PCA is demonstrated by many researchers. Additionally, its computational complexity is also a drawback that can't be ignored.

DCT is also a pop technique in signal processing community, especially for data compression, e.g. MPEG coding scheme uses DCT to reduce spatial redundancy in video frames [3]. When a fast algorithm is implemented, DCT can hold good information concentrating ability while lighten greatly the computation burden. Formulas are defined as Equation (1), for an M*N image A(m, n):

$$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+1)q}{2N}, \quad \begin{matrix} 0 \le p \le M-1 \\ 0 \le q \le N-1 \end{matrix}$$
$$\alpha_p = \begin{cases} 1/\sqrt{M}, & p = 0 \\ \sqrt{2M}, & 1 \le p \le M-1 \end{cases} \qquad \alpha_q = \begin{cases} 1/\sqrt{N}, & q = 0 \\ \sqrt{2N}, & 1 \le q \le N-1 \end{cases} \quad . \tag{1}$$

Considering correlations between different parts in a traffic scene, DCT is simply applied on the entire image, not to split it into sub-blocks so that some important relationship information can be preserved [6]. The result is a 2-D matrix with the same size as the image being transformed, containing all spatial frequency components. Actually, in such a DCT coefficients matrix for a real world image, most components have smaller magnitudes than the left minority [5], hence, by discarding these components, dimensionality reduction can be achieved without destroying the essential features. Since the larger ones are mainly located in the upper-left corner of the matrix, we follow a zigzag path [6] from this position to the bottom-right of the matrix, converting the 2-D matrix to a 1-D vector, as in **Fig.2**:

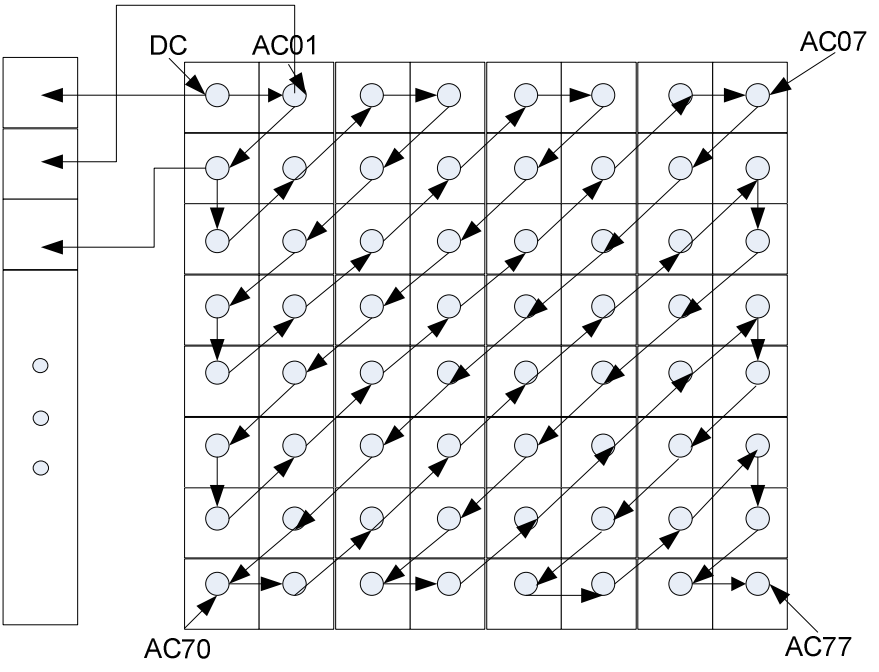


Fig. 2. Convert DCT Coefficient Matrix to 1-D Vector [6]

Besides, as researchers have pointed out, few of the lowest-frequency DCT coefficients are discarded in this paper in order to alleviate influences on the images of illumination variations. Different lengths of truncated vector for the same image are tested in our experiments.

2.2 Hidden Layer Nodes: Clustering

The kernel basis functions in the hidden layer of a RBF network are normally chosen to be Gaussians with two kinds of important parameters, the centers and the widths, giving radially symmetric responses for the Gaussian kernels [8]:

$$g_u(X) = \exp\left(-\frac{\|X - C_u\|^2}{w_u^2}\right). \tag{2}$$

Where C_u, w_u are respectively the center and width of the uth hidden node.

And it should be noted that the Euclidean distance definition is chosen in the formula above. Meanwhile, size of the hidden layer is also crucial for the network's quality: too small size can't represent well the varieties of structures within class and between classes, while too big size may lead to nontrivial computation. Customarily, k-means and other unsupervised clustering methods are used to acquire these parameters, nevertheless, now that traffic state in a sample image is known, a supervised

approach may lead to better clustering result. In this paper, integration of those methods for face recognition proposed in [4] and [6] is made and improved according to the new case to split the traffic state classes into more sub-clusters.

Below are the steps it follows:

STEP 1: Initially assume each sample in a class is a single cluster. Define:

$$c_{i,p} = \frac{1}{n_{i,p}} \cdot \sum_{t=1}^{n_{i,p}} X_j^{i,p} \qquad \sigma_{i,p} = \sqrt{\frac{1}{n_{i,p}} \cdot \sum_{t=1}^{n_{i,p}} \left\| X_j^{i,p} - c_{i,p} \right\|^2} \quad . \tag{3}$$

Where $c_{i,p}$ is the center of the ith cluster in the pth class, and $\sigma_{i,p}$ is the standard deviation of the samples in this cluster to its center.

STEP 2: Select any cluster A in a class. Search the nearest cluster B to A within the class. They hence determine a potential merge to a new cluster C with the new center and standard deviation: $c_{AB,p}$, $\sigma_{AB,p}$.

Next steps are designed to reject the merge in STEP 2 for those sub-clusters within each class which are around the borders between different classes.

STEP 3: For cluster C, search its nearest cluster D in other classes with the minimum distance between their centers: MINDIST.

STEP 4: If the inequality below is satisfied [4], simply accept the merge of A and B to C; or reject it so that A and B remain separated.

$$\text{MINDIST} > \lambda \cdot \sigma_{i,p} \quad . \tag{4}$$

Where λ is called clustering coefficient, which determines indirectly the number of sub-clusters. In [4], it is chosen to be 2. However, the value may be different for different cases. This should be test in experiments.

STEP 5: Repeat STEP 2 to Step 4 until all clusters are investigated. Actually, ambiguity exists here in [4]. That is, after a cycle ended, user may restart a new cycle in order to merge more clusters till the number of clusters doesn't change any more. That's reasonable; however, experiments show more cycles may rapidly decrease the number of clusters, which would subsequently affect the performance of the classifier. So, we fulfill just one cycle and quit clustering.

The ultimate sub-clusters determine the structure of the RBF hidden layer. Simply and reasonably choose the center of each sub-cluster as the center of a neuron while choosing the width of the radial function is a little more complex. The approach introduced in [6] is borrowed:

$$d_u = \underset{v,p \neq q}{mean} \left\{ \left\| c_{u,p} - c_{v,q} \right\| \right\} \qquad w_u = d_u \Big/ \sqrt{\left| \ln \gamma \right|} \quad . \tag{5}$$

The latter formula in equation (5) is derived from Gaussian function, that is to say, if another radial basis kernel is chosen, a modification should be correspondingly made to this equation. For the parameters, d_u is the median distance between the uth cluster, namely the uth center, to the clusters belonging to other classes, not to those in the same class because according to our clustering algorithm, usually the overlaps

between sub-clusters within one class are acceptable to be large [6]; γ is called overlapping coefficient which controls directly the overlaps between clusters. Much attention should be paid to the selection of a proper value for this coefficient, since too large overlaps may bring more misclassification while too small width leads to poor generalization of the network.

2.3 Weights Training and Classification

For each training sample, RBF gives an output while the known category information of this sample gives a target (ideal) one, so the weights training process just aim to minimize the mean error between the target and real outputs for all the samples:

$$\begin{aligned} &E = (\mathrm{T}-\mathrm{O})^{\mathrm{T}}(\mathrm{T}-\mathrm{O}) && O,T \in \mathbb{R}^{n\times C} \\ &O = H \cdot W && H \in \mathbb{R}^{n\times u}, W \in \mathbb{R}^{u\times C} \end{aligned} . \tag{6}$$

Where, T and O are respectively the target and real output matrix, n and C are respectively the number of samples and classes, H is the outputs of RBF hidden layer and W stands for the weights which is to be optimized. Traditionally, the linear least square (LLS) method is applied to solve this optimization problem, and the classical pseudo-inverse technique is also used in the matrix operation during the process.

$$W^{*} = H^{*} \cdot T \qquad H^{*} = \left(H^{T}H\right)^{-1} H^{T} . \tag{7}$$

Where W^{*} is the optimized weighting matrix, H^{*} is just the pseudo-inverse matrix of H. Unluckily, $H^{T}H$ is always singular or near singular, which will considerably reduce the accuracy. Singular value decomposition (SVD) is an effective tool to solve this problem:

$$\begin{aligned} &[U\ S\ V] = SVD(H) && \mathrm{H} = USV^{T} \quad UU^{T} = VV^{T} = I \\ &H^{*} = \left(H^{T}H\right)^{-1} H^{T} = \left[\left(USV^{T}\right)^{T} USV^{T}\right]^{-1} \left(USV^{T}\right)^{T} = VS^{*}U^{T} \end{aligned} . \tag{8}$$

The Classifier is finally constructed after obtaining the weights. Any test sample can be simply classified to the category that has the maximal component of outputs through the RBF network.

3 Experiments and Results

Lacking benchmark database just like those for face recognition research, in our experiments, we use the images extracted from some video clips that are photographed on the roof of a tall building roadside. The road in the scene is called Chengfu road, in Haidian district, Beijing city. Images are all transformed into gray-scale for consideration of computation reduction. Below are two samples indicating different traffic states, free and congested, respectively.

Fig. 3. Image Samples used in the Experiments

Additionally, these videos are not photographed in a short time, so illumination condition varies in different images; and the angle of view of the camera changes a little during the process, which makes the background composition in different images are not always the same. Both of these bring more difficulties to the experiments.

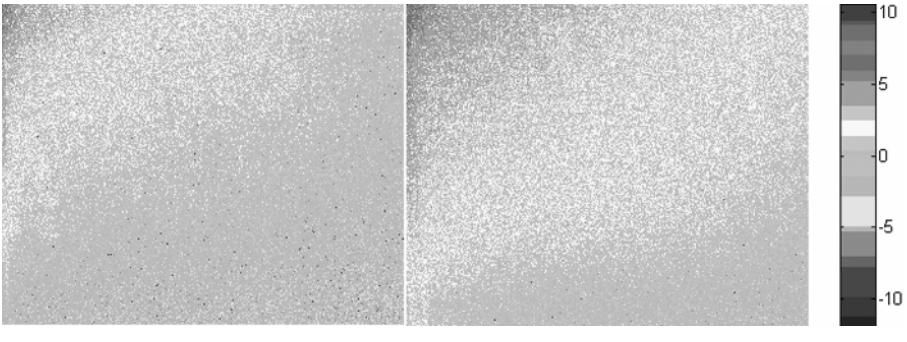


Fig. 4. DCT Comparison for Images with Different Traffic States

In **Fig. 4,** a comparison of DCT (before conversion from 2-D matrix to 1-D vector and the truncation operation) on two images with different traffic states is presented. The left one is for free-state with few cars in the scene and the right is for congested-state with many cars in the image. A logarithm conversion is added after DCT only for this color map demonstration (omitted in subsequent classifying experiments) in which relationships between numerical values and different colors are prompted by the rightmost color bar. So, the big difference between these two maps implies the applicability of DCT in this problem.

Below Experiments are organized according to the whole flow of the algorithm, and the results show its validness and efficiency as well as present comparisons of

performances for different values of the key parameters. The training set is composed of 150 free-state images and 349 congested-state images; while the testing set contains 73 free-states and 171 congested-states.

3.1 Performance of Truncated DCT

These results prove that properly discarding the low-frequency components of DCT coefficients can alleviate effect of brightness and illumination variations to some extent. In this case, truncating the lowest 2, called the DC and the first AC component, gives the best result while truncating more performs worse due to the loss of some important low-frequency information. However, the optimal length that should be truncated is not of course a fixed value, since how the frequencies represent the features of an image varies with different cases.

Table 1. Error Rates for Different Truncating Scheme

Number of Components Truncated	0	1	2	3	4
Error Rate	7.4%	6.6%	5.7%	8.6%	10.1%
Other Parameters	$\lambda = 10,\ \gamma = 0.1,\ length\_feature = 10$				

3.2 Variations in Feature Length

These results are obtained after truncating the first 2 low-frequency components. Longer length basically gives lower error rate, that's reasonable since more features contain more information that would be used for classification. Nevertheless, high-frequency DCT coefficients usually brings unstable factors to an image, even noise, which perhaps explains why the error rate for length being 30 is worse than the previous one, and also that's why we choose limited length as features starting from the low-frequency end, except for considerations of computation and low-magnitude of the high-frequency component.

Table 2. Error Rates for Different Feature Length

Length of Features	5	15	20	25	30	35	40
Error Rate	16%	5.6%	5.3%	4.1%	5.7%	4.1%	2.1%
Other Parameters	$\lambda = 10,\ \gamma = 0.1,\ length\_truncated = 2$						

3.3 Variations in Parameters

In **Table 3,** when clustering coefficient becomes larger, the cluster number gets bigger and error rate reduces, but the trend isn't so absolute and more neurons bring heavier computation.

Table 3. Error Rates for different Clustering Coefficients

λ	10	20	25	40	45
No. of Hidden Nodes	236	249	262	315	330
Error Rate	4.7%	5.3%	4.9%	2.5%	1.6%
Other Parameters	$\gamma = 0.3,\ length\_truncated = 2,\ length\_feature = 10$				

Table 4. Error Rates for different Overlapping Coefficients

γ	0.1	0.2	0.5	0.6	0.9	2	3	4
Error Rate	5.7%	4.5%	3.3%	2.1%	2.9%	3.3%	4.1%	4.5%
Other Parameters	$\gamma = 0.3,\ length\_truncated = 2,\ length\_feature = 10$							

In **Table 4**, the trend error rate varies with the overlapping coefficient reflects well the explanations in the end of part 2.2. For this case, 0.6 is a better choice.

3.4 Use PCA After Truncated DCT

Regarding the truncated DCT coefficients as the raw feature, we can apply PCA to transform the vectors to a lower-dimensionality space where the features may be easier to discriminate. Fixing the final dimensionality, experiments are made on different lengths of truncated DCT coefficients (the feature lengths before PCA):

Table 5. Error Rates for PCA after Truncated DCT

Length of Features before PCA	20	30	40
Error Rate	6.97%	6.14%	5.74%
Other Parameters	$\lambda = 10,\ length\_truncated = 2$ $\gamma = 0.3,\ length\_features = 10$		

Apparently, the longer the raw feature, the better the result, that's easy to understand. However, compared with the error rate obtained in the foregoing experiments without PCA, for the same feature length, 4.7%, these results are worse; and the same situation happens when the feature length changes to 5. That is to say, using PCA after DCT is not as good as what we think intuitively. Reasons may like what we have addressed in part **2.1**.

4 Conclusions and Future Work

This paper proposes a novel idea for vision-based traffic state recognition. Experiments show that the mature processing architecture for face recognition is efficient in

this application. Composed of more or less separated and different objects, traffic images have distinct features on the frequency spectrum, which can be efficiently extracted by DCT; in succession RBF network acts as a satisfying classifier to discriminate different traffic scenes. However, since big difference also exists between face recognition and traffic scene classification problem, some more modifications should be accordingly made to increase the accuracy. In other words, better and deeper analysis into image characteristics of traffic scenes may help improve the design of feature selection techniques and RBF training algorithms. Further, in order to fulfill online operation, more advancement must be done including online change of RBF structure, online optimization of weights, while more things should be taken into consideration, such as system memory demand and the system efficiency.

Acknowledgement

This work is supported by National Natural Science Foundation of China (NSFC) under the grant 60374059.

References

1. Benjamin, M., Osama, M., Nikolaos P.P.: Tracking All Traffic. IEEE Robotics and Automation (2005) 29-36
2. Hsu, W.L., Liao, H.Y.M., Jeng, B.S., Fan, K.C.: Real-time traffic parameter extraction using entropy. IEE Proceedings of Vision, Image and Signal Process (2004) 194-202
3. Fatih, P., Li, X.K.: Traffic Congestion Estimation Using HMM Models Without Vehicle Tracking. IEEE Intelligent Vehicles Symposium (2004) 188-193
4. Yang, F., Michel, P.: Implementation of an RBF Neural Network on Embedded Systems: Real-Time Face Tracking and Identity Verification. IEEE Transactions on Neural Networks 14(5) (2003) 1162-1175
5. Pan, Z.J., Alistair, G.R., Hamid, B.: Image Redundancy Reduction for Neural Network Classification using Discrete Cosine Transforms. Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (2000) 149-154
6. Meng, J.E., Chen, W.L., Wu, S.Q.: High-Speed Face Recognition Based on Discrete Cosine Transform and RBF Neural Networks. IEEE Transactions on Neural Networks 16(3) (2005) 679-690
7. Smith, R.M., Johansen, T.A.: Local Learning in Local Model Networks. IEEE International Conference on Artificial Neural Networks (1995) 40-46
8. Tarassenko, L., Roberts, S.: Supervised and Unsupervised Learning in Radial Basis Function Classifiers. IEE Proceedings of Vision, Image and Signal Processing (1994) 210-216

A New Method for Traffic Signs Classification Using Probabilistic Neural Networks

Hang Zhang and Dayong Luo

School of Information Science & Engineering, Central South University, Changsha, Hunan 410075, China
zhang22@mail.csu.edu.cn

Abstract. Traffic signs can provide drivers with very valuable information about the road, in order to make driving safer and easier. In recent years, traffic signs recognition has aroused wide interests in many scholars. It has two main parts-- the detection and the classification. This paper presents a new method for traffic signs classification based on probabilistic neural networks (PNN) and Tchebichef moment invariants. It has two hierarchies: the first hierarchy classifier can coarsely classify the input image into one of indicative signs, warning signs or prohibitive signs according to its background color threshold; the second hierarchy classifiers including of three PNN networks can concretely identify traffic sign. The inputs of every PNN use the new developed Tchebichef moment invariants. The simulation results show that the two-hierarchy classifier can improve the classification ability meanwhile can use in real-time system.

1 Introduction

The Traffic Signs Recognition (TSR) is a field of computer vision research concerned with the automatic detection and classification of traffic signs in traffic scene images acquired from a moving car. Traffic signs carry a lot of very valuable information necessary for successful driving - they describe current traffic situation, define right-of-way, prohibit or permit certain directions, warn about risky factors etc. At the same time, traffic signs also help drivers with navigation. So it is important for drivers to make their driving safer and easier. In recent years, traffic signs recognition has aroused wide interests in many scholars. In the course of many research programs such as the European EUREKA project PROMETHEUS (PROgraM for European Traffic with Highest Efficiency and Unprecedented Safety) and the Intelligent Vehicles Highway System (IVHS) in the USA, much significant improvement has been made. TSR has two main parts-- the detection and the classification. The latter is studied in this paper. Many authors carried out studies in this area. Sompoch et al. [1] classify the traffic signs by a simple image matching method using the minimum of Euclidean distance between the target image and the template in traffic signs library. Based on the same idea, Betke et al. [2] apply the normalized correlation coefficient as a measure of the match. Estable et al. [3-4] developed a new classification method based on radial basis functions (RBFs). Paclik et al. [5-7]

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 33 – 39, 2006.

describe several classification methods combining multi-hierarchy classifier with BP networks to classify the color images or the gray-level images. By the light of the technique of the Cell Neural Network, Adorni et al. [8] give a CNN-based classification algorithm. Moreover, other methods such as SVM-based, semantic network- based and various integrated classification methods are mentioned as in [9-11]. But all the proposed algorithms up to now are far from the real-time implementation in the recognition ability and the running time. Traffic signs are easily identified by human drivers mainly because their color and shapes are very different from natural environments. For example, the warning signs have a yellow background and are surrounded by a black triangle border. Prohibition signs are circles with a black background and a red border. Indicative signs are circles with a blue background. Taking advantage of these features, this paper presents a new two-hierarchy classifier for traffic signs classification based on PNN and Tchebichef moment invariants: the first hierarchy classifier is used to coarsely classify the input image into one of the indicative signs, warning signs and prohibitive signs; the second hierarchy classifiers including of three PNN networks are designed to concretely identify each traffic sign. The inputs of every PNN use the new developed Tchebichef moment invariants. The simulation results show that the new method can improve the classification ability and can be used in real-time system. The outline of this paper is as follows. Section 2 describes the architecture of our algorithm. Section 3 discusses the two different classifiers: the first hierarchy classifier which uses color threshold to coarsely classify the input image into one of indicative signs, warning signs or prohibitive signs; the second hierarchy classifiers including three PNN networks are designed to concretely identify traffic signs. Section 4 presents simulation results showing the effectiveness of the new method. We conclude in Section 5.

2 The Architecture of the Proposed Algorithm

The past studies show that multi-classifier method using multi-hierarchy structure has several advantages to the single-classifier method. The first one is the reduction of the class count. Moreover, each particular classifier may exploit the most descriptive features for its task. Satisfactory classification results can be obtained in smaller number of features compared to single-classifier [6]. The traffic signs classification is a large-scale and complicated subject. So we can decompose the process of classification into two steps. At the beginning, indicative signs, warning signs and prohibitive signs are split apart in the first hierarchy classifier (labeled as classifier 1) according to the embodied special color information. Then traffic sign is identified as a concrete sign through the second hierarchy classifiers. The architecture of the proposed classification algorithm is shown as Fig.1. The second hierarchy classifiers including three PNN classifiers are labeled as PNN1, PNN2, PNN3 respectively. During the training, test traffic sign is firstly fed into classifier 1 and identified as one of the three standard traffic sign groups. Then the result of the coarse classification is input into the corresponding PNN classifier, the traffic sign with concrete meaning is obtained at the end.

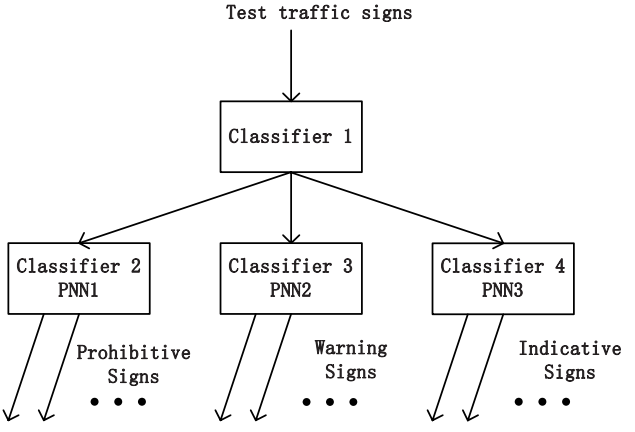


Fig. 1. The architecture of the proposed algorithm

3 The Proposed Classification Algorithm

The design of classifiers plays the important role in our classification algorithm. Classifier 1 and classifier 2-4 describe as followings:

3.1 Classifier Based on Color Threshold

Color information may use as a useful attribute for the decomposition of classification problem into several a priori defined traffic sign groups or sub-problems. In design of the Classifier 1, we use the color features to carry out the coarse classification. Before the classification, image segmentation must be processed. The simplest and most intuitive segmentation method is in RGB color space using the color threshold [12]. But one of the greatest inconveniences of the RGB color space is that it is very sensitive to varying illumination. In HSV color space, the hue component is non-correlated from the intensity component. Based on the Liu's works [13], in the process of the color segmentation in this paper, we firstly transform the color space model RGB to HSV, the segment the image according to proper thresholds on the hue and saturation which calculate the sum of the red, blue and yellow color pixels respectively. This approach allows outdoor light variations. Every color threshold can be determined by experiments. In order to coarsely classify indicative signs, warning signs and prohibitive signs, we apply the rate of the umber of the color pixels to the total number of the image pixels (RCT). RCT is defined as follows: $RCT_{COLOR} = N_{COLOR} / N_{TOTAL}$, Where N_{COLOR} equals to the number of the color pixels; N_{TOTAL} equals to the total number of the image pixels. After calculation of the RCT of the red, blue and yellow, we can simply use following rules in classification. The values of R_1, R_2, R_3 are selected through experiments.

Rule1: if $RCT_{RED} \geq R_1$, then the test traffic sign is classified as prohibitive sign group.

Rule2: if $RCT_{BLUE} \geq R_2$, then the test traffic sign is classified as warning sign group.

Rule3: if $RCT_{YELLOW} \geq R_3$, then the test traffic sign is classified as indicative sign group.

Rule4: if $RCT_{RED} < R_1$ and $RCT_{BLUE} < R_2$ and $RCT_{YELLOW} < R_3$, then the test traffic sign is not the traffic signs.

3.2 Classifier Based on PNN

Before the design of the second hierarchy classifiers, we must extract the inner geometry shapes of the test traffic signs by the use of the special color information. Then transform the extracted image to the black and white images. Next, extract the feature based on Tchebichef moment invariants. At last, classify the images using PNN classifiers.

3.2.1 Feature Vector Construction Using Tchebichef Moment Invariants

Image moments that are invariant with respect to the transformations of scale, translation, and rotation find a variety of applications in image analysis, like pattern recognition, object classification, template matching, etc. In 1961, Hu[14] introduced moment invariants. However, Hu's moments are not orthogonal and as a consequence, there is redundancy measure in a feature set. Teague[15] proposed continuous orthogonal moments such as Zernike and Legendre moments, but one main problem with these moments is the discretization error, and hence limits the accuracy of the computed moments. Mukundan [16] developed a new discrete orthogonal moment functions based on Tchebichef polynomials. On the basis of their works, Tchebichef moment invariants are presented in this paper and are used as feature vector for classification.

For an image intensity distribution $f(x,y)$ with $N \times N$ pixels, we define Tchebichef moment invariants, $\overline{T}_{mn}$ as:

$$\overline{T}_{mn} = \frac{1}{\tilde{\rho}(m,N)\tilde{\rho}(n,N)} \sum_{i=0}^{m} \sum_{j=0}^{n} c_{i,m,N} c_{j,n,N} \overline{M}_{ij} \tag{1}$$

$$\overline{M}_{nm} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \alpha^2 f(x,y) \times \left\{ [(x - \bar{x})\cos\theta + (y - \bar{y})\sin\theta] \times \alpha + \frac{N}{2} \right\}^n \times \left\{ [(y - \bar{y})\cos\theta - (x - \bar{x})\sin\theta] \times \alpha + \frac{N}{2} \right\}^m \tag{2}$$

$$\tilde{\rho}(n,N) = (N(1-1/N^2)(1-2^2/N^2)\cdots(1-n^2/N^2))/(2n+1) \qquad n = 0,1,2,\cdots N-1 \tag{3}$$

$$\begin{aligned}
&C_{0,0,N} = 1; \\
&C_{0,1,N} = \frac{1-N}{N}; C_{1,1,N} = \frac{2}{N}; \\
&C_{0,2,N} = \frac{N^2 - 3N + 2}{N^2}; C_{1,2,N} = \frac{6(1-N)}{N^2}; C_{2,2,N} = \frac{6}{N^2}; \\
&C_{0,3,N} = \frac{-N^3 + 6N^2 - 11N + 6}{N^3}; C_{1,3,N} = \frac{12N^2 - 30N + 22}{N^3}; \\
&C_{2,3,N} = \frac{30(1-N)}{N^3}; C_{3,3,N} = \frac{20}{N^3};
\end{aligned} \tag{4}$$

$\bar{x} = M_{10} / M_{00}, \bar{y} = M_{01} / M_{00}$, $\alpha = (\beta / M_{00})^{1/2}$, $\beta(n, N) = N^n / 4$, M_{ij} is the geometric moment. $\bar{T}_{mn}$ is invariant with respect to the transformations of scale, translation, and rotation. In this paper, we construct the feature vector as $\mathbf{v} = [\bar{T}_{20}, \bar{T}_{02}, \bar{T}_{12}, \bar{T}_{21}, \bar{T}_{30}, \bar{T}_{03}]$.

3.2.2 Classification Using PNN

For traffic signs classification, it is a supervised learning classification question to a large training set. So PNN is selected as the second hierarchy classifier in our system. PNN adopts multivariate Parzen window estimators to estimate the probability density functions (PDF) of the different classes [17]. The most obvious advantage of the PNN is that its training time is much shorter than that of popular BP network. Other characteristics of this network include: 1) the decision surfaces can approach Bayes-optimal. 2) it operates completely in parallel without a need for feedback from the individual neurous back to the inputs and can be easily used in hardware. The PNN has four layers or input layer, pattern layer, summation layer and output layer.Let $x \in R^d$ be a d-dimensional pattern vector and its associated class be $j \in \{1,2,\cdots K\}$, where K is the number of possible classes. The first layer of the PNN is the input layer which accepts input patterns. The nodes in the second layer are divided into K groups, one for each class. The generic second-layer node, the ith kernel in the jth group, is defined as a Gaussian basis function:

$$p_{ij}(V) = \exp(-\|V - C_{ij}\|^2 / 2\varsigma_{ij}) / (2\pi\sigma_{ij}^2)^{d/2} \tag{5}$$

where C_{ij} is the centre or mean vector and σ_{ij}^2 is a positive variance parameter. C_{ij} and σ_{ij}^2 can be choose in the training. Let the number of pattern units for class j be N_j. Then the total number of the pattern layer nodes is $N = \sum_{j=1}^{K} N_j$. The summation layer has K nodes, and each node estimates a class conditional PDF using a mixture of Gaussian kernels : $f_j(V) = \sum_{i=1}^{N_J} \beta_{IJ} p_{ij}(V), \quad i \le j \le K$ where the positive mixing coefficients β_{ij} satisfy: $\sum_{i=1}^{N_j} \beta_{ij} = 1, \quad 1 \le j \le K$;The output layer of the PNN makes the decision according to $g(V) = \arg(\max_{1 \le j \le K} \{\alpha_J f_j(V)\})$, where α_j is the a priori probability of class j.

4 Simulation Experiment

Training set: In our training set, there are 125 standard pictograms from the national standard traffic signs library in China. 1-42#, 43-102# and 103-125# pictograms are prohibitive signs group, warning signs group and indicative signs group respectively. Each pictogram is a 24-bit BMP color image with 60*60 pixels.

Test set: One parts of test set are achieved through adding Gaussian white noise to the correspond training set or rotating, translating, scaling the pictograms in correspond training set in MATLAB platform. With noise variance $D = 1\%, 2\%, 3\%$,

Rotating angles, $\Phi = 30^\circ, 15^\circ, -15^\circ$ and -30° and scaling factors, $S = 0.8, 0.9, 1.0$. Other parts of test set including 100 images are coming from the actual images.

In our experiment, classification accuracy η is defined as

$$\eta = \frac{Number\ of\ correctly\ classified\ images}{The\ total\ number\ of\ images\ used\ in\ the\ test}$$

In this paper, we compared our method with that used by reference 5-7. The average running time is less than 1s in our method. Table 1 shows the results of the classification. It can be seen that our algorithm shows the good robustness and classification accuracy and can be applied in real-time.

Table 1. Classification results

Noise variance D	1%	2%	3%
η (This paper)	100%	94.6%	90.1%
η (Reference 5-7)	80.4%	74.6%	50.5%

5 Conclusions

In this paper, a new method for traffic signs classification based on PNN is presented. It has two-hierarchy structure: the first hierarchy classifier which uses color threshold in HSV color space is used to coarsely classify the images into indicative signs, warning signs and prohibitive signs; the second hierarchy classifiers including of three PNN networks are designed to concretely identify each traffic signs with the help of the new developed feature extract method. The simulation results show that this method is an efficient one and can be used in real-time system.

Acknowledgement

This work is supported by Hunan Provincial Natural Science Foundation of China 05JJ30121 and the Innovation project in graduate student's education by Central South University 040122.

References

1. Sompoch, P.,Xiaoyangv, C., Michiro, K.: Automatic Recognition and Location of Road Signs from Terresterial Color Imagery. Geoinfomatics&DMGIS'2001, Bangkok, Thailand (2001) 238–247
2. Betke, M., Makris, N.C.: Fast Object Recognition in Noisy Images Using Simulated Annealing. Proceedings of the IEEE International Conference on Computer Vision, Cambridge, USA(1995) 523–530
3. Estable, S., Schick, J., Stein, F., Janssen, R., Ritter, R., Ott, W., Zheng, J.: Real-time Traffic Sign Recognition System. Proceedings of the Intelligent Vehicle'94 Symposium, Paris, France (1994)213–218

4. Gavrila, D.M.: Traffic Sign Recognition Revisited. Proceedings of the 21st DAGM Symposium, Bonn, Germany(1999) 86–93
5. Paclik, P., Novovicova, J., Pudil, P., Somol, P.: Road Sign Classification Using Laplace Kernel Classifier. Pattern Recognition Letters. 21(13~14) (2000) 1165–1173
6. Paclik, P., Novovicova, J.: Road Sign Classification without Colour Information. Proceedings of 6th Conference of Advanced School of Imaging and Computing, ASCI, Lommel, Belgium(2000)
7. Zhu, S.: Two Hierarchy Classifier for Recognition of Traffic Signs Based on Neural Network. Proceedings of the 5th World Congress on Intelligent Control and Automation, Hangzhou, China(2004)5302–5306
8. Adorni, G., Andrea, D., Destri, G., Mordoni, M.: Shape Searching in Real World Images: a CNN-based Approach. Proceedings of the IEEE International Workshop on Cellular Neural Networks and their Applications, Seville, Spain(1996)213–218
9. Dayan, G., Hait, N.: Road Sign Recognition Project Based on SVM Classification. http://visl.technion.ac.il/projects/2003w10
10. Bessere, B., Estable, S., Ulmer, B., Reichardt, D.: Shape Classification for Traffic Sign Recognition. Proceedings 1st IFAC International Workshop Intelligent Autonomous Vehicles(1993) 487–492
11. Sueteus, P., Fua, P., Hanson, A.J.: Computational Strategies for Object Recognition. ACM Computing Surveys. 24(1) (1992) 5–61
12. de la Escalera, A., Armingol, J.M., Salchs, M.A., Armingol, J.M.: Road Traffic Sign Detection and Classification. IEEE Transactions on Industrial Electronics. 44(6) (1997) 848–859
13. Liu, H.X., Ran, B.: Vision-based Stop Sign Detection and Recognition System for Intelligent Vehicle.Transportation Research Record. 1748 (2001) 161–166
14. Hu, M.K.: Visual Pattern Recognition by Moment Invariants. IRE Transactions on Information Theory. IT(8) (1962) 179–187
15. Teague, M.R.: Image Analysis Via the General Theory of Moments. Journal of Optical Society of American. 70(8) (1980) 920–930
16. Mukundan, R., Ong, S.H., Lee, P.A.: Image Analysis by Tchebichef Moments. IEEE Transactions on Image Processing. 10(9) (2001) 1357–1364
17. Specht, D.: Probabilistic Neural Network. Neural Networks. 3 (1) (1990) 109–118

Detection for Triangle Traffic Sign Based on Neural Network

Shuang-dong Zhu[1], Yi Zhang[2], and Xiao-feng Lu[1]

[1] College of Information Science and Technology, Ningbo University, China
[2] Kristianstad University, Kristianstad 291 88, Sweden
Zhushuangdong@nbu.edu.cn

Abstract. This literature critically explains the intelligent method for detection of traffic signs. This method uses a particular color and shape for the detection of traffic signs, as an example, we used red color down triangle shape traffic sign, to explain this method. This method is mainly carried out in four steps, which are as follows. First, convert RGB color space to HIS color space, and extract pixels with red color. Then perform LOG mask operation on the pixels got from step 1, for the detection of edges. By using neural network, we determine the angle pixels, and at the same time, we also determine on which specific angle the pixel is. And finally we detect the traffic sign by using the information of shape. We used 20 different images from different scenes to test this method, and the percentage of correctness is 100%.

1 Introduction

In the past two decades with the development of economy, traffic network system is not able to fulfil the needs of traffic loads which are increasing very fast. The increasing traffic jam, pollution and traffic accidents had divert the attention of international society, due to which the 'Intelligent Transportation System' (ITS) this new research came in to the picture or occurred and also growing very fast. Many people are working under this area. Intelligent Transportation Management is very important branch in ITS, the main method in this branch is 'image detection'. The image/visual information is very important in traffic scene for the onboard system, which is placed inside the car. To identify traffic sign efficiently is very important and also unsolved problem in ITS research area. Traffic sign recognition (TSR) system mainly consists of two basic steps viz; first step is the traffic sign location with some necessary pre produce and second step is to do identification. This paper mainly discusses the first step.

In the real world, traffic sign identification has several main difficulties such as, first is the complexity of the roads, leads to the complex background of the image. The traffic sign can even been faintness or been covered from other object. Second, is different condition of illumination, this leads to serious colour distortion. Third, is shape and scale distortion because of the angle of shoot that can cause geometric distortion, sometimes it is very serious. Based upon the above points, to use intelligent method is better choice to solve the TSR problem. In fact from the time of

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 40 – 45, 2006.

evolution the pattern recognition is an intelligent activity of human; because of this in future machine recognition the intelligent method will domain this area. Neural Network has advantages of nonlinear mapping; tolerate ability, and soft network structure. These makes neural network has very good appliance value in areas like pattern matching, pattern classification, pattern recognition and analysis. This study introduces a method with neural network with the traditional method.

2 Intelligent Detection Method

The traffic sign of all countries have national standard, they have the special shape, colour, sign and size, comparison, in order to identify.

Fig.1.(a), is road transportation for physically taking, among them have four traffic signs, from left to right is: one red colour triangle traffic signs and three circular traffic signs that have red frames. Region of interests is the key technique of the fast image resemble. This research in the Fig.1.(a) shows pours the triangle traffic sign, for example, and shows a new intelligence examination method. The method consists of four following main points:

1. The color extract: According to the traffic sign have the particular colour, resemble the image to convert the hue subspace first, extract the particular red colour, and fix the Region of Interest (ROI) region.
2. The Edge Extract: The red colour region of template withdraws out of standard LOG of adoption go forwards the line, and further contracts the interested region.
3. The top distinguish adoption neural network classification to all point those three different tops that ROI proceeds to distinguish, and make the marking to triangle respectively.
4. Fix the position of the triangle. Finally, use certain of matching standard to extract triangle.

Fig. 1. (a) Origin color real image

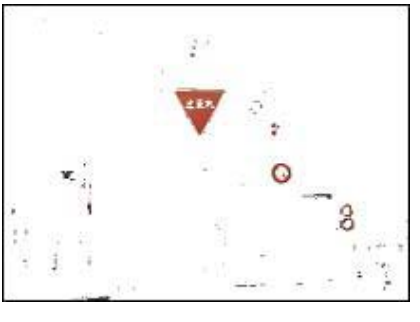


Fig. 1. (b) Red points of H value (color)

3 The Color Extract

The colour extract is on H (hue), the subspace of HSI (hue, saturation and intension). The image format should be in RGB and so there is need to convert HSI space to RGB colour space and the conversion formula used for this research is as follows [4]:

$$r = \frac{R}{255},\ g = \frac{G}{255},\ b = \frac{B}{255}, \qquad \text{(unitary)}$$

$$I = \frac{r+g+b}{3}$$

$$H = 90 - \arctan\left(\frac{F}{\sqrt{3}}\right) \times \frac{180}{\pi} + \{0, g > b; 180, g < b\} \tag{1}$$

where $$F = \frac{2r-g-b}{g-b}$$

$$S = 1 - \frac{\min(r,g,b)}{I}$$

Colour distribution of HSI space is as shown in figure.2, so according to figure.2 shows, we can extract colour region according to our need. The example demands to extract the red colour region. The H value is between [315,360] and [0, 23]; fig.1(a) is a scene image for 320×240 pixels, by using above method we extract red colour region from the figure.1.(a), results are shown in figure.1.(b), besides above 4 traffic signs rest dark region is mainly of the object of the red hue (though we may not see red in colourful image). This is the fringe ROI region.

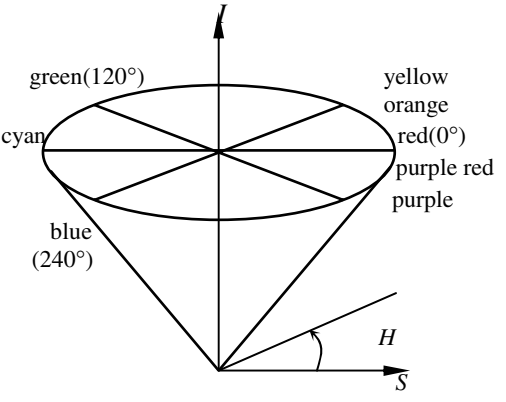


Fig. 2. HSI color space model delineation

4 The Edge Extract

The purpose of edge extraction is to decrease pixel number in the ROI, in order to availability and to reduce the workload of following process. Edge extraction uses LOG (Lapacian of guassian) calculation method. On the ash degree chart log calculates using gauss function first to make smooth handle for the image, then use no direction Lapacian to extract zero crossing point and make the edge examination. The formula of the log calculates and the 5×5 template use in this research as follows.

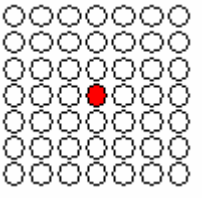


Fig. 3. Input layer

Among them r2=x2+y2, x, y is the abscissa value and horizontal value of the pixel. Due to the colors distill part has already distilled the red ROI area, the left dots of a real picture has reduced a lot. So, edge distill can be only implemented to the red ROI area, the time it should take will be reduced a lot. The left dots with edge distill reduce obviously, they all have the traffic sign colors and all are on the edge. These dots are those ROI dots which will be dealt with by NN, viz. which might be vertexes.

5 Vertex Distinguish

Vertex distinguish use the intelligent sorting which constructed by single latent BP network, the swatches of training setting and testing setting are all ash picture. Input layer is a scan window constituted with a 7×7 matrix, as Figure 3. This scan window is used to scan interested ROT dots in the scene picture of the testing settings, so that the three vertex of the triangle can be fixed. It has 49 input units each corresponding to a pixel, the center unit (the red unit in figure 3) corresponding to ROI dots. Uses a big scan window in order to insure the correctness of distinguishes. The output layers include 3 units, each stand for the bottom, left top, right top vertex of the triangle. Training setting are constituted with 4 kinds of swatches: bottom vertex, output is 1,0,0; left top vertex, output is 0,1,0; left top vertex, output is 0,0,1; yawp dots (here means any dots except the triangle vertexes), output is 0,0,0. The units number of the latent layer can adjust according to the real situation, the research use 9 latent units.

As we all know, when we using the NN way, the choosing of the training setting swatches is very important. First, we should choose enough swatches of these three vertexes, very sure every swatch is 7×7 matrixes and vertex is in the center. Second, choosing enough yawp dots swatches is also necessary. So that the resolving power can be increased greatly. Testing setting is some real traffic scene pictures. Use the fixed training setting to train the NN Before checking. When checking, use gained training results to scan and sort with every ROI dot of every picture in testing setting, in order to check out the correct three vertexes of the triangle. At last, give different signs to the different vertexes in order to be propitious to the following disposal. Till now, the left ROI dots are very few.

6 Orientation of Triangle

Athough the needed ROI dots have been checked out, BP network may take some yawp dots as vertexes. This part is to wipe off the mistake dots and distill the triangle structured by correct vertex. The detailed method is shown in figure.4 as follow;

(1) First, according the vertex signs which made in the vertex distinguish part to fix a bottom vertex (the top picture in figure.3).

(2) Second, in the pie slice area started from the bottom vertex, according the vertex sign made in vertex distinguish part to select left top vertex. If one left top vertex is found, then continue to implement step (3); if none left top vertex is found, it means current bottom vertex is a yawp dot, here it should be deleted, the come back to step (1) and go on searching other bottom vertexes.

(3) Then, in the superposition area of the pie slice areas which start from the fixed bottom vertex and left top vertex, according to the vertex sign signed in the vertex distinguish part to search a right top vertex. If it is found, continue to implement step (4) to distinguish the triangle's vertexes; if none of the right top vertex is found, it means the current left top vertex is a yawp dot, here should be deleted, then come back to step (2) continue searching other left top vertexes. After the left top vertex (x1, y1) is fixed, the scan range of superposition of two pie slice areas which search right top vertex (x2, y2) is decided by formula (2):

(4) At last, decide whether these found three dots are the three vertexes of the triangle: if they are, end the orientation task; otherwise it means the current right top vertex is a yawp dot, then should delete this yawp dot, and come back to step (3) continue searching other right top vertexes.

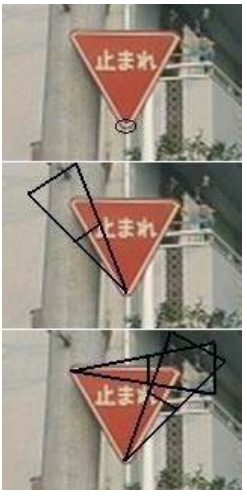


Fig. 4. Location delineation

The basic fundamental which used to judge whether these found three dots are the triangle's three vertexes is the three sides of a equilateral triangle is equal (correct to say, the standard figure of the converse triangle traffic sign in this example is a isosceles triangle closes to a equilateral triangle). But, the real took traffic sign figure usually will be varied degree geometric distortion. So, the matching rule advanced in this research is based on the equal three sides to give out a side ratio range [1/1r~1r]. When judging, first ratio each side of the three sides with other two sides, get three sides ratio, then judge whether the three ratios are all in the side ratio range. According to law of sines, consider that usually$\alpha \leq 20$, then the range of the above side ratio range [1/1r~1r] can be decided by formula (3) in the formula, k is the adjust coefficient, mostly can selected in the range [0.6~1]. When k = 1, it is the limit case, viz. the utmost side ratio when the excursion angle equal to, scilicet the case when the traffic sign may be utmost distortion. The parameters in this investigation isα = 15, k = 0.8. In this way, lr = 1.6, accordingly, the side ratio range in the above matching rule is [0.63~1.6].

From the above several steps, these three triangle vertexes can be fixed on and distill triangle traffic sign correctly and that yawp dot will be deleted for it can't accord the restriction qualification in the above steps. This kind of orientation method not only obviously reduced computation amount (only select ROI in pie slices), but also wipe off the misjudged dots effectively and allow the object exist geometric distortion in certain degree.

7 Experiment Result and Analysis

This research does a lot of resemble experiment, the three basic experiments as follows: (1) The first experiment is to examine the image in the same background. Among them, BP network use 50 sample that from 7 image to train as the training gather of BP network, then use above intelligence method to 14 images (among them 7 have once used for training) colour extract, edge examine triangle distinguish, then

all triangle traffic sign can be examined correctly. (2) The second experiment is to examine 20 images of different background. First use BP nerve network that have 50 samples training gather to distinguish the top, the result shows only 9 images can be examined correctly. To make sure that method can apply in the different background, we must increase to train the sample that gather. Then increase the number of the training sample. So we increase the number of training sample to repeat the above experiment. When the number of the training sample increase to 100, triangle traffic sign in 20 images all can be examined correctly, and have no the miscarriage of justice, the examination correctness come to a 100%. (3) The third experiment uses that method to the image that has more than one traffic sign .The result of emulation experiment shows the triangle traffic sign in the image can be examined accurately and have no the miscarriage of justice's circumstance.

From above experiment we can get the basic following conclusion about this intelligent method: (1) Use the nerve network examination triangle's top to extract the triangle traffic sign is available. This method adopt off-line training BP network to identify the three top of the traffic sign, make the TSR system have good adaptability to size change and shape lose. (2) The method adopt the HSI colour space transformation, use the H (hue) subspace to color extract make the TSR system have the good adaptability to road shine and color lose. (3) The method makes best use of the ROI concept, and is quicker, can meet the demand of traffic sign real time examination.

8 Future Scope

The TSR research area, people have already found a lot of non-intelligence methods, but these methods all have their localization and they are always sensitive to geometric distortion of and colour distortion of traffic sign. But make use of the BP network non-linear classification ability to examine traffic sign, and can make the TSR system have good to geometric distortion and colour distortion. Intelligent examination, this research refer the method make use of the certain ROI of colour image, the top distinguish; fix the position of the triangle. All process aches have good real time quality. But that method can't apply in the withdrawing of circular traffic sign, not yet in consideration of the traffic sign that is hidden.

References

1. Zhu, S.: Image Detection and Processing for ITS, Proceedings of 2000 Chinese Conference on Measurement, Beijing, China (2000) 512-516
2. Zhu, S.: Two Hierarchy Classifier for Recognition of Traffic Signs Based on Neural Network, Fifth World Congress on Intelligent Control and Automation(WCICA2004), Conference Proceedings of WCICA2004, Vol. 6, Hangzhou, China (2004) 5302-5306
3. Zhu, S.: The Classification of Traffic Sign Base on Fuzzy Characteristics Training set, Fifth World Congress on Intelligent Control and Automation (WCICA2004), Conference Proceedings of WCICA2004, Vol.6, Hangzhou, China(2004) 5266-5270
4. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. 2nd edn. Addison-Wesley (2002)
5. Marr, D., Hildreth, E: Theory of Edge Detection. Proc, R Sol, Long 207 (1980) 187-217

Vanishing Point and Gabor Feature Based Multi-resolution On-Road Vehicle Detection

Hong Cheng[1], Nanning Zheng[1], Chong Sun[1], and Huub van de Wetering[1,2]

[1] Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, China
hcheng@mail.xjtu.edu.cn, nnzheng@mail.xjtu.edu.cn
[2] Technische Universiteit Eindhoven, Eindhoven, The Netherlands

Abstract. Robust and reliable vehicle detection is a challenging task under the conditions of variable size and distance, various weather and illumination, cluttered background, the relative motion between the host vehicle and background. In this paper we investigate real-time vehicle detection using machine vision for active safety in vehicle applications. The conventional search method of vehicle detection is a full search one using image pyramid,which processes the image patches in same way and costs same computing time, even for no vehicle region according to prior knowledge.

Our vehicle detection approach includes two basic phases. In the hypothesis generation phase, we determine the Regions of Interest (ROI) in an image according to lane vanishing points; furthermore, near, middle, and far ROIs, each with a different resolution, are extracted from the image. From the analysis of horizontal and vertical edges in the image, vehicle hypothesis lists are generated for each ROI. Finally, a hypothesis list for the whole image is obtained by combining these three lists. In the hypothesis validation phase, we propose a vehicle validation approach using Support Vector Machine (SVM) and Gabor feature.The experimental results show that the average right detection rate reach 90% and the average execution time is 30ms using a Pentium(R)4 CPU 2.4GHz.

1 Introduction

Statistics shows that about 60% of the rear-end crash accidents can be avoided if the driver has additional warning time. According to the Ministry of Public Safety P.R.China, there were 567,753 reported road traffic accidents in 2004, among those about 80% of the severe police-reported traffic accidents are vehicle-vehicle crashes. Clearly, vehicle detection is an important research area of intelligent transportation systems [1] [2]. It is being used in, among others, adaptive cruise control (ACC), driver assistance systems, etc. However, robust detection of vehicle in real-life traffic scenes is challenging.

This paper introduces a hypothesis-validation structure which consists of the three steps described below. Firstly, ROI determination using the vanishing point of lane markings. Secondly, vehicle hypothesis generation for each ROI using horizontal and vertical edge detection. Finally, hypothesis validation using Gabor features of 9 sub-windows and SVM classifiers.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 46–51, 2006.

2 Related Work

Hypotheses are generated using some simple features, such as, color, horizontal\vertical edges, symmetry[2][3], motion, and stereo visual cue. Zehang Sun proposed a multi-scale hypothesis method, in which the original image was downsampled. His vehicle hypotheses were generated by combining the horizontal and vertical edges of these three levels and this multi-scale method greatly reduced the random noise. This approach can generate multiple-hypothesis objects, but a near vehicle may prevent a far vehicle from being detected. As a result, it fails to generate the corresponding hypothesis of the far vehicle reducing the vehicle detection rate.

Vehicle symmetry is an important cue in vehicle detection and tracking. Inspired by the voting of Hough Transform, Yue Du proposed a vehicle following approach by finding the symmetry axis of a vehicle [3]; however, his approach has several limitations, such as large computing burden, and it only generates one object hypothesis using the best symmetry. A. Broggi introduced a multi-resolution vehicle detection approach, and proposed to divide the image into three fixed ROIs: one near the host car, one far from the host car, and one in the middle[2]. This approach overcomes the limit that it only can detect a single vehicle in the predefined region of the image but it needs to compute symmetry axis making it not real-time.

Accordingly, in this paper we detect vehicles only in ROIs allowing us to make a real-time implementation. The ROI approach largely prevents a near car from hiding a far car. All the hypotheses are generated in these regions. The positions of vehicles are validated by SVM classifiers and Gabor features.

3 Generating Candidate ROIs

Inspired by A.Broggio [2], we extract three ROIs, a near one, one in the middle, and far one, from the 640×480 image. But his approach uses fixed regions at the cost of flexibility. In this paper, ROIs are extracted using lane markings. In a structured lane, we detect the vanishing point using the lane edges. For the consideration of real-time processing, we use a simple vanishing point detector rather than a complex one. Discontinuity and noise related problems can be solved by combining, for instance, 10 subsequent images. In edge detection is done on combined images consisting of ten overlapping subsequent images, and the equations of two lanes are deduced from a voting procedures like hough transform by analyzing horizontal and vertical edges. Four random points $P_{di},\ d = l\ or\ r;\ i = 1,\ \cdots,\ 4$, are selected on each lane line, and each tangent direction of two points between the closed 3 points

$$\{P_{di}, P_{dj}\}; d = r\ or\ l; i, j \in \{0, 1, 2, 3\}; i < j; |j - i| \leq 2 \tag{1}$$

are obtained by

$$\theta_{dij} = \overrightarrow{P_{di}P_{dj}}$$

The tangent directions of two lane lines are calculated using average value of the above tangent angles, described by

$$\bar{\theta}_d = \frac{\theta_{d01} + \theta_{d02} + \theta_{d12} + \theta_{d13} + \theta_{d23}}{5}, d = r \ or \ l \tag{2}$$

Combining the average coordinates of 4 interesting points $\bar{P}_d = \frac{1}{4}\sum_{i=0}^{3} P_{di}$with the average tangent angles $\bar{\theta}_d$, we can get the equations of two lane lines. The intersection point of the two lines is the approximation of vanishing point, shown in Fig.1.

Next we consider how to extract ROI from the original image. For the consideration of vehicle height and the camera parameters, the top boundaries of all the ROIs are 10 pixels higher than the vertical coordinates of the vanishing point. From the analysis of the camera parameters and image resolution, the heights of the near, middle, and far ROIs are 160, 60, and 30 pixels, respectively. The left and right boundaries of the near ROI are those of the image. The distance between the left boundary of the middle ROI and that of the image is just one-third of the distance between vanishing point and the left boundary of the image, and the right one of middle ROI is determined similarly. The distance between the left boundary of the far ROI and that of image is two-thirds of the distance between vanishing point and the left boundary of image, as well as the distance between the right boundary of the far ROI and that of image.Fig. 1(b) shows the results of each ROI.

(a) Linking Lines between Interesting Points (b) Division of ROI (c) Lane ROI Generation

Fig. 1. Vanishing Point and ROI Generation

4 Multi-resolution Vehicle Hypothesis

For traditional approaches, edges of small object cover those of a large one. The results of a global histogram of horizontal and vertical edges show the edge histogram without a peak for the small vehicle. Based on the preceding candidate regions, the histogram of a ROI shows a peak for a small object.The analysis of the peaks and valleys of an edge histogram results in several rectangles and each one represents a vehicle hypothesis. We use prior knowledge to eliminate some hypotheses. The minimum width of a vehicle can be set for each ROI. If the width

of a hypothesis is smaller than this width, the hypothesis will be eliminated. Additionally, the aspect ratio (width/height) of vehicles is in a certain range, we assume in the range $[0.67, 2.2]$; rectangles with other ratios are eliminated. Since the histogram is made by extracting edges from the ROIs and other objects, like power cables and traffic signs above the road are not in the ROIs, they do not disturb the edge histogram, reducing the false positive rate. The coordinates of all hypothesis objects will be translated into the coordinates of the original image, and then the hypotheses of different ROIs may be overlapping. According to the distance between two rectangles, $d(r_1, r_2)$, we can judge if the two rectangles ought to be incorporated into one. Equ. 3 defines the distance between rectangle r_1 and r_2, and here (x_{ij}, y_{ij}) is the coordinates of the j-th vertex of i-th rectangle, i=1,2, j=1,2,3,4. Through the above process,we finish the generation of vehicle hypotheses.

$$d(r_1, r_2) = \| r_1 - r_2 \|_2, \quad r_i = (x_{i1}, y_{i1}, x_{i2}, y_{i2}, x_{i3}, y_{i3}, x_{i4}, y_{i4}), i = 1, 2. \tag{3}$$

5 Vehicle Validation Using Gabor Feature and SVM

5.1 Vehicle Representation

We first introduce some necessary definitions for Gabor filters and basic concepts for vehicle representation. The 2D Gabor function can be defined as follows

$$G_{\{f,\varphi,\sigma_u,\sigma_v\}}(u, v) = \frac{1}{2\pi\sigma_u\sigma_v} exp[-\frac{1}{2}(\frac{U^2}{\sigma_u^2} + \frac{V^2}{\sigma_v^2})] \cdot exp[2\pi j f U] \tag{4}$$

Where

$$\begin{cases} U = (u, v) \cdot (cos\varphi, sin\varphi) \\ V = (-u, v)(sin\varphi, cos\varphi) \end{cases}$$

And f means the normalized spatial frequency of a complex sinusoidal signal modulating Gaussian function, φ is the direction of a Gabor filter, σ_u and σ_v are the scale parameters of filter. Therefore, $\{f, \varphi, \sigma_u, \sigma_v\}$ can represent the parameters of a Gabor filter. Actually, a Gabor filter is a bandpass filter, and the first step of vehicle detection is to select the Gabor filters strongly responding to the detected object.

We select the filter parameters with the strongest response for certain sub-window including a vehicle part, and use SVM as a performance estimation classifier. The test image is divided into 9 overlapping sub-windows, and the statistical Gabor features from convolution between sub-window image patch and a Gabor filter, mean μ, standard deviation θ, and the skewness κ, represent the vehicle [6]. We optimize the SVM parameters for each of the nine sub-windows, and test the resulting nine classifiers for each sub-window using test examples, and record the average error rate. At last, for each sub-window, the 4 Gabor filters with the minimum average error rate are combined into a filter bank for extracting a feature vector (See Table 1).

Table 1. Selection of the Optimized Gabor Features

(1).To give the test error rate for m'th sub-window by $(x_i, y_i)_{i=0}^{N}$,where x_i is the parameter vector,y_i is the error rate;$Y_0 = \{y_0, y_1, \ldots, y_N\}$.
$P_0 = \{\}$;
(2).To select the optimized filters
For t=0,1,2,3
Here:$index = argmin \parallel Y_t \parallel_\infty$
$Y_t = Y_t - \{y_{index}\}$
if $\parallel x_{index} - x_j \parallel > \epsilon, x_j \in P_t$
then $P_t = P_t + \{x_{max_t}\}$
else
goto Here
(3). To get the best Gabor filter bank for m'th sub-window.

Then the 9 sub-windows with 4 Gabor filters each make a feature vector of size 108,

$$[\mu_{11},\ \theta_{11},\ \kappa_{11},\ \mu_{12},\ \theta_{12},\ \kappa_{12},\ \cdots,\ \mu_{93},\ \theta_{93},\ \kappa_{93},\ \mu_{94},\ \theta_{94},\ \kappa_{94}]$$

Where $\mu_{ij},\ \theta_{ij},\ \kappa_{ij}$, are the mean, standard deviation, and skewness; i is the number of a sub-window, j is the number of a filter.

6 Results and Discussions

6.1 Test Results and Discussions

For training the SVM classifiers, we selected 500 images from our vehicle database which was collected in Xi'an in 2005. They contain 1020 positive examples and 1020 negative examples. In testing the classifier, we get above 90% average right detection rate using 500 negative and positive examples independent of the training examples, and the missing and error detection rate is below 10%.

Fig. 2. Vehicle Detection Result Under the General and Hard Conditions

We tested our vehicle detector on the collected video using the Springrobot platform. Fig. 2 show the results of our vehicle detector under general and hard conditions. We proposed an approach for vehicle classification and detection with good time performance using vanishing points and ROIs while achieving high detection accuracy using Gabor feature. The method using the vanishing point to define ROIs, eliminates the disturbing effects of some non-vehicle objects improving both the detection rate and the robustness of this approach. The detection speed of our vehicle detector is approximately 20 frames/second on a Pentium(R) 4 CPU 2.4GHz both for general and hard conditions.

Extension of this approach to unstructured roads needs to be investigated. Further research work will focus on these problems.

Acknowledgements

The work is supported by the National Natural Science Foundation of China (60021302) and OMRON company from Japan for their support and assistance.

References

1. Zheng, N. N., Tang, S., Cheng, H., Li, Q., Lai, G., Wang, F. Y.: Toward Intelligent Driver-Assistance and Safety Warning System, IEEE Intelligent Systems 19(2) (2004) 8 - 11
2. Broggi, A., Cerri, P., Antonello, P. C.: Multi-Resolution Vehicle Detection Using Artificial Vision. In: IEEE International Symposium on Intelligent Vehicle. (2004) 310 - 314
3. Du, Y., Papanikolopoulos, N.P.: Real-Time Vehicle Following Through a Novel Symmetry-Based Approach. In: IEEE International Conference on Robotics and Automation, Vol. 4. (1997) 3160 - 3165
4. Rasmussen, C.: Grouping Dominant Orientations for Ill-Structured Road Following. In: Proceedings of the 2004 IEEE International Conference on Computer Vision and Pattern Recognition, Vol. 1. (2004) 470 - 477
5. Manjunath, B. and Ma, W.: Texture Features for Browsing and Retrieval of Image Data, IEEE Transactions on Pattern Analysis and Machine Intelligence 18(8) (1996) 837 - 842
6. Sun, Z. H., Miller, R., Bebis, G., and DiMeo, D.: On-road Vehicle Detection Using Evolutionary Gabor Filter Optimization, IEEE Transactions on Intelligent Transportation Systems 6(2) (2005) 125 - 137

Recognition of Road Signs with Mixture of Neural Networks and Arbitration Modules

Bogusław Cyganek

AGH - University of Science and Technology,
Al. Mickiewicza 30, 30-059 Kraków, Poland
cyganek@uci.agh.edu.pl

Abstract. The automatic detection and recognition of road signs play important role in the driver assistance systems and can increase the safety on the roads. In this paper we propose a system of a road signs classifier which is based on ensemble of the non Euclidean distance neural networks and an arbitration unit. The input to this system comes from the sign detection module which supplies a normalized, binarized and resampled pictogram of a detected sign. The system performs classification on deformable models. The classifier is composed of a mixture of experts (binary distance neural networks) operating on slightly tilted or shifted versions of pictograms. This ensemble of experts is orchestrated by an arbitration module which operates in the winner-takes-all mode with a novel modification of promoting the most populated group of unanimous experts. The experimental results showed great robustness of the system and very fast response time which is an important factor in the driving assistance systems.

1 Introduction

The automatic road signs detection and recognition by a computer with a single camera is a very appealing idea which can increase safety on our roads. Such systems have been developed by many research groups – for review refer to [2][4][8][9]. Most of them can be divided into two distinctive modules: a shape detector and a classifier, although a merged version of the two is also possible [1].

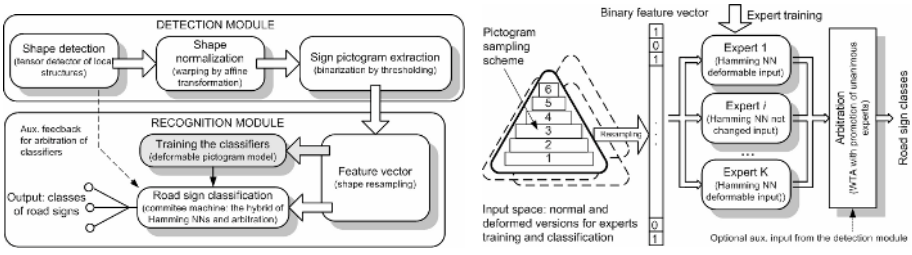
The very common practice at the recognition stage is to employ the back-propagation NN, usually operating on quite different input spaces [11][4]. The more robust classifier, which copes with occlusions and small shape rotations, is based on the Kohonen NN. The other NNs for sign recognition were also proposed, for instance the receptive field NN [7], radial basis function NN [12], and adaptive resonance NN.

In this paper we present the road sign classification system which is based on ensemble of experts: binary NNs and an arbitration module. The motivation behind using the binary NNs comes from the possible simple binarization of pictograms of the road signs, as well as very fast training and run-time response of this type of NNs. The connection of many binary NNs allows for simultaneous recognition of deformable versions of the same sign which can be encountered in images of natural scenes. The experiments and results concern the "A" group of road signs that can be found on Polish roads [10]. The presented methods can be easily extended, however.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 52 – 57, 2006.

2 Architecture of the System

Fig. 1a presents the structure of the complete system for road sign detection and recognition. Its operation begins with detection of the shape of a road sign from the certain group. In [3] the tensor based detector was presented for this purpose, although many other realizations are also possible [2][4]. After detection the shape has to be normalized to the reference shape and size. This image registration allows reliable comparison of a candidate shape with shapes from the data base of road signs.



a b

Fig. 1. Overall view of the system (a). The hybrid classifier for the "A" road signs (b).

The normalization is performed by affine image warping. Then a pictogram is binarized by simple thresholding. The practical threshold values are determined empirically (in our case it was 128). Last step of the preconditioning stage consists of pictogram resampling to create a binary feature vector.

The classification module is composed of the two main blocks (Fig. 1a): generator of the deformable reference objects, used only in the training phase (gray in Fig. 1a), and the hybrid classifier - a committee machine built of ensemble of expert NNs and the arbitration module. The deformable images are generated for training. The group of deformations is limited to the horizontal/vertical shifts (in practice 1-15 pixels).

During the recognition phase, the input image is only warped (registered) to the required size and put on the input layer of each expert. It is not deformed, however.

Fig. 1b depicts architecture of the neural hybrid classifier (lower path in Fig. 1a). This is a mixture of experts: each is in a form of a separate binary NN that is responsible for recognition in a group of a single deformation (e.g. shift, tilt, scale). The input of the arbitration module consists of the outputs of these experts. The most appropriate arbitration network turned up the WTA layer with promotion mechanism for unanimous experts.

Because of the binary input signals we choose the Hamming NN [6] (Fig. 2) which directly realizes the one-nearest-neighbor classification rule.

Each training data vector is assigned a single class and during recognition phase only a single nearest vector to the input pattern $\mathbf{x}$ is found and its class C_i is returned. There are two main phases of the operation of the expert-network: training (initialization) and classification. Training of the binary NN consists of copying reference patterns into the weights of the matrix $\mathbf{W_{pn}}$ in Fig. 2, as follows:

$$\mathbf{w}_i = \mathbf{x}_i \quad , \quad 1 \le i \le p \, , \tag{1}$$

where p is the number of input patterns-vectors $\mathbf{x}$, each of the same length n, w_i is the *i-th* row of the matrix $\mathbf{W}$ of dimensions p rows and n columns. For given n the computation time is linear with the number of input patterns p.

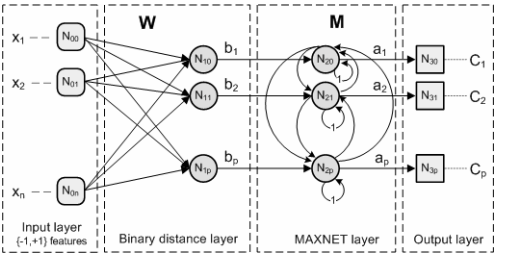


Fig. 2. The binary (Hamming) neural network as a classifier-expert module

The goal of the recursive layer $N_{2.}$ (Fig. 2) is selection of the winning neuron. The characteristic feature of this group of neurons is a self connection of a neuron to itself with a weight $m_{ii}=1$ for all $1 \le i \le p$, whereas all other weights are kept negative. Initialization of the $N_{2.}$ layer consists in assigning *negative* values to the square matrix $\mathbf{M}_{pp}$ except the main diagonal. Originally Lippmann proposed initialization [6]:

$$m_{kl} = -(p-1)^{-1} + \xi_{kl} \quad for \quad k \neq l, \quad 1 \quad for \quad k = l, \quad where \quad 1 \le k,l \le p, \quad p > 1 \quad , \tag{2}$$

where ξ is a random value for which $|\xi| \ll (p-1)^{-1}$. However, it appears that the most efficient and still convergent solution is to set equal weights for all neurons $N_{2.}$ which are then modified at each step during the classification phase, as follows [5]:

$$m_{kl} = \varepsilon_k(t) = -(p-t)^{-1} \quad for \quad k \neq l, \quad 1 \quad for \quad k = l, \quad where \quad 1 \le k,l \le p, \quad p > 1 \quad , \tag{3}$$

where t is a classification time step. In this case the convergence is achieved in $p-1-r$ steps, where $r>1$ stands for a number of nearest stored vectors in $\mathbf{W}$ [5].

In the classification phase the group $N_{1.}$ is responsible for computation of the binary distance between the input pattern $\mathbf{z}$ and the training patterns already stored in the weights $\mathbf{W}$. Usually this is the Hamming distance [5]:

$$b_i(\mathbf{z}, \mathbf{W}) = 1 - n^{-1} D_H(\mathbf{z}, \mathbf{w}_i) \quad , \quad 1 \le i \le p \ , \tag{4}$$

where $b_i \in [0,1]$ is a value of an *i-th* neuron in the $N_{1.}$ layer, $D_H(\mathbf{z},\mathbf{w}_i) \in \{0,1,...,n\}$ is a Hamming distance of the input pattern $\mathbf{z}$ and the *i-th* stored pattern $\mathbf{w_i}$ (*i-th* row of *W*).

In the classification stage the $N_{2.}$ layer operates recursively to select one winning neuron. This process is governed by the following equation [6]:

$$a_i[t+1] = \varphi\left(\sum_{j=1}^{n} m_{ij} a_j[t] \right) = \varphi\left(a_i[t] + \sum_{j=1, i \neq j}^{n} m_{ij} a_j[t] \right), \tag{5}$$

where $a_i[t]$ is an output of the *i-th* neuron of the $N_{2.}$ layer at the iteration step t, φ is a threshold function given as follows:

$$\varphi(x) = x \quad for \quad x > 0, \quad 0 \quad otherwise \, . \tag{6}$$

Depending on the chosen scheme (2)-(3) of the m_{ij} weights in (5) we obtain different dynamics of the classification stage. The iterative process (5) proceeds up to a point where only one neuron has value different than 0 – this neuron is a winner.

The arbitration unit is responsible of selecting the final winner, this time from the number E of competing expert modules . Its structure follows the same MAXNET scheme, analogous to the N_2 layer in Fig. 2. However, inputs to this layer comes directly from the experts $e \in \{1,...,E\}$ which reached their final states with a selected winner w_e. Their output values follow (5) when reaching the final step t_F and are real values $a^e_w[t^e_F] \in [0,1]$. Weights of the arbitration layer are initialized based on (2) with $p=E$. Due to randomization in (2) it appeared to be the most effective initialization method, especially in situations of similar output values a_w from the expert modules.

Sometimes many experts select the same class. It can happen especially for small amount of deformations and thus small number of experts. Such common voting can be a clue for selecting just this class (the similar behavior can be observed among human-experts or in democratic systems). To support this feature we introduced the "promotion" mechanism – *the most populous* group of *unanimous* experts (i.e. which voted for the same class) relax their inhibition weights to the members of their group, thus creating *a cooperating group of support*. If there is no such dominating group then this mechanism is not applied. The weight modification is done as follows:

$$m_{kl} = \begin{cases} m_{kl} + \gamma_G & \text{if} \quad m_{kl} + \gamma_G < 0 \\ m_{kl} \end{cases} \quad \text{for} \quad k \neq l, \{k,l\} \in G, \tag{7}$$

where γ_G is a positive constant value related with the mean of a cooperating group G:

$$\gamma_G = \frac{c}{\#G} \sum_{\{k,l\} \in G, k \neq l} |m_{kl}|, \tag{8}$$

where $\#G$ is a number of members in G, c is a constant (usually 0.01 or 0.05).

The classification process of the arbitration unit is launched just after all experts have selected their winners. It stops with only one winner selected by the arbitration unit which constitutes the output of the whole road-sign classification system. However, there is also a lower-bound threshold value imposed on the winner neuron, set to prohibit answers on the noise level (determined empirically to be 0.01-0.05).

3 Experimental Results

The presented system was implemented with the Microsoft® Visual C++ 6.0 on the IBM PC with Pentium 4 3.4G and 2GB RAM. After many experiments we choose the optimal size of a single (binary) reference image to 64×64 pixels. This gives 1278 bits for each pattern after the already described area sampling techniques that extracts pictograms. This is also a number of input neurons for each of the experts (Fig. 1b).

There are two main configurations of the system:

1. Create p experts, each classifying a sign based on q deformable patterns;
2. Create q experts, each classifying all road sign under a single deformation.

Since we changed q from 1 to 81 deformations, we choose the second option, which is also more suitable for incremental build of the system. The maximum tested version of such deformation was with shifts from -8 to 8 pixels in horz/vert directions with step of 2 pixels. This gives in total 9x9=81 possibilities. Tilts were not necessary.

To assess quality of the presented classification system our methodology consists of measuring Precision vs. Recall from the two sets of road signs data-bases (DBs):

1. The first DB consist of 50% of deformable (-8 to 8 shifts) signs that were used during the training and 50% of non-road-signs. Before classification each image was added the Gaussian noise at certain level. Results are presented in Table 1 for different signal-to-noise ratios and acceptance threshold in the arbitration.
2. The second DB is also 50/50, however the road-signs are taken from real scenes. The Gaussian noise was also added. Results are contained in Table 2.

Table 1. Objects (road signs) recognition accuracy from the DB with 50/50 of road-signs and non-road-signs images (6642 total). These road-signs were also used to train the system.

Arbit.Thresh→	0.005		0.01		**0.05**		0.07	
SNR [dB] ↓	Prec.	Recall	Prec.	Recall	Prec.	Recall	Prec.	Recall
40	0.5193	1.0000	0.6382	1.0000	1.0000	0.9935	1.0000	0.9873
14.2	0.5507	1.0000	0.6247	1.0000	1.0000	0.9888	1.0000	0.8925
8.1	0.5479	0.9936	0.6131	1.0000	1.0000	0.8147	1.0000	0.7932
4.6	0.5449	1.0000	0.6643	0.9747	1.0000	0.9454	1.0000	0.6593
2.1	0.5505	1.0000	0.6501	1.0000	1.0000	0.9740	1.0000	0.7497
0.2	0.5342	1.0000	0.6045	1.0000	1.0000	0.9815	1.0000	0.8654

In Table 1 we notice that the best results are for the acceptance threshold of 0.05. However, the higher value of this parameter causes fall of recall value, since some "good" objects are classified as "don't know". The perfect precision value reflects good auto-associative properties of the system.

Table 2. Road signs recognition accuracy from the DB with 50/50 of road-signs and non-road-signs images (436 total). These road-signs in the DB where extracted from the real road scenes.

Arbit.Thresh→	0.005		0.01		**0.05**		0.07	
SNR [dB] ↓	Prec.	Recall	Prec.	Recall	Prec.	Recall	Prec.	Recall
40	0.4321	0.8331	0.6665	0.9100	0.9664	0.9777	0.9393	0.9121
14.2	0.4303	0.8305	0.6104	1.0000	0.9898	0.9811	0.9478	0.8920
8.1	0.3666	0.9556	0.5699	0.8878	0.9002	0.8998	0.8921	0.7842
4.6	0.4001	0.7069	0.5070	0.9747	0.8813	0.8131	0.8005	0.6713
2.1	0.5505	0.8011	0.4439	0.7993	0.8201	0.8091	0.8101	0.8137
0.2	0.5342	0.6989	0.4078	0.8004	0.3002	0.6715	0.8099	0.7994

Table 2 presents results of recognition accuracy for images extracted from the real scenes. The results are quite promising, although due to some geometric deformations and intrinsic noise these results are a little bit worse than the presented in Table 1.

Table 3 presents the timings for the build (training) and classification modes of our system. They are quite good and allow real-time processing since the build stage can be off-line. It should be possible also to implement the system in hardware.

Table 3. Timings of the system build and classification for different configurations of experts

Number of experts:	81	49	25	9
System build time [s]:	0.687	0.406	0.203	0.078
Single classification time [s]:	0.031	0.015	<0.001	<0.001

4 Conclusions

The paper presents a committee machine for recognition of the road signs of the "A" group. The recognition is performed in a 1-nearest-neighbor fashion with the 2D deformable models. This allows recognition of input patterns from real scenes. The system consists of an ensemble of the experts and the arbitration module. Each expert is responsible for classification within a set of reference patterns which are uniquely deformed. Thanks to the easy registration of the triangle shaped patterns we could limit deformations only to the horizontal/vertical shifts. Since the input vectors are binarized pictograms, experts are Hamming NNs with the very good recognition quality and small computational resources – than also tested Hopfield NN. The output from all the experts is fed into the arbitration unit which is a modified version of the WTA NN. The introduced relaxation mechanism for a group of unanimous experts promotes "democratic-like" choice of a winner. The experiments showed low computational demands, fast execution and high robustness – it allows reliable classification of the road signs even from the highly deformed input.

Acknowledgement

This work was sponsored by the Polish scientific grant no. KBN 3T11C 045 26.

References

1. Amit, Y.: 2D Object Detection and Recognition, MIT Press (2002)
2. Chen, X., Yang, J., Zhang, J., Waibel, A.: Automatic Detection and Recognition of Signs From Natural Scenes. IEEE Trans. on Image Proc. 13(1) (2004) 87-99
3. Cyganek, B.: Object Detection in Multi-Channel and Multi-Scale Images Based on the Structural Tensor. Springer LNCS 3691 (2005) 570-578
4. Escalera, A., Armingol, J.A.: Visual Sign Information Extraction and Identification by Deformable Models. IEEE Trans. On Int. Transportation Systems 5(2) (2004) 57-68
5. Floréen, P.: Computational Complexity Problems in Neural Associative Memories. PhD Thesis, University of Helsinki, Department of Computer Science, Finland (1992)
6. Lippman, R.: An Introduction to Computing with Neural Nets. IEEE Transactions on Acoustic, Speech, and Signal Processing, v.3 ASSP-4 (1987) 4-22
7. Luo, R. C., Potlapalli, H.: Landmark Recognition Using Projection Learning for Mobile Robot Navigation, in Proc. IEEE Int. Conf. Neural Networks, v.4, (1994) 2703–2708
8. Nagel, H.H.: Computer Vision for Support of Road Vehicle Driver, Institut für Algorithmen und Kognitive Systeme, Karlsruhe (2003)
9. Piccioli, G., Micheli, E.D., Parodi, P., Campani, M.: Robust Method for Road Sign Detection and Recognition. Image and Vision Computing, v.14 (1996) 209-223
10. Road Signs and Signalization. Directive of the Polish Ministry of Infrastructure, Internal Affairs and Administration (Dz. U. Nr 170, poz. 1393) (2002)
11. Aoyagi, Y., Asakura, T.: A Study on Traffic Sign Recognition in Scene Using Genetic Algorithms and Neural Networks, in IEEE Conf. Electronics, Control (1996) 1838–1843
12. Zheng, Y. J., Ritter, W., Janssen, R.: An Adaptive System for Traffic Sign Recognition, in Proc. IEEE Intelligent Vehicles Symp. (1994) 165–170

Vehicle Classification in Wireless Sensor Networks Based on Rough Neural Network

Qi Huang[1,2], Tao Xing[2], and Hai Tao Liu[2]

[1] Department of Modern Physics, University of Science and Technology of China, Hefei, Anhui 230027, China
[2] Shanghai Institute of Microsystems and Information Technology, Shanghai 200050, China

Abstract. In this paper, a novel recognition system based on rough neural network is presented for the application of vehicle classification in wireless sensor network. The proposed system is evaluated using real-world signal datasets as well as two conventional methods. Compared with them, approach based on rough neural network achieves high performance improvement. Furthermore, the purposed system is extended for multi-channel sensor data fusion directly. Since the experiment results are attractive, an algorithm based on rough neural network is believed to have potential for applications of recognition and data fusion in wireless sensor networks.

1 Introduction

The emergence of small, low-power devices that integrate micro-sensing and actuation with on-board processing and wireless communication capabilities stimulates great interests in wireless sensor networks (WSN) [1]. A WSN is often deployed to perform tasks such as detection, classification, localization and tracking of one or more targets within the sensor field. The sensors are typically battery-powered and have limited wireless communication bandwidth. Therefore, efficient signal processing algorithms that consume less energy for computation and communication are needed for these applications.

Vehicle type classification is an important signal processing task that has found widespread military and civilian applications such as intelligent transportation systems. Typically, acoustic sensors are used for such a purpose.

Although there has been some several research devoted to multi-category classifications of vehicles based on acoustic signals [2, 3, 4], it is still very challenging to accomplish this work in WSN. The features that are extracted from the acoustic measurements are time-varying and contain a lot of uncertainties. This is due to variations of the environmental conditions (e.g., terrain and wind). Consequently, any classifier that makes use of acoustic measurements of vehicles for classification must account for these time-variations and uncertainties. Furthermore, processing nodes in WSN are always constrained with computation resources and energy consumption, the complexity of algorithms must be considered critically in practice.

In this paper, a novel recognition system based on rough neural network is presented. With features based on spectrum of acoustic signals, performance of the

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 58 – 65, 2006.

proposed system is evaluated using real-world signal datasets. Two familiar recognition systems, based on Gaussian mixture models (GMM) [5] and Back Propagation Neural Network (BPNN) [6], are also evaluated in experiments to provide a baseline performance. In contrast with them, the novel approach results in high performance improvement. Furthermore, system is extended for multi-channel sensor data fusion directly and demonstrated in experiments.

The paper is organized as follows: section 2 presents the theories of rough sets and rough neural network, section 3 describes the recognition system, section 4 presents the experiment results, and section 5 draws some conclusions and describes further work.

2 Theoretiacal Review

2.1 Rough-Set Theory

Rough set theory, proposed by Z. Pawlak [7, 8], provides a systematic framework for the study of the problems arising from imprecise and insufficient knowledge.

Let R be an equivalence relation on a universal set Z and Z/R denote the family of all equivalence classes induced on Z by R. There exits an equivalence class in Z/R, designated by $[z]_R$, that contains a training sample $z \in Z$. For any output class $A \subseteq Z$, the lower approximation $\underline{R}(A)$ and upper approximation $\overline{R}(A)$ are defined and they approach A as closely as possibly from inside and outside in the set respectively. Therefore, $\underline{R}(A)$ can be defined as the union of all equivalence classes in Z/R that are contained in A such that

$$\underline{R}(A) = \cup\{[z]_R \mid [z]_R \subseteq A, z \in Z\} \tag{1}$$

Whereas $\overline{R}(A)$ can be also defined as the union of all equivalence classes in Z/R that overlap with A like the following equation

$$\overline{R}(A) = \cup\{[z]_R \mid [z]_R \cap A \neq \phi, z \in Z\} \tag{2}$$

A rough set can be represented by $\underline{R}(A)$ and $\overline{R}(A)$ with the given set A as

$$R(A) = \left\langle \overline{R}(A), \underline{R}(A) \right\rangle \tag{3}$$

And the rough boundary of A by the equivalence classes Z/R is distinct as

$$RB(A) = \overline{R}(A) - \underline{R}(A) \tag{4}$$

2.2 Rough Neural Networks

The rough neural network (RNN) has been proposed by Lingras [9]. Combining artificial neural network with rough set theory, the input of network could be qualitative, i.e., the values of input can be in a certain range or varied during the

observation, as well as quantitative. Because of this point, RNN is believed to have potential for applications of recognition and data fusion.

As shown in Figure 1, there are three models for RNN. The value in a rough pattern is a pair of upper and lower bounds. A rough neuron s can be viewed as a pair neurons called *upper-bound neuron* ($\bar{s}$) and *lower-bound neuron* ($\underline{s}$) to receive a pair pattern. A *full-connected* model that displays a rough neuron r connects to a proper rough neuron s with four connections. If the rough neuron r *excites* the activity of s, the connections from $\bar{r}$ to $\bar{s}$ and from $\underline{r}$ to $\underline{s}$ are created respectively. On the other hand, if r *inhibits* the activity of s, the connections from $\bar{r}$ to $\underline{s}$ and from $\underline{r}$ to $\bar{s}$ are created respectively.

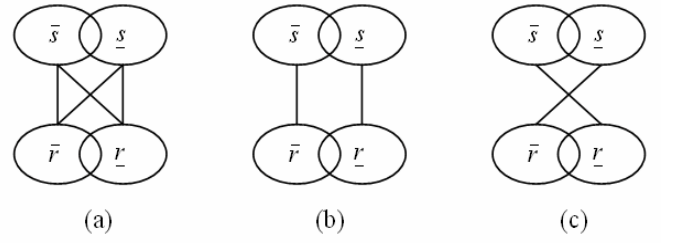


Fig. 1. Three different connections between two rough neurons

In this paper, the full-connected model rough neuron is applied into the neurons in the classification networks. The inputs in the full-connected model rough neuron are defined as

$$input_{\bar{s}} = \sum_{r=1}^{m} \left(w_{\bar{s},\bar{r}} \cdot output_{\bar{r}} + w_{\bar{s},\underline{r}} \cdot output_{\underline{r}} \right) \tag{5}$$

and

$$input_{\underline{s}} = \sum_{r=1}^{m} \left(w_{\underline{s},\bar{r}} \cdot output_{\bar{r}} + w_{\underline{s},\underline{r}} \cdot output_{\underline{r}} \right) \tag{6}$$

The outputs of a rough neuron s are expressed using an activation function as

$$output_{\bar{s}} = \max\left(t\left(input_{\bar{s}}\right), t\left(input_{\underline{s}}\right) \right) \tag{7}$$

and

$$output_{\underline{s}} = \min\left(t\left(input_{\bar{s}}\right), t\left(input_{\underline{s}}\right) \right) \tag{8}$$

where activation function can be defined as

$$t(x) = \frac{1}{1 + e^{-kx}} \tag{9}$$

and k is a constant with a range from 0 to 1.

The training stage in the development of a rough neural network is similar to the conventional neural network. In this paper, genetic algorithm is adopted to train the network.

3 Recognition Systems

3.1 Feature Extraction

Given the similarity between vehicle and speaker recognition, Mel-frequency cepstrum coefficients are modified to be used as feature vector.

The steps of the feature extraction can be summarized as follows:

1) Audio signals recording at 1025.641 Hz are normalized and framed using a 250 msec -long overlapping Hamming window, with a frame rate of 100 msec.
2) Spectrum of each frame is computed and then filtered by a filter bank. Since the frequency components using for classification are below 300 Hz, individual filter's center frequency is placed in accordance with the frequency scale as:

$$F_n = 900 \cdot \log 10(1 + f_n / 300) \tag{10}$$

$$f_n = 12n, n = 1 \ldots 25 \tag{11}$$

3) Discrete cosine transform (DCT) is applied to the log of the filter-bank output. Finally, the feature vector is composed of 25 cepstrum coefficients. The two baseline systems evaluated in this paper are based on these features.

In the case of the system based on RNN, the log of the filter-bank output is directly used as features. The calculating flowing is the same except setp3 is bypassed.

3.2 Rough Neural Network Classifier

In the problem of vehicle type classification in WSN, the vehicle moves throughout the field and as such the vehicle signals change with time due to variations of vehicle-traveling speed, and signal to noise ratio (SNR) of the acoustic measurements that is subject to the variability of the distance between the vehicle and the sensor system. That is to say the vehicle signals are *varied in time domain;* on the other hand, when a sensor array is used to record vehicle signals, these multiple simultaneous signals are believed to be *varied in space domain*, due to the difference between sensors in multiple channels.

Since the classifier based on RNN has ability to solve the qualitative problem, it is used to recognize type of vehicles in WSN with varied input either in time domain or in space domain. Figure 2 shows the structure of RNN classifier used in experiments. The network has 25 input neurons, corresponding to the 25-dimisional feature vector, 25 hidden layer neurons, and 4 output neurons. Rough neurons in network are all full-connected. Recognition result is the average between the outputs of upper-bound and lower-bound neurons in the output layer.

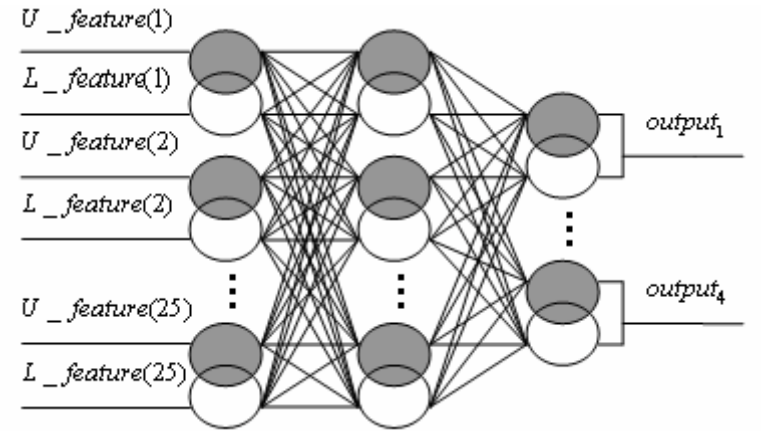


Fig. 2. Rough neural network classifier

In the case of single-channel classification, Group T feature vectors $feature_i, i = 1 \ldots T$ extracted from consecutive signal frames into a block, maximal and minimal feature vectors are generated as follow:

$$U\_feature = \max\{feature_i, i = 1 \ldots T\} \tag{12}$$

$$L\_feature = \min\{feature_i, i = 1 \ldots T\} \tag{13}$$

The max and min operations are performed in each component in feature vector. Then these features are fed to the corresponding upper-bound and lower-bound input neurons in RNN.

In the case of multiple-channel sensor data fusion, feature vectors extracted from simultaneous signal frames from multiple channels are grouped to generate new features for RNN. Other processing is as same as that mentioned above.

4 Experiments

The acoustic signature data set used in the experiments is collected from four real-world experiments. The database is composed by more than 100 data runs (single target) from four different types of ground vehicles (Table 1). The vehicles were traveling at constant speeds that varied from 5 km/h to 40 km/h depending upon the particular run, the vehicle, and the environmental condition. The closest point of approach to the sound-capture system varied from 25 m to 50 m. The acoustic data was collected with a 3-element equilateral triangular microphone array with an equilateral length of 10 inch. The microphone recordings were low-pass filtered at 400 Hz with a 6*th*-order filter to prevent spectral aliasing and high-pass filtered at 25 Hz with a 1*st* -order filter to mitigate wind noise. The data was digitized by a 16-bit A/D at the rate of 1025.641 Hz. The distance between microphones generated a time

Table 1. Dataset in experiments

	Describes	Runs	Records
Class 1	Heavy track vehicle	33	99
Class 2	Light track vehicle	30	90
Class 3	Heave wheel vehicle	29	87
Class 4	Light wheel vehicle	44	132

delay in waveform arrival to the microphones. But the delay was smaller than 1 millisecond for all the conditions and hence, the delay was negligible for all practical purposes at the given sampling rate of 1025.641 Hz.

The database was evenly divided into a training and a testing set in order to evaluate the performance of the system. Given that the microphone array provided three simultaneously-recorded utterances per run, all three recordings were assigned to either one of the sets.

The three systems described in this paper classify complete utterances as being produced by one of the vehicles. Since the microphone array provided three simultaneous recording of utterances, we can classify each recording independently (single channel), or make a joint use of the three recordings (multiple channel). For the system based on GMM and BPNN, this is accomplished by aggregating individual classification results with a voting procedure, while in the case of system based on RNN, multi-channel sensor data fusion is done as described in section 3.2.

Table 2 presents the error rates of the systems for both single-channel and multiple-channel testing. Features are extracted as described in section 3.1; The GMM used a 32-mixture models; BPNN is composed of 25 input neurons, 16 hidden layer neurons and 4 output neurons; RNN used a 25-nodes full connected model as shown in Figure 2. Table 3 shows the confusion matrices obtained with our system for single channel recognition.

As shown in Table 2, in the case of single-channel classification, approach based on RNN achieves high performance improvement in contrast with either GMM-based recognizer or BPNN-based recognizer. The experiment results also indicate that RNN is an efficient algorithm for multi-channel sensor data fusion.

In the case of single-channel classification, parameter T in eq.12 and eq.13, which indicates the number of consecutive signal frames used for feature extraction, seems to be critical. Recognition error rates for varied values of T are shown in Figure 3. However, even in the worst case, the approach based on RNN outperforms either GMM-based recognizer or BPNN-based recognizer.

Table 2. Recognition error rates

	RNN		GMM		BPNN	
	train	test	train	test	train	test
Single channel	0.0%	11.71%	3.45%	18.53%	0.49%	17.56%
Multiple channel	0.0%	7.31%	2.46%	15.61%	0.0%	16.10%

Table 3. Confusion matrices using RNN (single channel)

Class	1	2	3	4
1	94	4	1	0
2	3	86	1	0
3	0	2	77	8
4	0	0	5	127

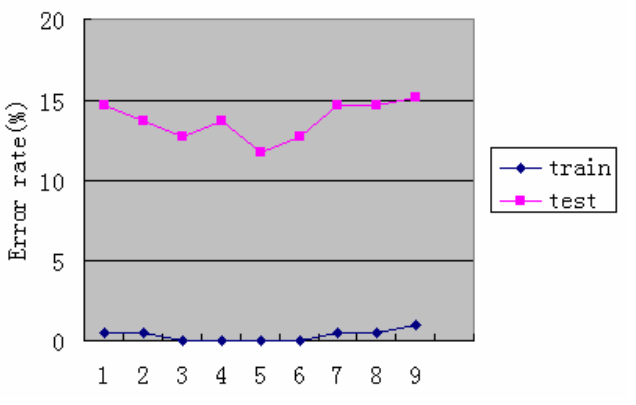


Fig. 3. Plot of recognition error rates using RNN classifier with varied value of T. The best performance is achieved when $T = 5$ with 11.71% error rate in test dataset, while the worst is 15.12% when $T = 9$.

5 Conclusions and Future Work

This paper has presented a novel approach based on rough neural network for acoustic signature recognition of vehicles in WSN. Experiment results for both single- channel classification and multiple-channel sensor data fusion indicate that the purposed algorithm outperform conventional system based on either GMM or BPNN.

However, previous research has focused on the classification based on signals obtained and processed in a centralized manner, while applications working in distributed way are more common in WSN. An ideal of extending the algorithm based on RNN to work with distributed sensors is attractive and we will focus on it in the feature.

References

1. Estrin, D., Girod, L., Pottie, G., Srivastava, M.: Instrumenting the World with Wireless Sensor Networks. In: Proceedings of International Conference on Acoustics Speech and Signal Processing, Salt Lake City, Utah (2001) 2675-2678
2. Eom, K.B.: Analysis of Acoustic Signatures from Moving Vehicles Using Time-Varying Autoregressive Models. Multidimensional Systems and Signal Processing 10 (1999) 357-378

3. Liu, L.: Ground Vehicle Acoustic Signal Processing Based on Biological Hearing Models. Master's thesis, University of Maryland, College Park (1999)
4. Wu, H., Siegel, M., Khosla, P.: Vehicle Sound Signature Recognition by Frequency Vector Principal Component Analysis. IEEE Trans. Instrumentation and Measurement 48(5) (1999) 1005-1009
5. Reynolds, D., Rose, R.: Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models. IEEE Trans. Speech and Audio Processing 3(1) (1995) 72–83
6. Bishop, C.M.: Neural Networks for Pattern Recognition. Clarendon Press, Oxford (1995)
7. Pawlak, Z.: Rough Sets. Int. J. of Information and Computer Science 11(5) (1982) 341-356
8. Pawlak, Z.: Rough Classification. Int. J. of Man-Machine Studies 20(5) (1984) 469-483
9. Lingras, P.: Comparison of Neofuzzy and Rough Neural Networks. Information Sciences 110(3) (1998) 207-215

Neural Network Approach to Identify Model of Vehicles

Hyo Jong Lee

Chonbuk National University, Jeonju 561-756, Korea

Abstract. The number of vehicles is rapidly increased as modern industrialization is developed worldwide. Vehicle recognition has been studied for some time because many people acknowledged it is critical information to solve vehicle-related problems. However, few researchers have tried to recognize the model of vehicles. In this paper a novel method is proposed to recognize vehicles' model corresponding manufacturers in order to increase the efficiency of recognition. Texture descriptors are computed from the front image of vehicles. Then, a three-layer neural network was built and trained with the texture descriptors for model recognition. The proposed method demonstrates 94% recognition rate for moving vehicles' models.

1 Introduction

In the recognition process of vehicles the features or attributes of the vehicles have critical roles. It is also the easiest way to recognize license plate numbers compared to other parts. The study of license plate recognition has been studied even before 1990. It was not simple as to measure traffic flow. Ikeda et al. measure how much traffic was congesting by vehicles by differential road images [1]. The idea of differential images can be applied to recognize moving vehicles.

It is widely recognized that neural network is applicable to various problems, such as artificial intelligence and pattern recognition, which are not suitable for traditional sequential computation. Many problems are not suited to sequential algorithms causing use to expend a great deal of effort on the development of sophisticated algorithms. Furthermore, it fails to find acceptable solutions in some cases.

The traditional approach is either to train segmented number regions with a neural network or to apply pattern recognition of number regions [2, 3, 4]. A few software are commercialized in order to recognize vehicles at a parking lot [5, 6]. However, these approaches require specific driving regulations for moving vehicles. For example, a vehicle is assumed to be stopped in front of parking lot gate to be recognized.

In this paper an algorithm is proposed to identify license plate and model name from driving Korean vehicles during the year 2003. A correct region of vehicle is extracted and fed into a three-layer neural network for training [7, 8]. The trained weight values of segmented numbers are used for regions of new vehicles.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 66–72, 2006.

2 Neural Network

In general artificial neural network consists with processing elements, which have predefined activation function, and connection weights that connect each processing element. Multi-layer networks use a variety of learning techniques, such as a back propagation multi-layer perceptron. Back propagated Delta Rule Networks is well-known developments of the Delta rule for single layer networks. In back propagation algorithm, the output values are compared with the correct answer to compute the value of some predefined error-functions. The error is then fed back through the network. Using this information the algorithm adjusts the weights of each connection in order to reduce the value of the error function. Thus the back propagation can be applied on networks with differentiable activation functions. The weight change rule is a developed form of the perceptron learning rule. Weights are changed by an amount proportional to the error at that within times the output of the unit feeding into the weight.

In this paper a back propagation learning algorithm in the form of supervised learning is adapted to recognize license plate numbers and model names of vehicles driving in Korea. Two networks were designed to recognize both license plate numbers and model names, respectively. The recognition of numbers and characters from license plates has been studied for some time. Thus, only the recognition part of model names is being focused here.

3 Image Pipeline

The process of vehicle recognition can be consisted with a sequence of preprocessing that handles video stream from a camera installed at roads. Fig. 1 displays the flow chart of image pipeline for vehicle recognition.

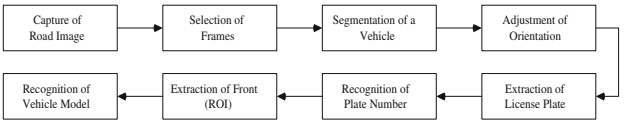


Fig. 1. Flow chart of recognition of vehicle model

It is necessary to remove more noise from the differential image because static region does not show exact zero intensity. A simple threshold process to the images generates binary images of vehicle's edges. The threshold value has been decided depending on a photographic condition such as weather and day light. For example, a pixel values background region has higher values when a slight movement of a camera is detected due to strong wind. If a threshold value is too low, edges of lines and signs on road are detected as vehicle's edge. Then projection histogram of binary images is calculated from both vertical and horizontal direction. The bounding box of a vehicle is defined from the projections.

The next step is to extract license plate and its plate numbers which are recognized. In this paper detail procedure is omitted. Readers, who are interested in the procedure of license plate numbers, refer Lee's work [9]. It has been tried to recognize vehicle's model by applying edge lines of vehicles [10, 11]. Their approach was able to classify only the type of vehicles, such as a sedan, a truck, or a van. The second to the last step of the image pipeline is to extract the region of interest (ROI) from a vehicle image. In this paper information of texture at front area of a vehicle has been disseminated to identify manufacturer and model of corresponding vehicle.

4 Texture Descriptors

In this paper texture descriptors are adapted to recognize model names of vehicles. Texture descriptors [12, 13] have been widely used to quantify texture of objects. Image texture is represented by difference between brightness of pixels. Gray level cooccurrence matrix(GLCM) is generated based on the spatial relationship of pixels to quantify texture. GLCM texture considers the relationship between two pixels at a time, called the reference and the neighbor pixel. GLCM matrix $G[i,\ j]$ represents the distribution probability of all pixels of which distance is d between two pixels of level i and level j. The dimension of GLCM is very large because gray images generally have 256 levels. Furthermore, many different kinds of texture descriptors can be also defined, such as contrast, dissimilarity and homogeneity based on brightness; momentum and entropy based on regularity; average, variance, and correlation based on distribution statistics. In this paper four basic texture descriptors contrast, homogeneity, entropy and momentum are defined for the GLCM matrix G as below and used.

$$Contrast = \sum_i \sum_j |i-j|^2 G[i,\ j] \tag{1}$$

$$Homogeneity = \sum_i \sum_j \frac{G[i,\ j]}{1+|i-j|^2} \tag{2}$$

$$Entropy = \sum_i \sum_j G[i,\ j](-log\ G[i,\ j]) \tag{3}$$

$$Momentum = \sum_i \sum_j G[i,\ j]^2 \tag{4}$$

5 Experiment

5.1 Environment

Video stream has been captured from a suburban area. A Sony TRV900 digital video camcorder was installed six meters above the surface of the road facing the sun as direct as possible in order to avoid problematic luminous changes. The

width of road was approximately three meters wide and the camera zoom was set as to capture a whole single lane so it touched the bottom of two corners of the viewfinder.Total length of video was approximately three hours long. Total number of vehicles used in this research was a 415 in 24 different vehicles' model. The neural network training was performed on IBM supercomputer RS/6000 SP9076-550.

5.2 Vehicle Extraction

The video images were fed into the system as the resolution of pixels through a frame grabber with a read-in speed of 28 frames per second. The front vehicles' bounding boxes were extracted from the input image. Fig. 2 shows some examples of the algorithm described in Section 3.

Fig. 2. Extracted bounding boxes of segmented vehicles

Fig. 2(a) and 2(b) are examples of successful extraction for a passenger vehicle and a truck. Fig. 2(c) is an example of bounding box extraction of abnormal location. Fig. 2(d) is a case that the algorithm ignores extraction due to the vehicles too close together driving into a viewfinder. Fig. 2(a)-(c) are cases of successful extraction of license plates. However, the algorithm extracts a license plate of the first vehicle only in Fig. 2(d). This kind of situation has been occurred only 0.25%. Thus, it is assumed that the performance does not depend on the wrong extraction of a bounding box in this research.

5.3 Extraction of License Plate

The algorithm described in Section 3 was applied to the images of segmented vehicles. The seven out of 415 vehicles produced wrong region of license plate. The 2% of errors were occurred in cases of shifted license plate, of partial plate, and of unnecessary region of inclusion. Fig. 3 shows some of those cases alongside successful extractions.

5.4 Extraction of ROI

Each vehicle model has its own pattern and shape at the front of vehicle with various placing of headlights, different pattern of the radiator grille and more features. Thus ROI for texture descriptors is defined around the upper front region of vehicles including the radiator grille and part of the head lights. In order to find the ROI, a base line has been set at the center of license plate. A bounding box is tweaked on the base line from the left to the right to adjust best position. Figure 4 shows the found ROI as examples.

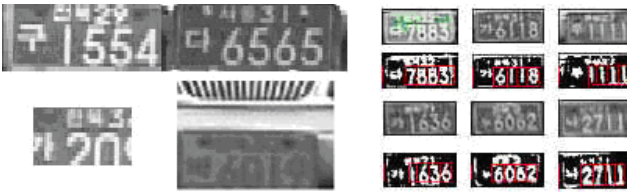


Fig. 3. Examples of failed and successful region of license plate

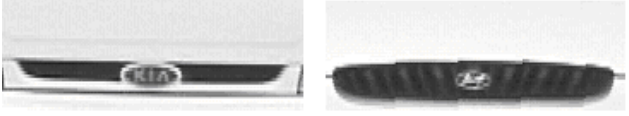


Fig. 4. Examples of ROI for texture descriptors

5.5 Recognition of Vehicles' Model

Texture descriptors are calculated with Equations (1)-(4) from the front region extracted described in Section 5.4. A good separator distinguishes one model from all others. However, some descriptors are not distinguishable from two different models. Figure 5 shows that texture descriptor Cont-1s are very similar for all ten vehicle model. Homo-4 ranges wider than Cont-1, however five Homo-4s are almost identical again. Thus, four texture descriptors contrast, homogeneity, entropy and momentum are computed in four directions east, west, south-east and north-west and used all together.

Those 16 texture descriptors are fed into the three-layer back propagation neural network. The number of output nodes is 24 because 24 different Korean vehicles are chosen in this research. The number of hidden layer has been set to 150 nodes considering faster convergence time and higher accuracy.

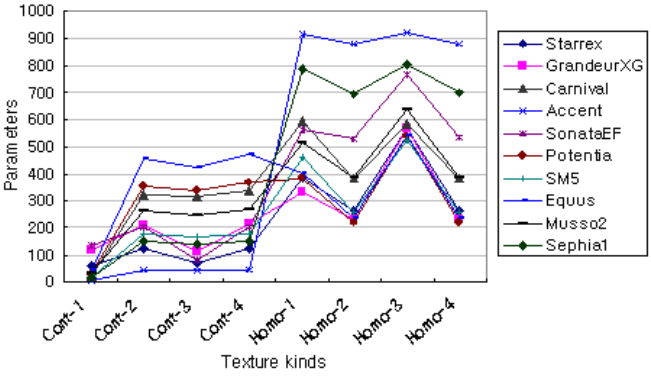


Fig. 5. Behavior of texture descriptors

In order to measure the efficiency of algorithms, sensitivity and specificity are defined with TP(True Positive) , TN(True Negative), FP(False Positive) and FN(False Negative) as below:

$$Sensitivity = TP/(TP + FN) \times 100$$

$$Specificity = TN/(TN + FP) \times 100 \quad (5)$$

The sensitivity of each model refers to the fraction of vehicles that were actually recognized from the total number of vehicles in the model group. The specificity of each model refers to the fraction of vehicles that are correctly not recognized from the total number of vehicles outside of the model group.

Table 1 shows the result of recognition of 24 different models and 415 vehicles. TP refers true positive, which are numbers of correctly identified vehicles. Miss refers to numbers of incorrect identifications, which is the sum of FN and FP. For example, 14 Frontier1s out of 15 Frontier1s are identified correctly and one Frontier1 is recognized as another model. Three other models were incorrectly identified as Frontier1s. Average sensitivity and specificity are 93.7% and 99.7%, respectively. There might be two ways to improve recognition rate. The first one is to train the neural network with more number of data. The other is to use better texture descriptors, which separates from different models accurately.

Table 1. Result of vehicle recognition

Model	Total	TP	Miss	Sens.	Spec.
Porter	50	50	3	100	99.2
Frontier1	15	14	3	93.3	99.3
StarRex	30	26	2	86.7	99.5
Frontier2	27	27	1	100	99.7
EF Sonata	21	21	0	100	100
Sonata2	17	16	2	94.1	99.5
NewEF	24	24	3	100	99.2
Sephia1	14	12	1	85.7	99.8
Carnival	23	21	1	91.3	99.7
Credos1	17	16	0	94.1	100
Grace	16	14	0	87.5	100
GranduerXG	18	17	0	94.4	100
Musso2	18	16	1	88.9	99.8
Istana	10	10	1	100	99.8
NewGrand	14	14	1	100	99.8
...	...	...	...	...	...
Total	415	389	26	93.7	99.7

6 Conclusion

It is critical to recognize license plate or model names of driven vehicles for traffic control or development of automated systems. The recognition of license plates

was tried by many scientists. However, the model names and manufacturers of vehicles have not been tried enough. In this paper algorithms are proposed to recognize vehicle models along with license plates. A typical three-layer back propagation neural network is constructed to train numbers from vehicle's license plate. The recognition rate was about 95% for moving vehicles.

Gray level co-occurrence matrix has been calculated from the front region of vehicles. Then four texture descriptors contrast, homogeneity, entropy and momentum were computed at four different directions.The proposed algorithm was tested with the 16 texture descriptors for 415 vehicles, which were driving in a suburban area in Korea. The performance of the proposed algorithm was measured with mixed data of trained and untrained data sets. The recognition rate was about 94%. The overall recognition rates will be higher with more numbers of training and with better texture descriptors.

References

1. Ikeda, T., Ohnaka, S., Mizoguchi, M.: Traffic Measurement with Roadside Vision System. Vol. 3, Proc. of 13th ICPR, 1996
2. Lotufo, R.A., Morgan, A. D., Johnson, A.S.: Automatic Number-plate Recognition. IEEE Colloquium on Image Analysis for Transport Applications (1990)
3. Wei, W., Li, Y., Wang, M., Huang, Z.: Research on Number-plate Recognition Based on Neural Networks. Proceedings of the 2001 IEEE Signal Processing Society Workshop on Neural Networks for Signal Processing XI, (2001) 529–538
4. Parisi, R., Claudio, E.D.,Lucarelli, G., Orlandi, G.: Car Plate Recognition by Neural Networks and Image Processing. Proceedings of the IEEE International Symposium on Circuits and Systems, Vol. 3, (1998) 195–198
5. Neuricam: Number Plate Recognition System NC6000 Data Sheet. http://www.neuricam.com (2002)
6. Rahman, C., Badawy, W., Radmanesh, A.: A Real Time Vehicle's License Plate Recognition System. Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance (2003)
7. Lee, H.: A Study on the Recognition of Vehicles on the Road. Chonbuk National University Thesis (2002)
8. Duda, R. O., Hart, P.E., Stork, D.G.: Pattern Classification. 2nd edn. Wiley Interscience, New York (2000)
9. Lee, H.J.: Recognition of Number Plates of Moving Vehicles. Proc. of Conference on Imaging Science, System, and Technology, (2005)
10. Wu, W., Sen, Z.O., Mingjun, W.: A Method of Vehicle Classification Using Models and Neural Networks. A Method of Vehicle Classification Using Models and Neural Networks, Vol. 4, (2001) 3022–3026
11. Limin, X.: Shape Recovery and Recognition Using Generic Models. Shape Recovery and Recognition Using Generic Models, (2002) 1055–1059
12. Parker, J.R.: Algorithms for Image Processing and Computer Vision. John Wiley and Son, New York (1996)
13. Pitas, I.: Digital Image Processing Algorithms and Applications. Wiley Inter-Science (2000)

An Intelligent Vehicle Security System Based on Modeling Human Driving Behaviors

Xiaoning Meng, Yongsheng Ou, Ka Keung Lee, and Yangsheng Xu

Department of Automation and Computer-Aided Engineering,
The Chinese University of Hong Kong,
Shatin, N.T., Hong Kong
xnmeng@acae.cuhk.edu.hk

Abstract. This paper presents an intelligent vehicle security system for handling the vehicle thefts problem under the framework of capturing and analyzing dynamic human behaviors. Since human driving skill is a kind of dynamic biometrical feature which is complex and difficult to imitate, it is unique and more secure than static features such as password, fingerprint and face. By utilizing this dynamic property we focus on the research ideal of classifying the drivers into authorized ones and unauthorized ones by modeling their individual driving performance. Firstly, we develop an experimental system architecture. We collect the data of steering, acceleration and braking directly from human driving behaviors as inputs to the system, which aims to achieve better robustness and efficiency. Then, we use fast fourier transform (FFT), principal component analysis (PCA) and independent component analysis (ICA) for data reduction. The features extracted are sent to support vector machine (SVM) for learning and recognition. In the next step, we embed the intelligent classifier into a security system to identify the authorized drivers in response to the real time driving performances. Finally, the experimental results verify that the proposed method is valid and useful against the vehicle thefts problem with a success rate of around 80%.

1 Introduction

1.1 Motivation

Nowadays, much research are focused on the development of intelligent vehicles. However the problem of vehicle thefts has attracted less attention. According to the National Insurance Crime Bureau, nearly 1.2 million vehicle thefts occurred each year since 2001 in the U.S., causing 8 billion US dollars in losses. Therefore the work on car security is significant. A concept of intelligent vehicle security based on capturing and analyzing dynamic human driving behaviors is proposed in this paper. It means that a car with this technology embedded can identify the driver through the driving performance in real time. When an illegitimate driver come to use the car and the demonstrated driving behaviors do not match the specified model, the car will automatically stop running and deliver alarm signals accordingly.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 73–81, 2006.

To achieve this goal, we present a framework of capturing and analyzing dynamic human driving behaviors in real time. Since different persons display different behaviors while driving and dynamic human behaviors are very difficult imitate by others, the dynamic features involved in human driving behaviors can be utilized as the unique and secure inputs to the system. By learning from the driving performances, the intelligent classifier can be embedded into an IC-based car key, through which the vehicle security system can identify the valid drivers based on the ways the vehicle are driven and the drivers behave.

1.2 Related Work

In the past decade, significant researches towards learning skills directly from human have been conducted primarily by Asada's group at MIT[1], the Navlab group at CMU[2] and our group at CUHK[3][4]. In [1], a debarring robot is controlled through an associative neural network which maps process parameter features to action parameters from human control data. In [2], Pomerleau implements real-time road-following with data collected from a human driver and a static feedforward neural network learns to map coarsely digitized images of the road ahead to a desired steering direction. In [3] and [4], we concentrate on modeling closed-loop reaction skills where sensory feedback to the human is required to successfully complete the task and abstracting models of dynamic human control strategy (HCS), a particular subset of human reaction skills. All these studies can model human driving behaviors for some specific maneuvering applications, however, they are not designed to identify whether the valid drivers are in terms of security applications, which can prevent the car theft problem.

1.3 Main Attributes of Our System

We highlight the following aspects in this paper.

Firstly, alive biometrical features in dynamic human behaviors are utilized in the system. Comparing with other patterns in common security applications, such as password, fingerprint, facial recognition, etc, dynamic human behaviors are unique and hard to duplicate. Therefore, it is more secure in utilizing dynamic human behavior models for the proposed security application.

Secondly, the system is robust and efficient in real time performance since we collect the signals directly from human driving controls, which include steering, acceleration and braking. We do not utilize other car dynamics and environmental variables such as the car's yaw angle with respect to the road, lateral offset to the road's center, the road curvature, etc. Capturing those data needs visual sensors, position or velocity sensors, which are not entirely robust in real traffic environments. More input data, especially the data for the visual analysis, consumes much resource for computing, causing the low efficiency in real time response.

Thirdly, the intelligent security system is easy to install on a normal vehicle by adding on functional modules. No complicated sensors such as camera and position sensor are needed, and not much space and time is required for system

installation and adjustment. The car therefore has little difference from a normal car and the drivers are not distracted by the monitoring system and sensors during driving.

Lastly, we develop a methodology to capture and analyze the characteristics of human behaviors into computational representations. It is easily scalable for other applications.

The remainder of this paper is organized as follows: section two introduces the experiment platform designed for the vehicle security system. Section three describes the methods for data preprocessing which are FFT, PCA as well as ICA and section four describes SVM for the learning process. Section five shows the experimental results. Finally, the conclusion and future work are presented in the last section.

2 Experimental Design

Figure 1 shows the architecture of the experimental intelligent vehicle security system. The proposed system consists of three subsystems, which are driving environment simulation subsystem, human behavior data capturing subsystem and human behavior analysis subsystem.

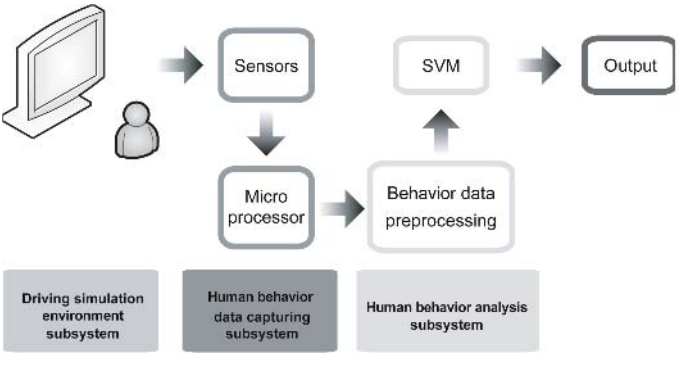


Fig. 1. Architecture of the experimental system

Figure 2 shows the driving simulation subsystem, a simulated driving environment which bears substantial resemblance to a comparable real driving task is developed. We apply one PC to offer the driving environment for the driver, including rendered 3-D graphics display as well as realistic controls, which are steering wheel and pedals for gas and brake, experienced as if driving in a real car on road.

In the data capturing subsystem, a processor circuit board is utilized to sense and gather the individual driving behavior data from the driving environment simulation subsystem. With the driving control sensing device, 3 channels of analog signals are gathered, as shown in Figure 3, which are steering, acceleration and braking. Those signals are processed by an analog-to-digital converter at

the sampling time of 100 ms and then the digitized values are sent to the micro computer (in our experiment ATmega8535L). Then the 3-D data are transferred to the human behavior analysis subsystem through the RS232 port for further processing.

Fig. 2. The driving simulation hardware

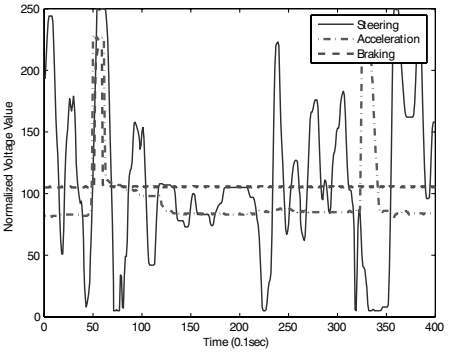


Fig. 3. Human driving behavior data

In the human behavior analysis subsystem, the methods introduced in the following sections are applied to the data retrieved. For our goal is to distinguish the authorized drivers from others, two model libraries of authorized drivers and unauthorized drivers are generated from the corresponding input data by machine learning. Once the models are ready, we implement them as the classifier in the system in response to the real time individual driving performance.

3 Data Preprocessing

It is necessary and important to apply data reduction in data preprocessing for modeling the human behavior data, since failures in feature selection reduces the efficiency of the system performance significantly. Among several feature extraction methods, Fast Fourier Transform (FFT), Principal Component Analysis (PCA) and Independent Component Analysis (ICA) are widely investigated in the area of pattern recognition.

3.1 PCA

The method can be described in brief as follows. Suppose that we have two sets of training samples: A and B. The number of training samples in each set is N. Φ_i represents each eigenvector produced by PCA. Each of the training samples, including positive samples and negative samples, can be projected into an axis extended by the corresponding eigenvector. By analyzing the distribution of the projected $2N$ points, we can roughly select the eigenvectors which have more human behavior information. The following is a detailed description of the process.

1. For a certain eigenvector Φ_i, compute its mapping result according to the two sets of training samples. The result can be described as $\lambda_{i,j}$ $(1 \leq i \leq M, 1 \leq j \leq 2N)$.
2. Train a classifier f_i using a simple method such as Perception or Neural Network which can separate $\lambda_{i,j}$ into two groups: valid driver and others with a minimum error $E(f_i)$.
3. If $E(f_i) < \theta$, then we delete this eigenvector from the original set of eigenvectors.

M is the number of eigenvectors and $2N$ is the total number of training samples. θ is the predefined threshold. The left few eigenvectors are selected.

3.2 ICA

A fixed-point algorithm is employed for independent component analysis [5]. The goal of the ICA algorithm is to search for a linear combination of the prewhitened data $x'_i(t)$, where $y = w^T x'(t)$, such that the negentropy (non-gaussianity) is maximized. w is assumed to be bounded to have norm of 1 and g' is the derivative of g. The fixed point algorithm [5] is as follows :

1. Generate an initial random vector w_{k-1}, $k = 1$
2. $w_k = E\{x' g(w_{k-1}^T x')\} - E\{g'(w_{k-1}^T x')\} w_{k-1}$
3. $w_k = w_k / \|w_k\|$
4. Stop if converged ($\|w_k - w_{k-1}\|$ is smaller than certain defined threshold). Otherwise, increment k by 1 and return to step 2.

If the process converges successfully, the vector w_k produced can be converted to one of the underlying independent components by $w_k^T x'(t), t = 1, 2, ..., m$. Due to the whitening process, the columns of B are orthonormal. By projecting the current estimates on the subspace orthogonal to the columns of the matrix B which are found previously, we are able to retrieve the components one after the other. Using this approach, the knowledge about the human driving behavior model is not required.

4 SVM Classifier

Our goal is to separate the drivers into two classes: valid drivers and others, according to a group of features. Support Vector Machine is a new technique in the field of statistical learning theory, which we choose as classifiers in our proposed system.

In SVM, the basic idea is to map the data X into a high-dimensional feature space f via a nonlinear mapping Φ, and to do linear regression in this space.

$$\mathrm{f}(\bar{x}) = (\omega \cdot \Phi(\bar{x})) + b \quad \text{with } \Phi : R^N \rightarrow \mathcal{F},\ \omega \in \mathcal{F}, \tag{1}$$

where b is a threshold. Thus, linear regression in a high dimensional (feature) space corresponds to nonlinear regression in the low dimensional input space R^N. Note that the dot product in Equation (1) between ω and $\Phi(\bar{x})$ would have to be computed in this high dimensional space (which is usually intractable), if we are not able to use the kernel that eventually leaves us with dot products that can be implicitly expressed in the low dimensional input space R^N. Since Φ is fixed, we determine ω from the data by minimizing the sum of the empirical risk $R_{emp}[f]$ and a complexity term $\| \omega \|^2$, which enforces flatness in feature space

$$R_{reg}[f] = R_{emp}[f] + \lambda \| \omega \|^2 = \sum_{i=1}^{l} C(f(\bar{x}_i) - y_i) + \lambda \| \omega \|^2, \tag{2}$$

where l denotes the sample size $(\bar{x}_1, ..., \bar{x}_l)$, $C(.)$ is a loss function and λ is a regularization constant. For a large set of loss function, Equation (2) can be minimized by solving a quadratic programming problem, which is uniquely solvable. It can be shown that the vector ω can be written in terms of the data points

$$\omega = \sum_{i=1}^{l} (\alpha_i - \alpha^*) \Phi(\bar{x}_i), \tag{3}$$

with α_i, α_i^* being the solution of the aforementioned quadratic programming problem. α_i and α_i^* have an intuitive interpretation as forces pushing and pulling the estimate $f(\bar{x}_i)$ towards the measurements y_i. Taking Equation (3) and Equation (1) into account, we are able to rewrite the whole problem in terms of dot products in the low dimensional input space

$$\begin{aligned} \mathrm{f}(\bar{x}) &= \textstyle\sum_{i=1}^{l} (\alpha_i - \alpha_i^*)(\Phi(\bar{x}_i) \cdot \Phi(\bar{x})) + b \\ &= \textstyle\sum_{i=1}^{l} (\alpha_i - \alpha_i^*) K(\bar{x}_i, \bar{x}) + b, \end{aligned} \tag{4}$$

where α_i, α_i^* are Lagrangian multipliers, and $\bar{x}_i$ are support vectors.

In Equation (4), we introduce a kernel function $K(\bar{x}_i, \bar{x}_j) = \Phi(\bar{x}_i) \cdot \Phi(\bar{x}_j)$. As explained in, any symmetric kernel function K satisfying Mercer's condition corresponds to a dot product in some feature space.

After the off-line training process, we obtain the values for Lagrangian multipliers and support vectors of SVM. Let $\bar{x} = [x_1, x_2, ..., x_N]^T$ (x_i is an element of $\bar{x}$ and $\bar{x}_i$ is a sample data of $\bar{x}$). By expanding Equation (4), we know that $\mathrm{f}(\bar{x})$ is a *nonhomogeneous form* of degree d in $\bar{x} \in R^N$

$$\mathrm{f}(\bar{x}) = \sum_{0 \le i_1 + i_2 + ... + i_N \le d} c_j x_1^{i_1} x_2^{i_2} ... x_N^{i_N}, \tag{5}$$

where $i_1, i_2, ..., i_N$ are nonnegative integers, and $c_j \in R$ are weighting coefficients. j can be $1, 2, ..., M$, where $M = \binom{N+d}{N}$.

5 Experimental Results

In this experiment, we try to recognize whether there is any unauthorized person driving the car by analyzing the real time driving performance. In order to estimate the performance of the proposed system, we invite 8 human subjects to drive on our experimental system, who are Meng, Ou, Ye, Huang, Wang, Wu, Shi and Shen. The driving behavior data are collected and then classified through SVM into 2 group: Meng as the authorized driver and other 7 people as the unauthorized drivers.

Table 1. Test results using different sampling time and FFT order

Sampling time	FFT (order 5) (number of error/total)	FFT (order 10) (number of error/total)	FFT (order 20) (number of error/total)
10 sec.	147/419(authorized) 77/979(unauthorized)	139/419(authorized) 80/979(unauthorized)	148/419(authorized) 79/979(unauthorized)
20 sec.	70/209(authorized) 55/489(unauthorized)	70/209(authorized) 56/489(unauthorized)	83/209(authorized) 71/489(unauthorized)
40 sec.	32/104(authorized) 22/244(unauthorized)	32/104(authorized) 36/244(unauthorized)	22/104(authorized) 46/244(unauthorized)
80 sec.	19/52(authorized) 11/122(unauthorized)	23/52(authorized) 16/122(unauthorized)	25/52(authorized) 17/122(unauthorized)
160 sec.	9/26(authorized) 13/61(unauthorized)	12/26(authorized) 12/61(unauthorized)	6/26(authorized) 9/61(unauthorized)

Table 1 shows the test results of classification using Fast Fourier Transform data preprocessing. Different sampling times are compared in the experiment which causing the different size in each data segment for training and testing. The sampling rate is set at 10 Hz in the experiment. With a sampling time of 10 seconds, a 100×3 data segment can be produced, totally generating 419 segments from Meng's driving performance and 979 from others' for evaluation. Using FFT, segment data is reduced to 5×3, 10×3 and 20×3 segments in different FFT orders and the segments retrieved are aligned to form 1×15, 1×30 and 1×60 vectors for the recognition and learning by SVM.

In the experiment, sampling time between 10 seconds and 160 seconds, and FFT at 5 order, 10 order and 20 order are conducted. From Table I, comparing the success rate on both authorized and unauthorized drivers, sampling time at 40 seconds and FFT order of 20 achieves the best results. In addition, the data segment length of 40 seconds is efficient for not causing huge data segments for computation and providing adequate sampling frequency for testing driving performance in real time.

We compare the data preprocessing using PCA and ICA with FFT. As the aforementioned finding, we use sampling time of 40 seconds and FFT or order 20, transforming a 400×3 data segment into a 20×3 segment every 40 seconds, then aligned the retrieved data to a 1×60 vector. In order to compare with the

Table 2. Test results using different preprocessing methods

Preprocessing method	Errors of authorized drivers	Errors of unauthorized drivers
FFT	22	46
PCA	39	82
ICA	47	79

data format of FFT, we align the original data from 400×3 to 40×30, then two components are extracted using PCA and ICA. With 2 PCs and 2 ICs, a 2×30 feature retrieved are aligned to form a 1×60 vector as input to the SVM.

Table 2 shows the test results using different data preprocessing methods. With the same sampling time of 40 seconds, each 400×3 data segment is transformed to a 1×60 vector through FFT, PCA and ICA, producing 104 test samples of Meng's driving performance and 244 of others. Then the retrieved vector is trained in the SVM and the results are listed in Table II. The data preprocessing method based on FFT is found to give the best data reduction results compared to the other two processing methods presented. By reducing the redundancy in the input data, the training process of the human driving behavior model becomes more efficient.

Table 3. Test results using different SVM kernels and parameters

Kernel	Order	Alpha	Gamma	Number of errors
Vapnik's Poly.	3	10	N/A	22(auth.) / 46(unauth.)
Vapnik's Poly.	3	100	N/A	30(auth.) / 39(unauth.)
Vapnik's Poly.	3	1	N/A	27(auth.) / 36(unauth.)
Vapnik's Poly.	2	10	N/A	30(auth.) / 30(unauth.)
Vapnik's Poly.	4	10	N/A	31(auth.) / 38(unauth.)
Simple Dot Pro.	N/A	N/A	N/A	37(auth.) / 22(unauth.)
RBF	N/A	N/A	1	32(auth.) / 22(unauth.)
RBF	N/A	N/A	5	33(auth.) / 30(unauth.)
RBF	N/A	N/A	10	33(auth.) / 36(unauth.)

Table 3 shows the test results using different SVM kernels and parameters at sampling time of 40s and FFT order of 20. From the result, it is found that the recognition performance reaches optimal with the use of Vapnik's Polynomial Kernel (Order = 3 and Alpha = 10).

Then we utilize the aforementioned methods in our practical intelligent vehicle security system. Table 4 list the test result on individuals in the practical system. It can be seen that there are 104 data segments of Meng's driving behavior applied in the evaluation process, 82 of them are identified as the authorized driver (Meng himself) and the success rate is 78.85%. And 244 data segments collected from 7 testers are used for the test and 198 of them are classified as the unauthorized drivers successfully.

Table 4. Test results on individuals

No.	Name	Totals	Correct	Failed	Successful rate
1	Meng	104	82	22	78.85%
2	Ou	47	44	3	93.62%
3	Ye	48	37	11	77.08%
4	Huang	26	22	4	84.62%
5	Wang	29	23	6	79.31%
6	Wu	30	27	3	90.00%
7	Shi	45	28	17	62.22%
8	Shen	19	17	2	89.47%
TOTAL		244	198	46	81.15%

6 Discussion and Conclusions

This paper has addressed an intelligent vehicle security system towards the solving of vehicle thefts problem. By capturing and analyzing human driving behaviors, unauthorized drivers can be recognized while driving illegally and alarm can be delivered. The advantage of the proposed methodology lies in the fact that the dynamic biometrical features involved in human driving behavior are unique and more secure than static features.

In this paper, we have built an experimental system for capturing and analyzing human driving behaviors in a simulated driving environment. Data is collected from several testing subjects and then processed through fast fourier transform (FFT), principal component analysis (PCA) and independent component analysis (ICA) for data preprocessing and support vector machine (SVM) is applied as intelligent classifier through leaning. Finally, experiments for evaluating the performance of the proposed system are conducted and the results verify that the proposed method is valid and useful against the vehicle thefts problem with a success rate of around 80%.

References

1. Asada, H., Liu, S.: Transfer of Human Skills to Neural Net Robot Controllers. Proc. IEEE Int. Conf. on Robotics and Automation, Vol. 3. (1991) 2442-2447
2. Pomerleau, D. A.: Neural Network Perception for Mobile Robot Guidance. Kluwer Academic Publishing, Boston (1994)
3. Nechyba, M., Xu, Y.: Learning and Transfer of Human Real-time Control Strategies. Int. J. Advanced Computational Intelligence 1 (1997) 137-154
4. Nechyba, M., Xu, Y.: Stochastic Similarity for Validating Human Control Strategy Models. IEEE Trans. Robotics and Automation 14 (1998) 437-451
5. Hyvarinen, A.: Fast and Robust Fixed-point Algorithms for Independent Component Analysis. IEEE Trans. Neural Networks 10 (1999) 626-634
6. Giannakopoulos, X., et al.: Experimental Comparison of Neural ICA Algorithms. Proc. Int. Conf. on Artificial Neural Networks, (1998) 651-656

Adaptive Neural Network Control of Helicopters

Shuzhi Sam Ge* and Keng-Peng Tee

Department of Electrical & Computer Engineering,
National University of Singapore, Singapore 117576
elegesz@nus.edu.sg

Abstract. In this paper, we propose robust adaptive neural network (NN) control for helicopter systems by using the Implicit Function Theorem and the Mean Value Theorem, which are useful tools for handling nonlinear nonaffine systems. We focus on single-input single-output (SISO) helicopter systems, which are exemplified by certain single-channel modes of operation, such as vertical flight and pitch regulation, and also by special conditions under which the multiple channels become decoupled. It is shown that under the proposed NN control, the output tracking error converges to a small neighbourhood of the origin, while all closed loop signals are Semi-Globally Uniformly Ultimately Bounded (SGUUB).

1 Introduction

Unmanned helicopters pose a challenge to nonlinear control design and development. Besides the problem of strong nonlinear couplings between system states, helicopters are also inherently unstable, unlike many classes of mechanical systems.

Based on dynamic models, various nonlinear control techniques have been applied to ensure stable control of unmanned helicopters. For altitude regulation, dynamic sliding mode and feedback linearization methods were proposed [1, 2]. In [3], approximate input-output linearization yielded a dynamically linearizable system without zero dynamics, which possesses the desirable property of differential flatness. Using internal model based control, the problem of landing on an oscillating ship deck was tackled [4].

The foregoing works require reasonably precise knowledge of the dynamic models to achieve satisfactory performance. To deal with the presence of model uncertainties, approximation based techniques using neural networks have been proposed. Approximate dynamic inversion with augmented neural networks was proposed to handle unmodelled dynamics in [5, 6], while neural dynamic programming was shown to be effective for tracking and trimming control of multi-input multi-output (MIMO) helicopters in [7].

Motivated by recent development in NN control of nonlinear systems [8, 9, 10], we use Lyapunov-based design techniques to construct robust adaptive NN control for helicopters, with guaranteed stability and performance bounds. The Implicit Function Theorem and the Mean Value Theorem are used as tools for handling nonaffine nonlinearities in the helicopter dynamics.

* Corresponding author.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 82–87, 2006.

2 Problem Formulation and Preliminaries

Consider the class of SISO helicopter systems described by the following differential equations nonaffine in the control:

$$\dot{x} = f(x, u) \ , \quad y = h(x) \ , \tag{1}$$

where $x \in R^n$ are the states of the system; $u, y \in R$ denote the input and output respectively; and $f : R^n \times R \to R^n$ is an unknown function. SISO systems in helicopters can be found in certain single-channel modes of operation, such as vertical flight and pitch regulation, and also by special conditions under which the multiple channels become decoupled.

The control objective is to achieve practical output tracking of a desired reference trajectory, i.e. the tracking error converges to a neighborhood of zero, while all closed loop signals remain bounded.

Assumption 1. *The reference trajectory $y_d(t)$ and its ρ derivatives are bounded, i.e., $x_d \in \Omega_d$, $\forall t \geq 0$, where ρ is the relative degree of (1).*

Assumption 2. *System (1) is input-output linearizable with strong relative degree $\rho < n$.*

Define $\phi_j(x) = L_f^{j-1} h$ for $j = 1, ..., \rho$, where $L_f h$ denotes the Lie derivative of the function $h(x)$ with respect to the vector field $f(x, u)$. Due to Assumption 2, it can be shown that there exist mappings $\xi_j = \phi_j(x)$ such that (1) can be expressed in the normal form:

$$\dot{\eta} = q(\eta, \xi) \ , \quad \dot{\xi}_j = \xi_{j+1} \ , \quad \dot{\xi}_\rho = b(\xi, \eta, u) \ , \quad y = \xi_1 \ , \tag{2}$$

where $j = 1, ..., \rho - 1$, $b(\xi, \eta, u) := L_f^\rho h$ and $x = \Phi^{-1}(\xi, n)$, for $(\xi, \eta, u) \in \bar{U} := \{(\xi, \eta, u) | (\xi, \eta) \in \Phi(\Omega_x); u \in \Omega_u\}$. Throughout this paper, we use the notation $g(x, u) := \frac{\partial b(\xi, \eta, u)}{\partial u}$.

Assumption 3. *The zero dynamics of system (2), given by $\dot{\eta} = q(0, \eta)$ are exponentially stable. In addition, the function $q(\xi, \eta)$ is Lipshitz in ξ.*

Assumption 4. *There exist smooth functions $\bar{g}(\xi, \eta)$ and a positive constant $d > 0$, such that $\bar{g}(\xi, \eta) \geq |g(\xi, \eta, u)| \geq d > 0$ holds for all $(\xi, \eta, u) \in \bar{U}$. Without loss of generality, it is further assumed that the sign of $g(\xi, \eta, u)$ is positive for all $(\xi, \eta, u) \in \bar{U}$.*

Lemma 1. *(Implicit Function Theorem) [8] For a continuously differentiable function $b(\xi, \eta, u) : R^n \times R \to R$ satisfying Assumption 4, there exists a continuous (smooth) function $u^* = u(\xi, \eta)$ such that $b(\xi, \eta, u^*) = 0$.*

Assumption 5. *There exist a positive function $g_0(\xi, \eta)$ such that $|\dot{g}(\xi, \eta, u)| \leq 2g(\xi, \eta, u)g_0(\xi, \eta)$, $\forall (\xi, \eta, u) \in \bar{U}$.*

In this paper, we consider nonlinearly-parameterized Multi-Layer Neural Networks (MNN), which are used to approximate the continuous function $f(Z) : R^q \rightarrow R$ as follows:

$$f(Z) = W^T S(V^T Z) + \varepsilon(Z) \ , \tag{3}$$

where the vector $Z = [\zeta_1, \zeta_2, \cdots, \zeta_m, 1]^T \in R^{m+1}$ are the input variables to the neural networks; $S(\cdot) \in R^l$ is a vector of known continuous basis functions; $W \in R^l$ and $V \in R^{(m+1)\times l}$ are adaptable weights; and ε is the approximation error which is bounded over the compact set, i.e., $|\varepsilon(Z)| \leq \bar{\varepsilon}$, $\forall Z \in \Omega_Z$ where $\bar{\varepsilon} > 0$ is an unknown constant. The ideal weights W^* and V^* are defined as the values of W and V that minimize $|\varepsilon(Z)|$ for all $Z \in \Omega_Z \subset R^{m+1}$.

Lemma 2. *[9] The approximation error can be expressed as*

$$\hat{W}^T S(V^T Z) - W^{*T} S(V^{*T} Z) = \tilde{W}^T(\hat{S} - \hat{S}'\hat{V}^T Z) + \hat{W}^T \hat{S}' \tilde{V}^T Z + d_u \ , \tag{4}$$

where $\hat{S} := S(\hat{V}^T Z)$; $\tilde{W} := \hat{W} - W^*$ and $\tilde{V} := \hat{V} - V^*$ are the weights estimation errors; $\hat{S}' := diag\{\hat{s}_1', ..., \hat{s}_l'\}$ with $\hat{s}_i' = s'(\hat{v}_i^T Z) = \frac{ds(z_a)}{dz_a}\Big|_{z_a = \hat{v}_i^T Z}$, , i=1,2,...,l, and the residual term is bounded by

$$d_u \leq \|W^*\|\|\hat{S}'\hat{V}^T Z\| + \|V^*\|_F \|Z\hat{W}^T\hat{S}'\|_F + \|W^*\| \ . \tag{5}$$

3 Control Design

We employ backstepping for the ξ subsystem, and then make use of the exponential stability of the zero dynamics to show that the overall closed loop system is stable and that output tracking is achieved.

Step 1: Let $z_1(t) = \xi_1(t) - y_d(t)$ and $z_2(t) = \xi_2(t) - \alpha_1(t)$, where $\alpha_1(t)$ is a virtual control function to be determined. Define quadratic function $V_1 = \frac{1}{2} z_1^2$. Choosing the virtual control $\alpha_1 = -k_1 z_1 + \dot{y}_d$, we can show that $\dot{V}_1 = -k_1 z_1 + z_1 z_2$, where the term $z_1 z_2$ will be cancelled in the subsequent step.

Step i ($\mathbf{i = 2}, ..., \rho - \mathbf{1}$): Let $z_i(t) = \xi_i(t) - \alpha_{i-1}(t)$. Define quadratic function $V_i = V_{i-1} + \frac{1}{2} z_i^2$. Choosing the virtual control $\alpha_i = -k_i z_i - z_{i-1} + \dot{\alpha}_{i-1}$, we can show that $\dot{V}_i = -\sum_{j=1}^{i} k_j z_j^2 + z_i z_{i+1}$, where the term $z_i z_{i+1}$ will be cancelled in the subsequent steps.

Step ρ: This is the final step where the actual control law u will be designed. Let $z_\rho(t) = \xi_\rho(t) - \alpha_{\rho-1}(t)$. From Assumption 4, we know that $g(\xi, \eta, u) \geq d > 0$ for all $(\xi, \eta, u) \in R^{n+1}$. Define $\nu := -\dot{\alpha}_\rho + g_0(\xi, \eta) z_\rho$, from which $\frac{\partial \nu}{\partial u} = 0$, yielding the inequality $\frac{\partial [b(\xi, \eta, u) + \nu]}{\partial u} \geq d > 0$. According to Lemma 1, for every value of ξ, η and ν, there exists a smooth ideal control input $u^* \in R$ such that $b(\xi, \eta, u^*) + \nu = 0$.

Using the Mean Value Theorem [11], there exists $(0 < \lambda < 1)$ such that

$$b(\xi, \eta, u) = b(\xi, \eta, u^*) + g_\lambda (u - u^*) \ , \tag{6}$$

where $g_\lambda := g(\xi, \eta, u_\lambda)$. Then, the derivative of z_ρ can be written as

$$\dot{z}_\rho = -g_0(\xi, \eta) z_\rho + g_\lambda (u - u^*) \ . \tag{7}$$

We employ a robust MNN controller of the form:

$$u = \hat{W}^T S(\hat{V}^T Z) - k_\rho z_\rho - z_{\rho-1} - k_b \left(\left\| Z\hat{W}^T S' \right\|_F^2 + \left\| S' \hat{V}^T Z \right\|^2 \right) z_\rho \ , \quad (8)$$

where $\hat{W}^T S(\hat{V}^T Z)$ approximates $W^{*T} S(V^{*T} Z) = u^* - \varepsilon$, where $Z = [\xi, \eta, z_\rho, \dot{\alpha}_{\rho-1}]^T \in \Omega \subset R^{n+2}$, and $|\varepsilon| \leq \bar{\varepsilon}$ is the approximation error with constant $\bar{\varepsilon} > 0$.

For stability analysis, consider the Lyapunov function candidate:

$$V_\rho = V_{\rho-1} + \frac{1}{2g_\lambda} z_\rho^2 + \frac{1}{2} \tilde{W}^T \Gamma_W^{-1} \tilde{W} + \frac{1}{2} \mathrm{tr} \left\{ \tilde{V}^T \Gamma_V^{-1} \tilde{V} \right\} \ , \quad (9)$$

where $\tilde{W} := \hat{W} - W^*$ and $\tilde{V} := \hat{V} - V^*$, and the following adaptation laws

$$\dot{\hat{W}} = -\Gamma_W \left[(\hat{S} - \hat{S}' \hat{V}^T Z) z_\rho + \sigma_W \hat{W} \right], \ \dot{\hat{V}} = -\Gamma_V \left[Z\hat{W}^T \hat{S}' z_\rho + \sigma_V \hat{V} \right] \ , \quad (10)$$

where $\Gamma_W = \Gamma_W^T > 0$, $\Gamma_V = \Gamma_V^T > 0$, $\sigma_W > 0$ and $\sigma_V > 0$ are constant design parameters. Then, substituting (8) and (10) into the derivative of V_ρ, and by completion of squares, together with Lemma 2, it can be shown that

$$\begin{aligned} \dot{V}_\rho \leq & -\sum_{j=1}^{\rho-1} k_j z_j^2 - \left(g_\lambda (k_\rho - 1) + g_0 + \frac{\dot{g}_\lambda}{2g_\lambda} \right) \frac{z_\rho^2}{g_\lambda} \\ & - \left(k_b - \frac{1}{2} \right) \left(\left\| \hat{S}' \hat{V}^T Z \right\|^2 + \left\| Z\hat{W}^T \hat{S}' \right\|_F^2 \right) z_\rho^2 \\ & - \frac{\sigma_W}{2} \left\| \tilde{W} \right\|^2 - \frac{\sigma_V}{2} \left\| \tilde{V} \right\|_F^2 + \frac{1}{2} \bar{\varepsilon}^2 + \frac{\sigma_W + 2}{2} \left\| W^* \right\|^2 + \frac{\sigma_V + 1}{2} \left\| V^* \right\|_F^2 \ . \end{aligned} \quad (11)$$

From Assumption 5, we know that $\left(g_0 + \frac{\dot{g}_\lambda}{2g_\lambda} \right) \geq 0$. Hence, by choosing the control parameters $k_\rho > 1$ and $k_b > 1/2$, it can be shown that

$$\dot{V}_\rho \leq -c_1 V_\rho + c_2 \ , \quad (12)$$

$$c_1 = \min \left\{ k_1, \ k_2, ..., \ k_{\rho-1}, \ k_\rho - 1, \ \frac{\sigma_W}{2}, \ \frac{\sigma_V}{2} \right\} \ , \quad (13)$$

$$c_2 = \frac{1}{2} \bar{\varepsilon}^2 + \frac{\sigma_W + 2}{2} \left\| W^* \right\|^2 + \frac{\sigma_V + 1}{2} \left\| V^* \right\|_F^2 \ . \quad (14)$$

Lemma 3. *[9] Given that Assumptions 1 and 3 are satisfied, there exist positive constants a_1, a_2 and T_0 such that the trajectories $\eta(t)$ of the internal dynamics satisfy $\|\eta(t)\| \leq a_1(\|z(t)\| + \|\xi_d(t)\|) + a_2$, $\forall\ t > T_0$.*

Theorem 1. *Consider the SISO helicopter dynamics (1) satisfying Assumptions 1-5, with control law (8) and adaptation laws (10). For initial conditions $\xi(0), \eta(0), \tilde{W}(0), \tilde{V}(0)$ belonging to any compact set Ω_0, all closed loop signals are Semi-Globally Uniformly Ultimately Bounded (SGUUB), and the tracking error $z_1 = y - y_d$ converges to the compact set $\Omega_{z_1} := \left\{ z_1 \in R \left| |z_1| \leq \sqrt{\frac{2c_2}{c_1}} \right. \right\}$.*

Proof: According to Lemma 1.1-1.2 in [10], we know from (12) that z, $\tilde{W}$ and $\tilde{V}$ are bounded within compact sets. Since W^* and V^* are bounded, it is obvious that $\hat{W}$ and $\hat{V}$ are also bounded. From the fact that $z,\ y_d,\ y_d^{(1)},...,\ y_d^{(\rho)}$ are bounded, we know that the virtual controls $\alpha_i(\bar{z}_i, y_d, y_d^{(1)},...,\ y_d^{(i)})$, for $i = 1, 2, ..., \rho$, are bounded. Hence, there exists a constant $a_3 > 0$ such that $\|\xi_d\| \leq a_3$.

From Lemma 3, it can be seen that η is bounded if both z and ξ_d are bounded. As a result, we can conclude that the states of the internal dynamics will converge to the compact set $\Omega_\eta := \left\{\eta \in R^{n-\rho} \left| \|\eta\| \leq a_1 \left(\sqrt{\frac{2c_2}{c_1}} + a_3\right) + a_2 \right.\right\}$. Since the control signal $u(t)$ is a function of the weights $\hat{W}(t)$, $\hat{V}(t)$ and the states $\xi(t)$, $\eta(t)$, we know that it is also bounded. Therefore, we have shown that all the closed loop signals are SGUUB. To show that the tracking error $z_1 = y - y_d$ converges to the compact set Ω_{z_1}, we multiply (12) by $e^{c_1 t}$ and integrate over [0,t] to obtain that $|z(t)| \leq \sqrt{2(V_\rho(0) + \frac{c_2}{c_1})e^{-c_1 t} + 2\frac{c_2}{c_1}}$, from which it is easy to see that $|z_1(t)| \leq \sqrt{\frac{2c_2}{c_1}}$ as $t \to \infty$. ■

4 Simulation

In Section 2, we have considered a general representation of helicopters as nonaffine nonlinear systems. While it would be ideal to perform simulation on a nonaffine helicopter model, an accurate model is difficult to obtain. Since the class of nonaffine systems include linear systems as special cases, we shall apply our proposed adaptive NN control for general nonlinear systems to linear helicopter models, which are widely available in the literature. In particular, the effectiveness of the control scheme will be investigated on the linearized model of

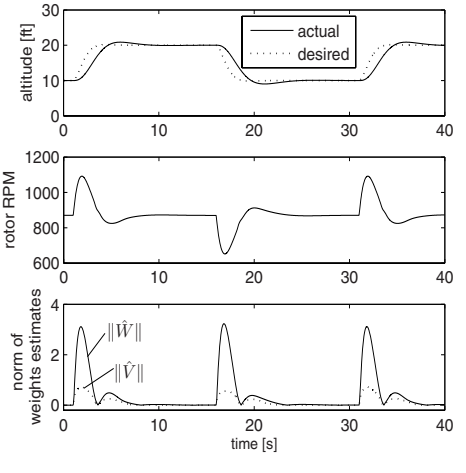


Fig. 1. Altitude tracking

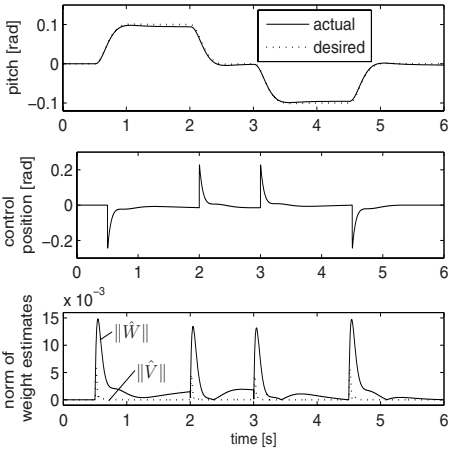


Fig. 2. Pitch tracking

the Yamaha R50 helicopter for two tasks: altitude tracking and pitch tracking. We use an MNN with identical activation functions in the hidden layer described by $s_i(a_i) = \frac{1}{1+e^{-a_i}}$, $i = 1, 2, ..., l$ so that $S(a) = [s_1(a_1), s_2(a_2), ..., s_l(a_l)]^T$.

For altitude tracking, we consider the model as detailed in [5], with the longitudinal cyclic input δ set to zero, and control parameters set as $k_1 = 1.0$, $k_2 = 1.5$, and $k_b = 1.0$. For pitch tracking, the model considered contains actuator dynamics, as detailed in [6], with control parameters set as $k_1 = 1.0$, $k_2 = 1.0$, $k_3 = 1.5$, and $k_b = 1.0$. The adaptation parameters for both cases are identically chosen as $\Gamma_W = 5I$, $\Gamma_V = 50I$, and $\sigma_W = \sigma_V = 5$.

It can be seen in both Figures 1 and 2 that the tracking performance in both tasks is satisfactory. The actual output remains in a small neighborhood of the desired output, while the control signal and neural weights are bounded.

References

1. Sira-Ramirez, H., Zribi, M., Ahmad, S.: Dynamical Sliding Mode Control Approach for Vertical Flight Regulation in Helicopters. IEE Proc. - Control Theory Appl. **141** (1994) 19–24
2. Vilchis, J.A., Brogliato, B., Dzul, A., Lozano, R.: Nonlinear Modelling and Control of Helicopters. Automatica **39** (2003) 1583–1596
3. Koo, T.J., Sastry, S.: Output Tracking Control Design of a Helicopter Model Based on Approximate Linearization. In: Proc. 37th IEEE Conf. Decision & Control, Tampa, Florida USA (1998) 3635–3640
4. Isidori, A., Marconi, L., Serrani, A.: Robust Nonlinear Motion Control of a Helicopter. IEEE Trans. Automatic Control **48** (2003) 413–426
5. Kim, N., Calise, A., Corban, J., Prasad, J.: Adaptive Output Feedback for Altitude Control of an Unmanned Helicopter Using Rotor RPM. In: Proc. AIAA Guidance, Navigation and Control Conference. (2004) 3635–3640
6. Hovakimyan, N., Nardi, F., Calise, A., Kim, N.: Adaptive Output Feedback Control of Uncertain Nonlinear Systems Using Single-Hidden-Layer Neural Networks. IEEE Trans. Neural Networks **13** (2002) 1420–1431
7. Enns, R., Si, J.: Helicopter Trimming and Tracking Control Using Direct Neural Dynamic Programming. IEEE Trans. Neural Networks **14** (2003) 929–939
8. Ge, S.S., Wang, C.: Adaptive NN Control of Uncertain Nonlinear Pure-Feedback System. Automatica **38** (2002) 671–682
9. Ge, S.S., Zhang, J.: Neural Network Control of Nonaffine Nonlinear System with Zero Dynamics by State and Output Feedback. IEEE Trans. Neural Networks **14** (2003) 900–918
10. Ge, S.S., Wang, C.: Adaptive Neural Control of Uncertain MIMO Nonlinear Systems. IEEE Trans. Neural Networks **15** (2004) 674–692
11. Apostol, T.M.: Mathematical Analysis. 2nd edn. Addison-Wesley, Reading, Mass (1974)

Modified Hopfield Neural Network for CDMA Multiuser Detection

Xiangdong Liu, Xuexia Wang, Zhilu Wu, and Xuemai Gu

School of Electronics and Information Technology, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China
htdzl@sina.com, {wangxuexia, wuzhilu, guxuemai}@hit.edu.cn

Abstract. We investigate the application of modified Hopfield neural networks (HNNs) based on Annealing Techniques to the problem of multiuser detection in spread spectrum CDMA communication systems. It is shown that the NP-complete problem of minimizing the objective function of the optimal multiuser detector (OMD) can be translated into minimizing an HNN "energy" function, thus allowed to take advantage of the ability of HNNs to perform very fast gradient descent algorithms in analog hardware. The performance of the proposed HNN receiver is evaluated via computer simulations and compared to that of the general HNN as well as to that of the OMD for CDMA transmission cases. It is shown that the modified HNN detection scheme exhibits a number of attractive properties and that it provides in fact more powerful performance than the general HNN scheme or the OMD scheme.

1 Introduction

Code Division Multiple Access (CDMA) have reached a large development as a promising multiple access technique for 3rd generation communication systems. This technique permits a high number of users communicating simultaneously on the same frequency band. Nevertheless, the orthogonality between users codes at the receiver is lost and the high-power users can significantly corrupt the received signals of low-power users (near-far problem), which leads to performance degradation of the detection. To overcome this problem, the concept of multiuser detection (MUD) is needed and become essential structure in wireless communications.

Sergio Verdu has proposed the optimal MUD [l] and has proven that his algorithm deals with the minimization of a quadratic form. However, its complexity is on the order of $o(2^k)$ for k active users. To avoid the complexity of implementing this algorithm, several sub-optimal linear and nonlinear solutions have been proposed such as MMSE detector, Successive Interference Cancellation (SIC) detector, Parallel Interference Cancellation (PIC) detector, Multistage detector and based on this later the Hopfield neural network (HNN) detector.

Neural networks are known for their robustness in high computational processing with simple nonlinear processors. However, due to local optimization, the neural structure does not assure good performance of detection and many authors have proposed modified structure to assure more performance to the algorithm [2-4]. Yoon,

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 88 – 93, 2006.

Chen et a1 proposed the annealed neural network detector. Another hybrid structure was proposed by Kechriotis and Manolakos. In this paper, we discuss the performance of the modified HNN where a noise term is added to the Hopfield iterations in goal to avoid the local minima and assure more and more performance for multiuser detection.

2 Optimal Multiuser Detector Scheme

Assume that K active transmitters share the same Gaussian channel at a given time instance. A signature waveform $s_k(t)$, time limited in the interval $t \in [0, T]$, is assigned to each transmitter. Denote the ith information bit of the kth user as $b_k^{(i)} \in \{+1, -1\}$. In a general CDMA system, the signal at a receiver is the superposition of K transmitted signals and additive channel noise

$$r(t) = \sum_{i=-P}^{P} \sum_{k=1}^{K} b_k^{(i)} s_k (t - iT - \tau_k) + n(t), \ t \in R . \tag{1}$$

In (1), $\tau_k \in [0, T)$ are the relative time delays between the users and $2P+1$ is the packet size. In case that the stations cooperate to maintain synchronism, it holds that $\tau_k=0$, $k=1, \ldots, K$. The ith information bit of the kth user based on the sign of the ith output of the kth matched filter yk, shown as function (2), where $y^{(i)}= [y_0^{(i)} y_1^{(i)} \ldots y_{K-1}^{(i)}]^T$.

$$y_k^{(i)} = \int_{iT-\tau_k}^{(i+1)T-\tau_k} r(t) s_k (t - iT - \tau_k) dt . \tag{2}$$

The Optimal Multiuser Detector (OMD) produces an estimate for the information vector transmitted at the discrete time instant i, based on the maximization of the logarithm of the likelihood function. In a CDMA transmission case it, holds [5]:

$$b_{OMD}^{(i)} = \arg \max_{b \in \{-1,+1\}^K} \left\{ 2 y^{(i)^T} b - b^T \mathbf{H} b \right\}, \tag{3}$$

where $\mathbf{H} \in R^{K \times K}$ is the matrix of signal waveform cross-correlations.

3 Modified Hopfield Neural Network for MUD

Because of the exponential growth of the computational complexity of the OMD with the number of active users, many other multiuser detection schemes have been proposed. Modified HNN is discussed in this paper.

3.1 Hopfeild Neural Network

HNNs are single layer networks with output feedback consisting of simple processors (neurons) that can collectively provide good solutions to difficult optimization problems. A connection between two processors is established through a conductance T_{ij} which transforms the voltage outputs of amplifier j to a current input for amplifier i. Externally supplied bias currents I_i are also present in every processor j [6].

Each neuron i receives a weighted sum of the activations of other neurons in the network, and updates its activation according to the rule

$$V_i = g(U_i) = g(\sum_{j \neq i} T_{ij} V_j + I_i) \,. \tag{4}$$

The function $g(U_i)$ can be either a binary or antipodal thresholding function for the case of the McCulloch-Pitts neurons

$$V_i = g(U_i) = sign(U_i) \,, \tag{5}$$

or any monotonically increasing nonlinear function. One example of such a nonlinear function often used in simulations is the sigmoid function, defined by

$$V_i = g(U_i) = sigm(\alpha U_i) = \frac{1 - e^{-\alpha U_i}}{1 + e^{-\alpha U_i}} \,, \tag{6}$$

where α is a positive constant that controls the slope of the nonlinearity. In particular, when $\alpha \to \infty$, then $g(U_i) \to sign(U_i)$.

It has been shown by Hopfield that, in the case of symmetric connections ($T_{ij} = T_{ji}$), the equations of motion for the activation of the neurons of a HNN always lead to convergence to a stable state, in which the output voltages of all the amplifiers remain constant. Also, when the diagonal elements (T_{ij}) are zero and the width of the amplifier gain curve is narrow, (i.e., the nonlinear activation function $g(\cdot)$ approaches the antipodal thresholding function), the stable states of a network of N neuron units are the local minima of the quantity (energy function)

$$E = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} T_{ij} V_i V_j - \sum_{i=1}^{N} V_i I_i \,. \tag{7}$$

The equations of motion for the ith neuron may be described in terms of the energy function (7) as follows:

$$\frac{dU_i}{dt} = -\frac{\partial E}{\partial V_i} - \frac{U_i}{\tau} = -\frac{U_i}{\tau} + \sum_{i \neq j} T_{ij} + I_i \,, \tag{8}$$

where $\tau = RC$ is the time constant connected to neuron i. With the exception of pathological cases, when the matrix $\mathbf{T}$ is negative or positive definite, networks with vanishing diagonal elements have minima only at the corners of the N-dimensional hypercube $[-1 \; +1]^N$. HNNs have been employed extensively to solve a variety of difficult combinatorial optimization problems.

If we write (7) using matrix notation, then the recurrent network maximizes the following quadratic (9), where $\mathbf{V}(k)=\{Y_i(k)\}$, $\mathbf{T}=\{T_{ij}\}$ is the weight matrix and $\mathbf{I} = [I_j]$.

$$J(\mathbf{V}(k)) = \mathbf{V}^T(k)\mathbf{T}\mathbf{V}(k) - 2\mathbf{I}^T\mathbf{V}(k) \,, \tag{9}$$

One may note the similarity between the objective function (3) of the optimal detector and the HNN energy function (9).

3.2 Stochastic Neural Model (Based on Annealing Techniques)

Kechriotis et *al.* [4] has shown that Hopfield detector outperforms the conventional detector and approaches the performance of the OMD. However this optimization process is localized and therefore the detector can get stuck in a local minimum resulting in a bad steady state**.** In [7] the authors proposed a stochastic Hopfield network (SHN) which avoids local minima. They achieve this by adding a noise term in the state transition rule of (10), given as:

$$V_i(k+1) = g\left(\sum_{j=1}^{p} T_{ij}V_j(k) + I_i + i(k)\right), \tag{10}$$

with the distribution given by (11) below, the global optimum has maximal stationary probability.

$$F(x) = \frac{1}{1+e^{-\alpha x}}. \tag{11}$$

In determining the annealing schedule or change in $\alpha(k)$**,** a tradeoff must be made between convergence and bit-error-rate performance. If $\alpha(k)$ is increased quickly over the iterations, the network converges faster at the expense of performance. If $\alpha(k)$ is increased very slowly, the performance is near optimum but this occurs over a large number of iterations. Our current work is based on a network that utilizes a stochastic neuron model and our aim is ideally to decrease the dependence of the network on $\alpha(k)$ or alternatively decrease the number of iterations required to achieve near-optimal performance. Stochastic neurons are used to take into account the noise generated by random fluctuations and other probabilistic causes that may occur during synaptic transmission. In [7] the authors made a similar argument and proposed the introduction of randomly distributed noise in the state transition rule (10) described above. A traditional method of accounting for noise effects is to introduce a probabilistic mechanism in the firing of neurons. Such a method combines the properties of both annealed and stochastic networks. We have used this type of neuron to propose a model which is a probability dependent HNN. The noise may be represented by thermal fluctuation that reduces with the number of iterations. Thus the deterministic rule of (10) is altered as follows:

$$V_i(k+1) = +1 \text{ with probability } P\left(\sum_{j=1}^{p} T_{ij}V_j(k) + I\right), \tag{12}$$

$$V_i(k+1) = -1 \text{ with probability } 1 - P\left(\sum_{j=1}^{p} T_{ij}V_j(k) + I\right). \tag{13}$$

The function P is defined as:

$$P(x) = \frac{1}{1+\exp(-2x/T_a)}. \tag{14}$$

T_a is the pseudo temperature that controls the noise level. As T_a decreases the level of noise is reduced and the probability of firing is increased.

3.3 Multiuser Detector with Hopfield Network

A block diagram of the multiuser detector with the Hopfield network is shown in Figure 1. With all the amplifier inputs u_k, k=1, 2, …, K, initiated at zero, switches SW_k, k=1, 2, …, K simultaneously closed, the network is allowed to settle. The output of each neural amplifier is applied to the input of a decision device which outputs a +1 or -1 for a positive or negative input, respectively. Once the network is settled, the estimated sequence is read at the output of the decision as shown in Figure 1.

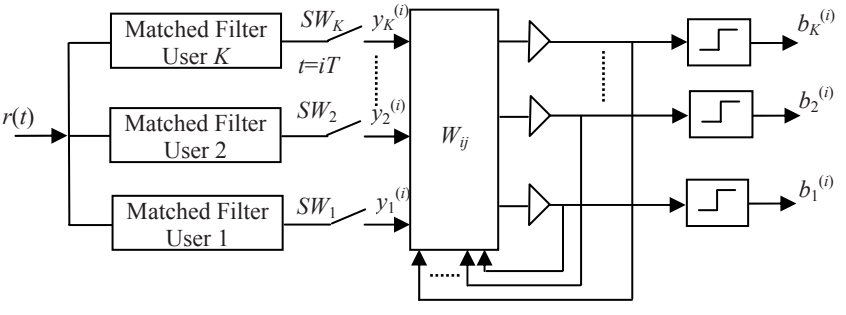


Fig. 1. Multiuser detector with Hopfield network

4 Simulation Results

We use a network size of 60 neurons to store 8 vector patterns of length 60 each. Errors are introduced uniformly in the desired vector pattern. The stochastic based networks require a fewer number of iterations to classify the pattern, and notably the stochastic neuron models follow the same search path. The simulation environment used in this paper is added white Gaussian noise (AWGN). The chip sequence is derived using a Pulse Amplitude Modulation (PAM) waveform. The spreading codes are taken from the set of length-31Gold Codes, giving a processing gain of N=31.

The bit-error rate (BER) versus signal-noise ratio (SNR) for a single user is depicted in Fig.2. The performance of OMD, general HNN and the modified HNN shows that the modified HNN detector achieves acceptable performance with fewer calculations than the OMD detector and has better performance than the general HNN detector. We also deal with the near-far problem by applying normally distributed power for each user and varying the power deviation of users. Simulation results of the BER versus near-far ratio (NFR) are shown in Fig.3. From the simulation results, we can see that the modified HNN detector is clearly superior to the general HNN detector and comparable with the OMD detector. Therefore we believe that overall modified HNN schemes are more suitable as neural network based multiuser CDMA receiver structures, and we expect them to find many applications in high-speed communications.

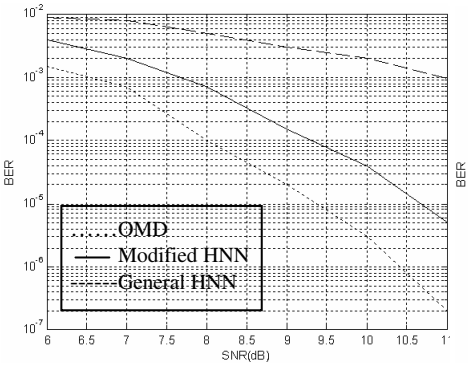


Fig. 2. Bit-Error Rate vs. Signal-Noise Ratio

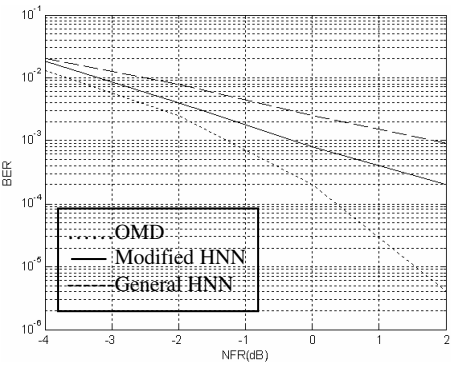


Fig. 3. Bit-Error Rate vs. Near-Far Ratio

5 Conclusions

In this paper, a Hopfield model based on stochastic neurons, for use in multiuser detection, was proposed. These neurons utilize statistical physics to help overcome the problem of local minimization. The classification performance of the proposed scheme has been shown to be comparable to that of the general HNN and OMD schemes through simulation of pattern storage and retrieval. Future work-will include implementation if a partially connected network to investigate the performance of the stochastic detectors in the AWGN environment.

References

1. Verdu, S.: Multiuser Detection. Cambridge Univ. Press (1984)
2. Yoon, S.H., Rao, S.: Multiuser detection in CDMA Based on the Annealed Neural Network. IEEE Int. Conf. Neural Networks **4** (1996) 2124-2129
3. Chen, D.C., Sheu, B.J.: A Compact Neural network Based CDMA Receiver for Multimedia Wireless Communication. IEEE Conf. Comp. Design (1996) 99-103
4. Kechriotis, G.I., Manolakos, E.S.: Implementing the Optimal CDMA Multiuser Detector with Hopfield Neural network. IEEE Workshop on Appl. Neural Networks and telecomm. (1993) 60-66
5. Verdu, S.: Computational Complexity of Optimum Multiuser Detection. Algorithmica **4** (1998) 303-312
6. Hopfield, J.J.: Neurons with Graded Response Have Collective Computational Properties like Those of Two-stage Neurons. Proc. Nat. Acad. Sci. USA **81** (1984) 3088-3092
7. Jeney, G., Levendovezky, J., Imre, S., Pap, L.: Comparison of Different Neural Network Based Multiuser Detector. EUNICE 2000, Enschede (2000) 117-123

Blind Multiuser Detection Based on Kernel Approximation

Tao Yang and Bo Hu

Department of Electronics Engineering, Fudan University, Shanghai, China
taoyang@fudan.edu.cn

Abstract. A kernel based multiuser detection (MUD) scheme in code-division multiple-access (CDMA) system is proposed. In this scheme, the support vector (SV) under support vector (SVM) framework is obtained through a kernel sparsity approximation, which regulates the kernel width parameter via a heuristic approach to obtain an approximate equivalent SV. The corresponding SV coefficient is attained through evaluation of generalized eigenvalue problem, which avoids the conventional costly quadratic programming (QP) computation procedure in SVM. Simulation results show that the proposed scheme has almost the same BER as standard SVM and is better than minimum mean square error (MMSE) scheme when sample set is relatively large, meanwhile the proposed scheme have a low computation complexity.

1 Introduction

Code-division multiple-access (CDMA) technology constitutes an attractive multiuser scheme that allows users to transmit at the same carrier frequency in an uncoordinated manner. However, this creates multiuser interference (MUI) which can seriously degrade the quality of reception. Mutually orthogonal spreading codes for different users can provide an inherent immunity to MUI in the case of synchronous systems. Unfortunately, multipath distortions are often encountered in CDMA system and will reduce this inherent immunity to MUI. Multiuser detection (MUD) is such a technology can be used to overcome the channel fading and alleviate the MUI through the joint processing of the spreading code, timing, and signal strength as well as phase information. Conventional MUD schemes include optimal and sub-optimal algorithm, most of these algorithms are constructed under signal processing framework, which is computation intensive and is too expensive for the both power and dimension limited personal mobile communication application.

The learning algorithm based on kernel function implies potential application in pattern recognition, machine learning, image processing etc. and has been a research focus of intelligent learning in the past decade, the MUD algorithm based on kernel function has appeared in recent years [1][2][3], which show that under a certain condition the learning algorithm based on kernel function can improve the effectiveness of computation, thus improve the detection performance. In the proposed scheme, we adopt a heuristic not a training approach to find the kernel parameter, comparing with the conventional training based learning algorithm, the detection procedure is simplified and complexity is reduced.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 94 – 101, 2006.

The reminder of the paper is as follows, in section 1, we give the equivalent support vector (SV) finding process through sparsity kernel approximation algorithm, the SV coefficient evaluation is presented in section 2, the simulation and analysis is given in section 3 and finally we concluded this paper in section 4.

2 Analysis of Sparsity Kernel Approximation Algorithm

Conventional training based kernel algorithm proceed like this, the sample data in low dimension space is firstly mapped into high dimension space, then through an inner product calculation, the signature vector form sample data is selected, which is further projected into a low dimension space to provide classification output in an expression of linear combination of weight (such as training algorithm in SVM) [4]. Generally the map function is obtained through the kernel function, which give rise to some application specific problem, such as how to select kernel and how to simplify the training process? which is a fundamental but important issue. In SVM, SV and its corresponding coefficient can be obtained through quadratic programming (QP) procedure, which has the complexity of $O(k^2)$ (where k is the number of classes) and is often computation expensive. In this paper we attain the equivalent SV in the mapped space via a heuristic approach rather than a QP computation, and then the equivalent SV is approximated as input sample sequence, which reduces the search space effectively. The algorithm description is presented as follows.

Assume there are total l inputs sample $x_i (i = 1,...,l)$, which construct a new vector Ψ and is denoted as

$$\Psi = \sum_{i=1}^{l} \alpha_i \Phi(x_i) \tag{1}$$

where $\Psi \in F$, $x_i \in \mathbb{R}^N$, $\alpha_i \in \mathbb{R}$. In order to find an approximate expression of (1) in space F, let

$$\Psi' = \sum_{i=1}^{z} \beta_i \Phi(y_i) \tag{2}$$

where $z < l$, $\beta_i \in \mathbb{R}$, $y_i \in \mathbb{R}^N$. Then get the minimization of norm margin between Ψ and Ψ', we have [4]

$$\left\| \Psi - \Psi' \right\|^2 = \sum_{i,j=1}^{l} \alpha_i \alpha_j k(x_i, x_j) + \sum_{i,j=1}^{z} \beta_i \beta_j k(y_i, y_j) - 2\sum_{i=1}^{l}\sum_{j=1}^{z} \alpha_i \beta_j k(x_i, y_j) \tag{3}$$

generally inner-product calculation of F in (3) can be replaced by kernel function, let $z = 1$, then the extreme value evaluation can be represented as the minimal distance of the projection of Ψ and Ψ' on the $\Phi(y)$, namely

$$\min \left\| \frac{\Psi \cdot \Phi(y)}{\Phi(y) \cdot \Phi(y)} \Phi(y) - \Psi \right\|^2 = \|\Psi\|^2 - \frac{(\Psi \cdot \Phi(y))^2}{\Phi(y) \cdot \Phi(y)} \tag{4}$$

furthermore, if $k(y, y) = 1$, then the evaluation of extreme value of (4) is equivalent to $\max(\Psi \cdot \Phi(y))^2$, use Gaussian kernel function to evaluate the extreme of (4), we get an equivalent iterative expression[4]

$$y_{n+1} = \frac{\sum_{i=1}^{l} \alpha_i \exp\left(-\|x_i - y_n\|^2 / 2\sigma^2\right) x_i}{\sum_{i=1}^{l} \alpha_i \exp\left(-\|x_i - y_n\|^2 / 2\sigma^2\right)} \tag{5}$$

when the iterative result satisfy the stop criterion, an approximate projection vector y is obtained. Generally in the application of kernel function, two issues is crucial to the application performance, one is the relation between the optimal classification performance and number of y, the other is how to find the equivalent item in the input sample. The former is still an open problem, conventional SV evaluation assume a given kernel and parameter, through QP calculation the SV is obtained, but this is not optimal, from [6] we know that for the width parameter σ of a Gaussian kernel function, when it get larger, the total number of SV decrease in general, here we take an heuristic approach to decide the number of SV. Assume the initial σ is σ_0, the number of y is Z ,then we can get the generalization of (4), the value of Z vector can be expressed as

$$\min \sum_{k=1}^{Z} \left\| \frac{\Psi \cdot \Phi^k(y)}{\Phi^k(y) \cdot \Phi^k(y)} \Phi^k(y) - \Psi \right\|^2 = \sum_{k=1}^{Z} \left(\|\Psi\|^2 - \frac{\left(\Psi \cdot \Phi^k(y)\right)^2}{\Phi^k(y) \cdot \Phi^k(y)} \right) \tag{6}$$

Similarly we can get y_i, i=1,…,Z. So the classification output function is

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{z} \beta_i k(x, x_i) + b \right) \tag{7}$$

with (7), we can verify the test sample (β_i will be given in next part), if classification error satisfy $e \geq e_0$, where e_0 is the classification error threshold value, through the regulation of σ, the corresponding regulation of Z is obtained as

$$\sigma_i = \sigma_{i-1} + ec \tag{8}$$

where c is a constant. Z decrease in proportion to eZ, meanwhile if the number of y_i is m and satisfy $\|y_i\| \doteq 0$, then the number of Z is $eZ - m$ for the next iteration, when such an iteration process repeat for several times, the required e and sparsity will be satisfied.

3 Evaluation of Equivalent SV Coefficient

In standard SVM algorithm, the SV coefficient is provided through a QP procedure, the sparsity and SV coefficient is attained simultaneously. From [7] we know that the evaluation of SV coefficient can be obtained via various methods, but the result is the same, while sparsity will differs from the methods adopted. The receiving signal of CDMA terminal is just the mix of multiuser transmitting signal, due to nonlinearity and fading of channel, the receiving signal in matrix representation satisfy non-singularity, assume the signal source satisfy spatially irrelevant and second order non-stationary. Considering from the perspective of blind signal source separation, MUD can be formulated as the following process, source signal is firstly mixed linearly,

then the mixed signal is piped into a nonlinear processor (multipath and fading), and output is just the receiving signal, where through a nonlinear map of kernel, a linear classification processing is carried out in high-dimension space, the classification result is projected on the signature vector, which correspond to the SV, the signature vector is just the mix matrix W need to be evaluated, which is equivalent to the β_i in (7).assume the sample signal is x, then the estimated long-term and short-term covariance matrix in signature space is as [8]

$$\begin{aligned} \overline{C} &= \frac{1}{l}\sum\nolimits_i \overline{\Phi(x_i)}\,\overline{\Phi(x_i)}^T \\ \widetilde{C} &= \frac{1}{l}\sum\nolimits_i \widetilde{\Phi(x_i)}\,\widetilde{\Phi(x_i)}^T \end{aligned} \tag{9}$$

where $\overline{\Phi(x_i)}$ and $\widetilde{\Phi(x_i)}$ is denote as

$$\begin{aligned} \overline{\Phi(x_i)} &= \Phi(x_i) - \frac{1}{l}\sum\nolimits_{k=1}^{l} \Phi(x_k) \\ \widetilde{\Phi(x_i)} &= \Phi(x_i) - \frac{1}{2m+1}\sum\nolimits_{k=i-m}^{i+m} \Phi(x_k) \end{aligned} \tag{10}$$

with $\overline{C}$ and $\widetilde{C}$, W is estimated. The joint diagonalizaton of $\overline{C}$ and $\widetilde{C}$ can be obtained by solving the following generalized symmetric eigenproblem:

$$\overline{C}W = D\widetilde{C}W \tag{11}$$

where the columns of W are eigenvectors orthogonal in the metrics $\overline{C}$ and $\widetilde{C}$, i.e., $W^T\overline{C}W = diag\left(\overline{r}_1 ... \overline{r}_n\right)$, $W^T\widetilde{C}W = diag(\widetilde{r}_1 ... \widetilde{r}_n)$ and $D = diag\left(\lambda_1 = \overline{r}_1/\widetilde{r}_1 ... \lambda_n = \overline{r}_n/\widetilde{r}_n\right)$. here we denote by λ the largest eigenvalue and w the corresponding eigenvector. In this case, solving the the generalized eigenproblem $\overline{C}w = \lambda\widetilde{C}w$ is equivalent to maximizing the following measure

$$F = \frac{w^T\overline{C}w}{w^T\widetilde{C}w} \tag{12}$$

which is the ration between the long-term variance and the short-term variance of the source. Here the recover of source signal is attained by finding w that maximizes the degree of predictability defined by F. In order to get the extreme value of F, let

$$w = \sum\nolimits_{i=1}^{l} \alpha_i \Phi(x_i) \tag{13}$$

from which the numerator of F can be expressed as

$$w^T\overline{C}w = \frac{1}{l}\sum\nolimits_{i,j} \alpha_i\alpha_j \Phi(x_i)^T \sum\nolimits_k \overline{\Phi(x_k)}\,\overline{\Phi(x_k)}^T \Phi(x_j) = \frac{1}{l}\alpha^T\overline{K}\,\overline{K}^T\alpha \tag{14}$$

in a similar way we get the denominator as

$$w^T\widetilde{C}w = \frac{1}{l}\alpha^T\widetilde{K}\,\widetilde{K}^T\alpha \tag{15}$$

therefore F is expressed as

$$F = \frac{\alpha^T \overline{K}\overline{K}^T \alpha}{\alpha^T \widetilde{K}\widetilde{K}^T \alpha} \tag{16}$$

from (16) we know the generalized eigenproblem can be formulated as

$$\overline{K}\overline{K}^T \alpha_i = \lambda_i \widetilde{K}\widetilde{K}^T \alpha_i \tag{17}$$

in which λ_i and α_i correspond the largest eigenvalue and eigenvector, respectively. In the receiving end, the test sample x need to be projected nonlinearly on w, and thus to get the classification output, that is, the original sample space is $y = \sum_{i=1}^{l} w_i^T \Phi(x_i)$, in kernel representation, we have

$$y = \sum_{i=1}^{l} \beta_i k(x_i, x) \tag{18}$$

where $K_{ij} = \Phi(x_i)^T \Phi(x_j)$ is $Z \times L$ matrix, $\overline{K_{ij}} = K_{ij} - (1/l)\sum_{k=1}^{l} K_{ik}$, thus $\overline{K}\overline{K}^T$ and $\widetilde{K}\widetilde{K}^T$ is $Z \times Z$ matrix, so the number of β_i satisfy the requirement for SV.

4 Simulations and Performance Analysis

Based on the theoretic analysis of kernel function above, in this part we will verify the algorithm through a Matlab simulation, an illustration of the simulation structure description is presented in Figure1. Moreover, a MUD performance comparison between three schemes, SVM, minimum mean square error (MMSE) and sparsity kernel approximation scheme is given, here MMSE scheme is a linear MUD scheme and serve as a performance comparison criterion, SVM scheme is based on sampletraining and is closely relevant to the sparsity kernel approximate scheme proposed in this paper. The simulation environment is configured as follows: the synchronized CDMA system with user number $N = 5$, number of multipath $p = 3$, and multipath time delay is less than a symbol period, Gold sequence is selected as the spreading code, code length $L = 31$, the multi-access interference(MAI) is distributed randomly in [0,0.4], transmission interference is Gaussian white noise. The training sample and test sample is comprised of two parts: I: (30, 100) and II: (100, 300). The channel

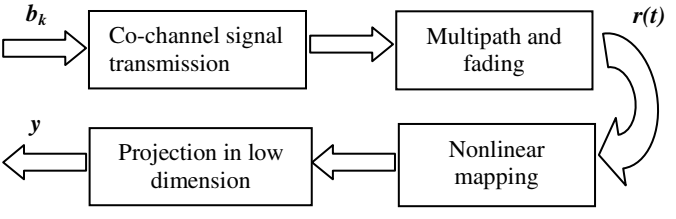
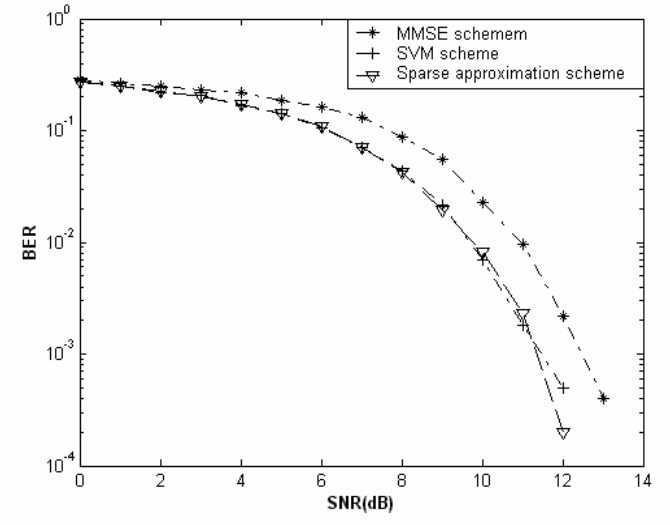
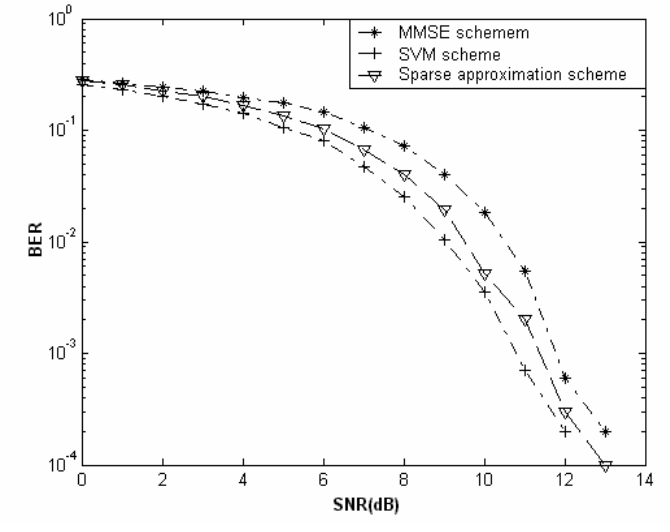


Fig. 1. Illustration of MUD based on kernel function



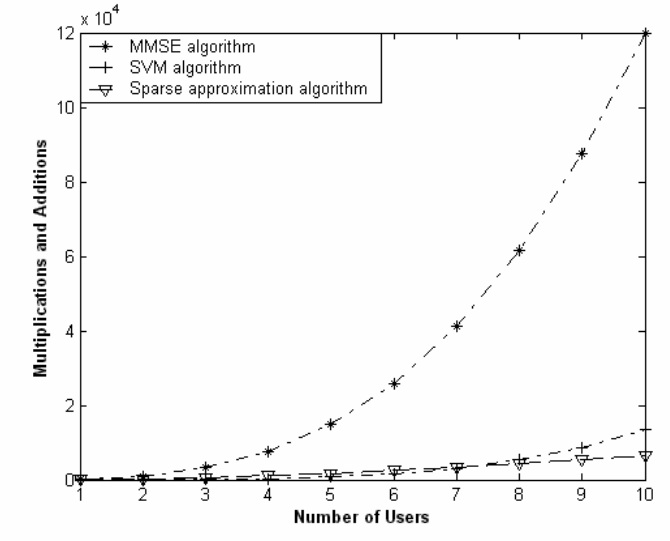
(a) Sample set I



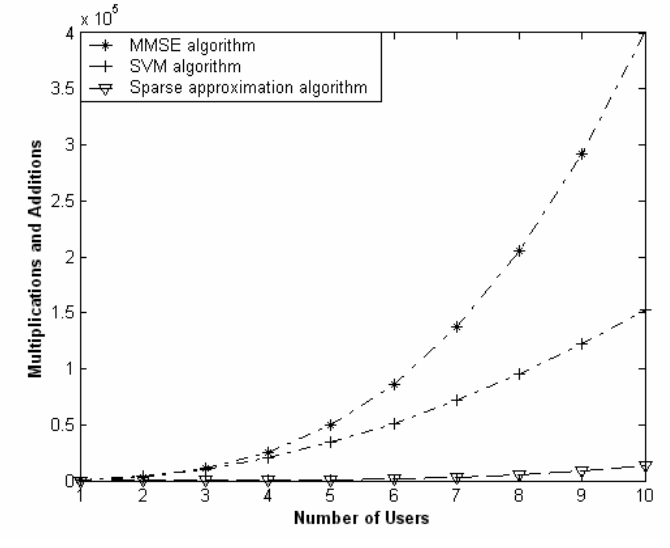
(b) Sample set II

Fig. 2. BER performance comparison for different sample set

impulse response (CIR) is $H(z) = 0.5 + 0.6z^{-1} + 0.3z^{-2}$, the initial width value for Gaussian kernel function is $\sigma_0 = 0.15$, regulation factor $c = 0.8$, classification error threshold is $e_0 = 0.05$. Based on above configuration, the computation complexity and 100,000 Monte-Carlo bit error rate (BER) performance simulation is given in Figure 2



(a) Sample set I



(b) Sample set II

Fig. 3. Complexity comparison for different samples set

and Figure3, respectively. The theoretic complexity for SVM and MMSE is $O(N^2)$ and $O(8N^3)$. The complexity of eigenvalue problem for sparse kernel approximate algorithm is $O(S^3)$, where S is the number of SV, because the complexity of iteration evaluation of SV increase linearly, so the total complexity is still $O(S^3)$.

From simulation result we know that among the three schemes the BER performance get closely each other between sparsity kernel approximate and SVM scheme under a small sample set I, and is a little better than MMSE, the complexity is in increase order as $C_{sparse} < C_{SVM} \ll C_{MMSE}$. For large sample set II, BER performance of SVM is slightly better than sparsity kernel scheme, but both are better than MMSE scheme, the complexity order is $C_{sparse} << C_{SVM} < C_{MMSE}$. Comparing with SVM, sparsity kernel approximate scheme is enhanced with training adaptation ability, under the given threshold e_0, via regulation of kernel parameter and sparsity evaluation, an approximate optimal equivalent SV is obtained, especially under a large sample set case, such a heuristic approach will maintain a given BER performance, meanwhile, the computation complexity increase a little.

5 Conclusions

A kernel function based multiuser detection under the support vector machine framework is proposed in this paper. The scheme can be summarized as a process of attaining the equivalent support vector as well as its coefficient, where the heuristic approach play an important role in acquisition of equivalent support vector and improve the computation efficiency. Comparing with conventional signal process based MUD, complicated matrix decomposition and inversion is avoided. The simulation results show that the proposed scheme can keep an appropriate compromise between BER performance and complexity especially under a large sample set. So for heavy multiuser interference environment, this scheme is of great potential for application.

References

1. Rahman, S. A., Saito, M., Okada, M. et al.: An MC-CDMA Signal Equalization and Detection Scheme Based on Support Vector Machines. In: Proceedings of 1st International Symposium on Wireless Communication Systems (2004) 11-15
2. Zhou, W., Zhang, L., Jiao, L..: Adaptive Support Vector Machine Multiuser Detection. Chinese Journal of Electronics 31(1) (2003) 92-97
3. Chen, S., Samingan, A. K., Hanzo, L.: Support Vector Machine Multiuser Receiver for DS-CDMA Signals in Multipath Channels. IEEE Trans. on Neural Networks 12(3) (2001) 604-611
4. Vapnik, V.: The Nature of Statistical Learning Theory. Springer-Verlag, New York (1995)
5. Scholkopf, B., Knirsch, P., Smola, A. et al.: Fast Approximation of Support Vector Kernel Expansions and An Interpretation of Clustering as Approximation in Feature Spaces. In: Proceedings of DAGM-Symposium (1998) 125-132
6. David, M. J., Robert, T., Duin, P. W.: Data Domain Description Using Support Vectors. In: Proceedings of ESANN, Belgium (1999) 251-256
7. Scholkopf, B., Mika, S., Burges, C. J. C.: Input Space versus Feature Space in Kernel-based Methods. IEEE Trans. on Neural Networks 10(5) (1999) 1000 - 1017
8. Martinez, D., Bray, A.: Nonlinear Blind Source Separation Using Kernel. IEEE Trans. on Neural Networks 14(1) (2003) 228 - 235

A Novel Blind Multiuser Detection Model over Flat Fast Fading Channels

Hongbo Tian, Qinye Yin, and Ke Deng

Institute of Information Engineering, School of Electronics and Information Engineering, Xi'an Jiaotong University, 710049, Xi'an, China
jhk@mailst.xjtu.edu.cn

Abstract. A novel auxiliary system (AS) in particle filtering detector (PFD) for blind multiuser detection in synchronous system over flat and fast Rayleigh fading channels is proposed. We adopt an autoregressive-moving-average (ARMA) process to model the temporal correlation of the channels. Based on the ARMA process, the Hopfield neural network is adopted as an auxiliary system, the auxiliary system allows the particle filtering select fitted size of trajectories, and we further propose to obtain soft multiuser detection from the particle filtering and the auxiliary system. Simulation results demonstrate the performance of the proposed model.

1 Introduction

Since multiuser detection (MUD) was introduced, it has received a great deal of attention, numerous detectors have been proposed in the literature including the decorrelating detector, and the minimum mean square error (MMSE) detector [1]. In practice, while performing MUD, the distortion in signal strength which is due to the time varying fading channels must be estimated. Recently, for this purpose various blind schemes have been proposed. Among various blind algorithms, those based on particle filtering [2][3] are very particular for it can be computed in parallel.

Among the proposed particle filtering algorithm for the wireless communications, [2] applied particle filtering (PF) to a time-observation state space model, it not only achieve efficient performance but also circumvent the difficulty of complexity which is exponential with the users. Nevertheless, the complexity of PF is also increase with the number of the trajectories, and the fitted number of trajectories is not certain. In this paper, a novel auxiliary system (AS) in particle filtering detector (PFD) is developed to circumvent the difficulty of selecting the fitted size of trajectories, it can also help the PF to obtain soft estimations by utilizing the AS.

2 Signal Model

Consider a synchronous CDMA system with a processing gain C and K users. The transmitted symbols are belong to a finite alphabet set $B= \{b_1,\ldots,b_Q\}$. At the n-th symbol interval, the set of matched filter outputs at the receiver can be represented in vector-matrix form according to

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 102 – 107, 2006.

$$y_n = RA_n b_n + u_n \tag{1}$$

where y_n=[$y_{n,1}$, …, $y_{n,K}$]$^{\mathrm{T}}$ with $(\cdot)^{\mathrm{T}}$ denoting matrix transposition, R is the crosscorrelation matrix whose element r_{k_1,k_2} represents the crosscorrelation between the signature waveforms of the k_1th and the k_2th user, A_n=diag($a_{n,1}$, …,$a_{n,K}$) is the diagonal matrix of the channel state information, b_n=[$b_{n,1}$, …, $b_{n,K}$]$^{\mathrm{T}}$ is the antiphonally modulated user data vector, and u_n is the complex Gaussian noise vector with the with covariance matrix equal to $\sigma^2\mathbf{I}$. The channel for each user is considered as Rayleigh flat fading channel and ARMA processes can be adopted to model its time correlation [4]. Thus, $a_{n,k}$ can be represented by an ARMA($r1$, $r2$) model as

$$a_{n,k} + \phi_{k,1}a_{n-1,k} + \cdots + \phi_{k,r_1}a_{n-r_1,k} = \rho_{k,0}v_{n,k} + \cdots + \rho_{k,r_2}v_{n-r_2,k} \tag{2}$$

where $v_{n,k}$ is an i.i.d. random complex Gaussian process, and $\{\phi_{k,1},\cdots\phi_{k,r_1}\}$ and $\{\rho_{k,0},\cdots,\rho_{k,r_2}\}$ are the AR and MA coefficients of the model.

3 Particle Filtering Detector

The time-observation state space model (TOSSM) representation of CDMA systems in flat fading channels can be found in existing literature [2]. Based on this, in a Bayesian framework, the optimum decision on b_n can be obtained by the marginalized posterior mode criterion, which is expressed as

$$\hat{b}_{n,k} = \mathrm{sgn}\left(\sum_{b_n\in\{B\}^K} b_{n,k}\, p\left(b_n \middle| \overline{y}_{1:nK}\right)\right) \tag{3}$$

where $p\left(b_n \middle| \overline{y}_{1:nK}\right)$ is the posterior distribution which is essential for computing (3) and the subscript 1:nK denotes a collection of the variable indexed from 1 to nK, e.g., $\overline{y}_{1:nK} = \{\overline{y}_1,\cdots,\overline{y}_{nK}\}$ and $\overline{y}_n$ is the whitened matched filter(WMF). In the context of the proposed problem, when $\overline{y}_n$ is observed at time interval n, the objective of particle filtering is to draw, say, J weighted random samples $\left\{b_{1:n}^{(j)}, \omega_{nK}^{(j)}\right\}$ from $p\left(b_{1:n} \middle| \overline{y}_{1:nK}\right)$, where $\omega_{nK}^{(j)}$ is the weight of the j-th sample $b_{1:n}^{(j)}$. With the samples, $p\left(b_n \middle| \overline{y}_{1:nK}\right)$ can be approximated by

$$P\left(b_{1:n} \middle| \overline{y}_{1:nK}\right) \overset{\Delta}{=} \sum_{j=1}^{J} \omega_{nK}^{(j)} \prod_{l=1}^{nK} \delta\left(b_{i,k} - b_{i,k}^{(j)}\right) \tag{4}$$

and hence the MPM solution of $\boldsymbol{b}_n$ by a simple weighted summation as

$$\hat{b}_{n,k} = \mathrm{sgn}\left(\sum_{j=1}^{J} b_{n,k}^{(j)} \omega_{nK}^{(j)}\right) \tag{5}$$

and the weights can be calculated as

$$\omega_{nK}^{(j)} \propto \frac{p\left(b_{1:n}^{(j)} \middle| \overline{y}_{1:nK}\right)}{\pi\left(b_{1:n}^{(j)} \middle| \overline{y}_{1:nK}\right)} \propto \mu_{nK-1}^{(j)} \omega_{nK-1}^{(j)} \tag{6}$$

$p\left(\overline{y}_{nK} \middle| b_{1:n}^{(j)}, \overline{y}_{1:nK-1}\right)$ is the likelihood function after marginalizing out h_l, and can be obtained from the predictive procedure of the Kalman filter[2] .

4 Hopfield Neural Network

In a general CDMA system, the optimal multiuser detection (OMD) produces an estimate for the information vector transmitted at the discrete-time instant n, In the synchronous case it holds that

$$b_{n,OMD} = \arg \max_{b \in \{B\}^K} \left\{ 2 r_n^T b - b^T S_n b \right\} \tag{7}$$

where $r_n = \left[r_{n,1}, \cdots, r_{n,K} \right]^T$, $r_{n,k}$ is the output of the matched filter-bank corresponding to the n-th data packet. $S_n \in R^{K \times K}$ is the (symmetric) matrix of signal cross correlations at n-th symbol interval. Because of the exponential growth with the computational complexity of the OMD with the number of active users, suboptimal detection schemes have been proposed. One of them is the multiuser suboptimal detection (MSD). The MSD consists of a collection of stages m = 1, 2, …, each producing an estimate $b_{n,MSD}(m+1)$ as follows

$$b_{n,MSD}(m+1) = sign\left[r_n - \left(S_n - E_n \right) b_{n,MSD}(m) \right] \tag{8}$$

where $\boldsymbol{E_n}$ is a diagonal matrix with elements $e_{n,i,i} = s_{n,i,i}$ i=1,…,K.

In HNN, Each neuron i receives a weighted sum of the activations of other neurons in the network, and updates its activation according to the rule:

$$V_i = g\left(U_i\right) = g\left(\sum_{j \neq i} T_{ij} V_j + I_i \right) \tag{9}$$

It has been shown [5] that, in the case of symmetric connections (T_{ij}= T_{ji}), It is apparent from (7) that the OMD objective function is very similar to a HNN energy function. Moreover, (7) can be rewritten as

$$b_{n,OMD} = \arg \min_{b \in \{+1,-1\}^K} \left\{ -r_n^T b + \frac{1}{2} b^T \left(S_n - E_n \right) b \right\} \tag{10}$$

since $b^T E_n b$ is always a positive number. The matrix -(S_n-E_n) is symmetric, and has zero diagonal elements since s_{ii}=e_{ii}. Therefore, the OMD objective function can be

directly translated into the energy function of an HNN with weight matrix $T = -(S - E)$ and biases $I = r_n$. So, (9) can be rewritten in matrix form as

$$V_n(m+1) = g\left[r_n - (S_n - E_n)V_n(m) \right]. \tag{11}$$

5 Auxiliary System Model in Particle Filtering Detector

Even though the PFD has potential to provide near-optimum performance, the size of trajectories isn't discussed in [2][3] and a fixed number is adopted directly. Normally, the setting should ensure the performance of the PFD, thus the size should large enough. The implementation is, in fact, not all iterations need same size of trajectories. Comparing with $b_{k,n}$ which need m trajectories to compute the exact posterior distribution, $b_{i,j}$ may need d trajectories and d may differ from m. When the upper bound trajectories are adopted to assure the performance of PF, quite a number of trajectories are unnecessary. Therefore, an AS in PFD (AS-PFD) is proposed, which are affordable to reduce the size of trajectories while maintain the performance.

The function of AS is to justify whether the size of the trajectories are adequacy, and therefore the AS demands lower complexity and high performance. According to [6], the NP-complete problem of minimizing the objective function of the OMD can be translated into minimizing an HNN "energy" function, thus affordable to take advantage of the ability of HNN's to perform very fast gradient descent algorithms. The channel state information which has obtained during the procedure of the TOSSM can be used to form the HNN's energy. In this paper, HNN is adopted as the AS. The algorithm can be outlined as follows:

```
Initialization: set J: the size of trajectories
At the m-th iteration (m=n×Q+k):
•Trajectory expansion
  For j=1 to J
      Calculate the weight ω_nk^(j) according to (6).
      If j=J_min & k=1
            calculate the output V_n according to (11).
            end
      If j>J_min  evaluate (b̂_{n,k})_MPM according to (5).
            if (b̂_{n,k})_MPM = V_{n,k}  break;  end
       end
  end
```

If j<J, we get $\hat{b}_{n,k} = \left(\hat{b}_{n,k}\right)_{MPM}$, if j=J, which means $\left(\hat{b}_{n,k}\right)_{MPM} \neq V_{n,k}$, the final results should consider the PFD and HNN. Therefore, we propose a soft decision to obtain the final results, and the criterion is as follows:

$$\hat{b}_{n,k} = \begin{cases} \left(\hat{b}_{n,k}\right)_{MPM} & \left|p(b_{n,k}=+1) - p(b_{n,k}=-1)\right| > threshold \\ v_{n,k} & \left|p(b_{n,k}=+1) - p(b_{n,k}=-1)\right| \le threshold \end{cases}. \quad (12)$$

6 Simulation Result

In this section, we present several simulation results that show the performance of the proposed PFD. In all the experiments, the differential binary phase shift keying (DBPSK) is exploited to overcome the phase ambiguous. A second AR process was adopted for the fading process, which was normalized to have a unit power, and thus the signal-to-noise ratio (SNR) was obtained by $10\log\left(1/\sigma^2\right)$. Both code rates are 1/2, and both block size of code bits are 512 while that of information bits are 256.

In Fig. 1, we provide the BER vs. SNR for the different algorithms on a scenario of Ω_d=0.03. The maximum size of the trajectories is the J (here, set J=151), and the mean one is the mean value of the trajectories adopted by the AS-PFD. It can be observed that the AS-PFD can provide a performance of about 2~3dB at average over the PFD using maximum size of trajectories and about 3~6dB at average over the PFD with mean size of trajectories. In Fig. 2, similar observations can be drawn as for the previous case on a scenario of Ω_d=0.03 even though the overall performance of the detectors is worse, which is reasonable considering that the channel are fading fast. The number of the mean size of the trajectories is about 70 percents of the total trajectories on a scenario of Ω_d=0.03 and 75 percents of the totals on Ω_d=0.05.

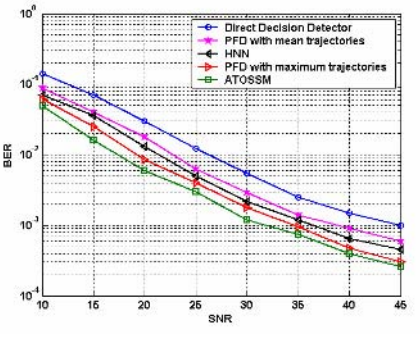


Fig. 1. BERs vs. SNR. Ω_d=0.03.

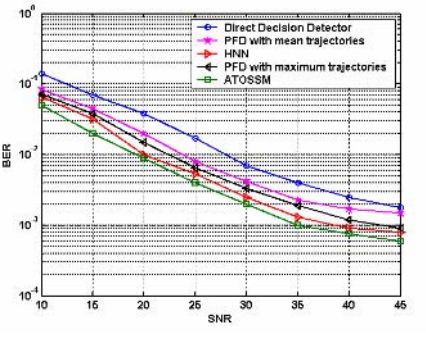


Fig. 2. BERs vs. SNR. Ω_d=0.05.

7 Conclusion

In this paper, we proposed to solve blind MUD over flat fast fading channels. We constructed a novel auxiliary system in particle filtering detector, Simulation results demonstrate that the performance of AS-PFD is superior to that of the PFD and the HNN, meanwhile the complexity of the AS-PFD is smaller than the PFD.

Acknowledgements

This work was supported by the project of National Natural Science Foundation of China (No. 60502022, No. 60572046), Doctor Program Foundation of University of China (No. 20030698027).

References

1. Verd'u, S.: Multiuser Detection. New York: Cambridge University Press (1998)
2. Huang, Y., Zhang, J., and Djuri´c, P. M.: Adaptive Blind Multiuser Detection over Flat Fast Fading Channels using Particle Filtering. IEEE Trans.on Communications 4 (2004) 2419-2423
3. Punskaya, E., Andrieu, C., Doucet, A., Fitzgerald, W.J.: Particle Filtering for Multiuser Detection in Fading CDMA Channels. IEEE Signal Processing Workshop on Statistical Signal Processing (2001) 38-41
4. Huang, Y., Djuri´c, P. M.: A Blind particle Filtering Detector of Signals Transmitted over Flat Fading Channels. IEEE Trans.on Signal Processing 52 (2004) 1891-1900
5. Hopfield, J. J.: Neurons with Graded Response Have Collective Computational Properties like Those of Two-State Neurons. In: Proc. Nut. Acad. Sci. USA, 81 (1984) 3088-3092
6. Kechriotis, G., Manolakos, E.: Hopfield Neural Network Implementation of the Optimal CDMA Multiuser Detector. IEEE Trans.on Neural Networks 7 (1996) 131-141

Robust Multiuser Detection Method Based on Neural-net Preprocessing in Impulsive Noise Environment

Ying Guo and Tianshuang Qiu

School of Electronic and Information Engineering,
Dalian University of Technology,
Dalian, Liaoning 116024, China
lovelygy2002@yahoo.com.cn, qiutsh@dlut.edu.cn

Abstract. This paper models the ambient noise as α stable distribution and proposes a novel robust multiuser detection (MUD) method that involves an adaptive nonlinear preprocessor based on multilayer perceptron neural-network whose action is to suppress the negative effect of impulsive noises on the followed decorrelating decision-feedback (DDF) multiuser detector. Simulation results indicate the proposed new method is robust and offers performance enhancement over traditional technology in impulsive noises.

1 Introduction

A typical characteristic of code division multiple access (CDMA) system permits many users sharing the same bandwidth by using non-orthogonal spreading waveforms. But that will cause the multiple-access interference (MAI) unavoidably. Many multiuser detection (MUD) technologies [1] have been developed to suppress the MAI. Most of them focus on the unique and outstanding MAI instead of the channel noises, so Gaussian noise assumption is popular for its mathematical tractability. While recently extensive experimental evidences show in many physical channels, such as urban and indoor radio channels, the ambient noise is non-Gaussian due to the impulsive nature of the man-made electromagnetic interference and a great deal of natural noise as well, which is modeled as α stable distribution[2,3]. What is attracted is the α stable model is not limited to particular situation unlike the commonly used Gaussian mixture model [4]. However the conventional receivers designed under Gaussian assumption always incurs significant performance degradation in the presence of impulsive noises. This raises the issues of devising robust detector under non-Gaussian noise.

The decorrelating decision-feedback (DDF) multiuser detector [1] is a multistage detector which could achieve error-free demodulation. First it sorts the users according to their powers, with the strongest user followed by the weaker users. Then it employs Cholesky decomposition to determine decorrelating and feedback matrices for successive interference cancellation. Such that the strongest one is detected first and could be obtain the same performance as the decorrelating detector, while the weakest one gets single-user performance if all feedback decision are correct. In this

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 108 – 113, 2006.

paper, we investigate the performance degradation of conventional DDF multiuser detector under α stable distributed noise and propose a neural-net based method to improve the robustness of traditional DDF multiuser detector. The reason of choosing neural-net as the preprocessor is that it is such a simple and calculative technique that could realize the complex nonlinear mapping. Comparing to other nonlinear function such as clipping, puncher function and etc. which should be set clipping parameter in advanced, the most important advantage of our proposed method is it could form the clipping nonlinearity automatically to reduce the influence of impulsive noise on the followed DDF multiuser detector by self-learning and self-adjusting.

2 System and Noise Model

In this paper a coherent synchronous CDMA system employing binary phase-shift keying (BPSK) is considered. At the receiver, the received signal is filtered by a chip-matched filter and sampled at the chip rate. The resulting discrete-time signal corresponding to the i th symbol is given by

$$\mathbf{r}(i) = \sum_{k=1}^{K} A_k b_k(i)\mathbf{s}_k + \mathbf{n}(i) \qquad i = 1 \cdots M \tag{1}$$

where A_k is known received amplitude and $A_1 \geq A_2 \geq \cdots \geq A_K$; M , $\{b_k(i)\}_{i=0}^{M-1}$ and $\mathbf{s}_k$ denote, respectively, the number of data symbols per user in the data frame of interest, the transmitted symbol stream and the normalized signaling waveform of the k th user, $\mathbf{n}(i)$ is the ambient noise.

The α stable distribution is introduced as the impulsive noise model in wireless channel. It is an excellent statistic non-Gaussian noise model which provides a flexible and broad range of impulsiveness to model different impulsive noise processes [2]. Gaussian distribution is only a special member of it. And it satisfies the generalized central limit theorem and stability property. Its excellent difference from Gaussian distribution is its thick tail in distribution and spikes in its waveform which result in many algorithms under Gaussian assumption failed. It can be conveniently described by its characteristic function as $\varphi(t) = e^{\left\{ jat - \gamma |t|^{\alpha}[1 + j\beta \operatorname{sgn}(t)\omega(t,\alpha)] \right\}}$ .Where $\omega(t,\alpha) = \tan\frac{\alpha\pi}{2}$ ($\alpha \neq 1$), $\omega(t,\alpha) = \frac{2}{\pi}\log|t|$ ($\alpha = 1$), and $\operatorname{sgn}()$ is a sign function. α is the characteristic exponent restricted in $0 < \alpha \leq 2$, it controls the thickness of the tail in the distribution. The dispersion parameter γ ($\gamma > 0$) is similar to the variance of Gaussian process and β ($-1 \leq \beta \leq 1$) is the symmetry parameter. If $\beta = 0$, the distribution is symmetric and the observation is referred to as the $S\alpha S$ (symmetry α stable) distribution. a ($-\infty < a < \infty$) is the location parameter. When $\alpha = 2$ and $\beta = 0$, the α stable distribution becomes the Gaussian distribution. Throughout this paper $S\alpha S$ process is chosen.

3 Proposed Multiuser Detection Method Based on Neural-net Preprocessing

Depending on the received signal $\mathbf{r}(i)$, our objective is to find the bit estimate $\mathbf{b}(i)$. Since $\mathbf{r}(i)$ contains a great deal of impulsive noise it could result in increasing bit error rate dramatically. While it's well known a nonlinear process could eliminate the large peaks of signal, so we consider adding a preprocessor in front of the multiuser detector in order to achieve robust performance. There are many kinds of nonlinear function such as clipper function, puncher function and etc. They are all on the basis of explicit cutoff parameter that is hard to be specified in advance. In order to overcome this problem we investigate a nonlinear preprocessor based on a MLP neural-network which could achieve nonlinearity and its corresponding cutoff parameter during training period. The scheme of the proposed architecture is shown in Fig.1.

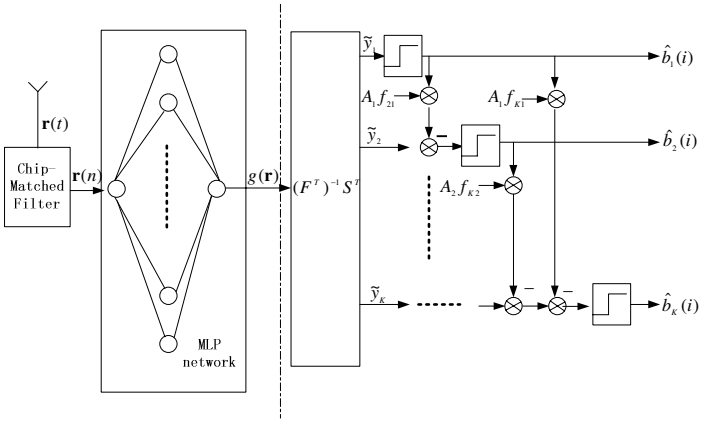


Fig. 1. Scheme of the proposed DDF multiuser detection method based on neural-net preprocessing

The neural-net consists of one input and one linear output node, 15 sigmoid nodes in the hidden layer. And then it forms nonlinearity to prevent the impulsive noise from entering the subsequent mulituser detector via back propagation algorithm. The input signal is noisy signal denoted by formula (1) and the target signal is the noise free signal:

$$\mathbf{r}(i) = \sum_{k=1}^{K} A_k b_k(i)\mathbf{s}_k \ . \tag{2}$$

After conforming the nonlinearity all users share the same function. What should be mentioned is that (1) and (2) are discrete time and chip-based so that all network weights are updated at chip-rate through back propagation training. It is seen that the spreading codes, training sequence and signal amplitudes of all users are required. Therefore this device could be implemented at the base station receiver [5].

After the nonlinear filter based on MLP neural-net, the signals passed the matched filter can be expressed as [6]

$$y(i) = \mathbf{S}^T \mathrm{g}(\mathbf{r}(i)) \tag{3}$$

where $\mathrm{g}(\mathbf{r}(i)) = [\mathrm{g}(r_1(i)), \cdots, \mathrm{g}(r_N(i))]$. And then define $\mathbf{R} = \mathbf{S}^T\mathbf{S}$ as the cross correlation matrix of the signature waveforms. Applying Cholesky decomposition $\mathbf{R}$ can be factored as

$$\mathbf{R} = \mathbf{F}^T\mathbf{F} \tag{4}$$

Where $\mathbf{F}$ is a lower triangular matrix (i.e., $F_{kl} = 0$ if $k < l$). If $\mathbf{R}$ is nonsingular, so is $\mathbf{F}$, and we can process the vector output by matched filter through the upper triangular matrix and yielding the whitened matched filter outputs:

$$\tilde{\mathbf{y}}(i) = (\mathbf{F}^T)^{-1}\mathbf{S}^T \mathrm{g}(\mathbf{r}(i)). \tag{5}$$

It is shown that $\tilde{y}_1(i)$ contains no interference from other interfering users. Its decision is therefore made first by passing through zeros-thresholding device. Similarly, the k th component of $\tilde{\mathbf{y}}(i)$ contains no trace of users $k+1, \cdots, K$. Since decisions for the stronger users $1, \cdots k-1$ have already been made, they can be used to form a feedback term. Following the philosophy of successive interference cancellation, the sequential demodulation process can be expressed by

$$\hat{\mathbf{b}}(i) = \mathrm{sign}((\mathbf{F}^T)^{-1}\mathbf{S}^T g(\mathbf{r}(i)) - (F - \mathrm{diag}(\mathbf{F}))\mathbf{A}\hat{\mathbf{b}}(i)). \tag{6}$$

4 Simulation Results

We consider a synchronous CDMA network with five users under perfect power controlled, each user is assigned a Gold code of length $N = 31$. Bit error rates (BER) of the first and last user are investigated. Firstly, we introduce the Generalized Signal-to-Noise Ratio (GSNR) [2] defined as

$$\mathrm{GSNR} = 10\log\frac{|A_K|^2}{\gamma}. \tag{7}$$

Here γ ($\gamma > 0$) denotes the dispersion parameter of $S\alpha S$ noise process and in this paper, the GSNR is referred to the last user.

Experiment 1: Under Gaussian noise conditions. In Fig.2 (a), it is supposed all users have equal power. It is seen that when the noise is Gaussian the proposed DDF method based on neural-net preprocessing have almost identical performance with that of the traditional DDF detector. And there is similar phenomenon in Fig.2 (b), when the near-far effect is taken into account the power of each interfering user is the

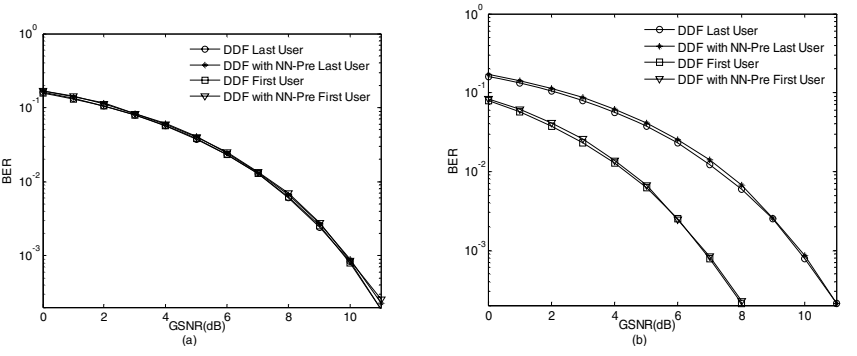


Fig. 2. Bit error rate against GSNR in five-user channel under Gaussian noise assumption (a) all user have equal powers (b) the power of each interference is 6dB above the last user

same as 6dB above the last user. That is to say, the proposed DDF method based on neural-net preprocessing is not sensitive to the presence of MAI too.

Experiment 2: Under highly impulsive with $\alpha = 1.5$ noise condition. First the BER performance of each user equal power controlled is plotted in Fig.3 (a). As expected, the performance of the traditional DDF method dropped drastically resulting from the influence of impulsive noise contamination. On the other hand, due to the nonlinearity of the preprocessor based on neural-net, the influence of impulsive noises is greatly reduced and the performance is improved significantly. Fig.3 (b) shows the performance of the two DDF methods when both MAI and impulsive noise present together. MAI condition is the same as that in experiment 1.

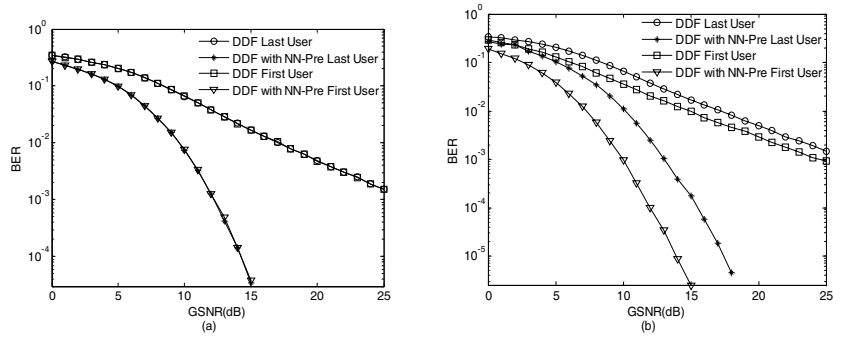


Fig. 3. Bit error rate against GSNR in five-user channel under impulsive noise assumption ($\alpha = 1.5$) (a) all users have equal powers (b) the power of each interference is 6dB above the last user

Therefore from these experiments it could be said the proposed DDF method based on neural-net preprocessing is robust, it is not only proper to Gaussian noise condition but also to impulsive noise condition.

5 Conclusion

In this paper, we model the impulsive communication channel noise as α stable distribution and propose a novel receiver structure which involves a nonlinear preprocessor based on neural-net for impulsive noise followed a DDF multiuser detector for MAI. The simulation results show that the novel method has superior capability of jointly suppressing the MAI and α stable distributed ambient noises, because the neural-net preprocessor could form a nonlinearity function with approximate cutoff well and then limits the impulsiveness of α stable distributed noise. So the proposed DDF method based on neural-net preprocessing is a better method compared to conventional DDF multiuser detection method because it enhances the robustness no matter for Gaussian noise or impulsive noise.

Acknowledgements

This work is supported by National Science Foundation of China under Grants 60372081, 60172072, 30170259, 30570475, and the Science Foundation of Liaoning Province under Grant 2001101057.

References

1. Verdu, S.: Multiuser Detection. Cambridge, Cambridge University Press (1998)
2. Nikias, C. L., Shao, M.: Signal Processing with Alpha-stable Distribution and Application. New York, John Wiley & Sons Inc (1995)
3. Middleton, D.: Non-Gaussian Models in Signal Processing for Telecommunications: New Methods and Results for Class A and Class B Noise Models. IEEE Trans. on Information Theory, 45(4) (1999) 1129–1149
4. Aazhang, B., Poor, H.: Performance of DS/SSMA Communications in Impulsive Channels-Part Ⅰ: Linear Correlation Receivers. IEEE Trans. on Communications, 35(11) (1987) 1179-1188
5. Chuah, T. C., Sharif, B.S., Hinton, O. R.: Robust CDMA Multiuser Detection Using a Neural-Network Approach. IEEE Trans. on Neural Networks, 12(6) (2001) 1546-1558
6. Chuah, T. C., Sharif, B.S., Hinton, O.R.: Robust Decorrelating Decision-Feed Back Multiuser in Non-Gaussian Channels. IEEE Trans. on Signal Processing, 81(8) (2001) 1997-2004

Channel Equalization Using Complex Extreme Learning Machine with RBF Kernels

Ming-Bin Li, Guang-Bin Huang, Paramasivan Saratchandran, and Narasimhan Sundararajan

School of Electrical and Electronic Engineering,
Nanyang Technological University, Singapore
egbhuang@ntu.edu.sg
http://www.ntu.edu.sg/home/egbhuang

Abstract. This paper studies the performance of extreme learning machine with complex-valued radial basis function (ELM-CRBF) in the channel equalization applications. Comparing with complex minimal resource allocation network (CMRAN), complex radial basis function (CRBF) network and Bayesian equalizers, the simulation results show that ELM-CRBF equalizer is superior in terms of symbol error rate (SER) and learning speed.

1 Introduction

It is well known that equalizers are popularly alternative at receivers to recover the original symbols from the received signals in digital communication systems. In recent years, several neural network models such as feedfordward neural networks, radial basis function (RBF) networks and recurrent neural networks (RNN) have been successfully used for solving equalization problems as neural networks are well suited for nonlinear classification problems [1]. Complex-valued neural networks have attracted considerable attention in channel equalization applications in the past decade. Cha and Kassam [2] have proposed a complex-valued radial basis function(CRBF) network which adopts the stochastic gradient learning algorithm to adjust parameters. Comparing with previously existing equalizers, the CRBF equalizer is superior in terms of symbol error rate (SER). Jianping et al [3] have developed a complex-valued minimal resource allocation network (CMRAN) equalizer. Applying the growing and pruning criterion, the CMRAN equalizer can realize a more compact structure and can obtain better performance than CRBF and some other equalizers.

Recently, a new learning algorithm for single-hidden-layer feedforward neural networks (SLFNs) named extreme learning machine (ELM) is proposed by Huang, et al[4, 5, 6, 7, 8, 9]. Unlike the traditional approaches (such as BP algorithms) which may face difficulties in manually tuning control parameters (learning rate, learning epochs, etc) and/or local minima, ELM can avoid such trivial issues by reaching the good solutions analytically and the learning speed of ELM is extremely fast compared to other traditional methods. The universal approximation capability of extreme learning machine has been proved by

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 114–119, 2006.

Huang, et al[10]. In this paper, the ELM algorithm with RBF activation function is given to deal with complex-valued signal called ELM-CRBF equalizer. The simulation results show that the ELM-CRBF equalizer is superior to CRBF [2] and CMRAN [3] equalizers in terms of symbol error rate (SER), learning speed and ease of implementation.

This paper is organized as follows. Section 2 introduces the complex extreme learning machine with RBF (ELM-CRBF). Section 3 shows the performance comparison of ELM-CRBF equalizer with the CRBF and CMRAN equalizers for a typical nonlinear channel equalization problem. Section 4 is the conclusions of this paper.

2 Proposed Extreme Learning Machine with Complex-Valued RBF Kernels

Given a series of complex-valued training samples $(\mathbf{z}_i, \mathbf{y}_i)$, $i = 1, 2, \cdots, N$, where $\mathbf{z}_i \in \mathbf{C}^n$ and $\mathbf{y}_i \in \mathbf{C}^m$, the actual outputs of the single-hidden-layer feedforward network (SLFN) with RBF activation function for these N training data is calculated by

$$\sum_{k=1}^{\tilde{N}} \beta_k \phi_k(\mathbf{z}_i, \mu_k, \sigma_k) = \mathbf{o}_i, \quad i = 1, \cdots, N, \tag{1}$$

where $\phi_k(\mathbf{z}_i, \mu_k, \sigma_k)$ is the response of the k-th hidden neuron:

$$\phi_k(\mathbf{z}_i, \mu_k, \sigma_k) = \exp\left(-\frac{(\mathbf{z}_i - \mu_k)^H(\mathbf{z}_i - \mu_k)}{\sigma_k^2}\right) \tag{2}$$

μ_k and σ_k are the center and width of the k-th hidden neuron, respectively. $\beta_k \in \mathbf{C}^m$ is the complex output weight vector connecting the k-th hidden neuron and the output neurons. H presents the complex conjugate transposition. The complex SLFNs with $\hat{N}$ hidden neurons can approximate the N input samples with zero error means that $\sum_{i=1}^{N} \|\mathbf{o}_i - \mathbf{y}_i\| = 0$, in other words, the network parameters β_i, μ_i and σ_i need to be found such that

$$\sum_{k=1}^{\tilde{N}} \beta_k \phi_k(\mathbf{z}_i, \mu_k, \sigma_k) = \mathbf{y}_i, \quad i = 1, \cdots, N, \tag{3}$$

The above N equations can be written compactly as

$$\mathbf{H}\beta = \mathbf{Y} \tag{4}$$

where

$$\begin{aligned} &H(\mu_1, \cdots, \mu_{\tilde{N}}, \mathbf{z}_1, \cdots, \mathbf{z}_N, \sigma_i, \cdots, \sigma_{\tilde{N}}) \\ &= \begin{bmatrix} \phi_k(\mathbf{z}_1, \mu_1, \sigma_1) & \cdots & \phi_k(\mathbf{z}_1, \mu_{\tilde{N}}, \sigma_{\tilde{N}}) \\ \vdots & \cdots & \vdots \\ \phi_k(\mathbf{z}_N, \mu_1, \sigma_1) & \cdots & \phi_k(\mathbf{z}_N, \mu_{\tilde{N}}, \sigma_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}} \end{aligned} \tag{5}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}{}^T \end{bmatrix}_{\tilde{N}\times m} \quad \text{and} \quad \mathbf{Y} = \begin{bmatrix} \mathbf{y}_1^T \\ \vdots \\ \mathbf{y}_N^T \end{bmatrix}_{N\times m} \tag{6}$$

The matrix $\mathbf{H}$ is called hidden layer output matrix and the number $\tilde{N}$ of the hidden neurons is usually much less than the number N of training samples. Linearly similar to Huang et. al. [6, 7] the hidden neuron parameters (complex-valued centers μ_i and the impact width σ_i) can be randomly generated and for randomly generated hidden neuron parameters we can get the least-squares solution $\hat{\beta}$ of the linear system $\mathbf{H}\beta = \mathbf{y}$ with minimum norm of output weights β, which usually tend to have good generalization performance:

$$\hat{\beta} = \mathbf{H}^{\dagger}\mathbf{Y} \tag{7}$$

where matrix $\mathbf{H}^{\dagger}$ is the Moore-Penrose generalized inverse. The three-step complex ELM with RBF(ELM-CRBF) algorithm can be summarized as follows:

Algorithm ELM-CRBF: Given a training set $\aleph = \{(\mathbf{z}_i, \mathbf{y}_i)|\mathbf{z}_i \in \mathbf{C}^n, \mathbf{y}_i \in \mathbf{C}^m, i = 1, \cdots, N\}$, RBF activation function $\phi(z)$, and hidden neuron number $\tilde{N}$,

1. Randomly choose the complex hidden neuron center μ_k and width σ_k, $k = 1, \cdots, \tilde{N}$.
2. Calculate the hidden layer output matrix $\mathbf{H}$.
3. Calculate the complex output weight β using formula (7).

3 Performance Evaluation

A third order complex channel model as considered by Chen et al. [11] for 4-QAM signaling is given by

$$A(z) = (0.7409 - j0.7406)(1 - (0.2 - j0.1)z^{-1})((1 - (0.6 - j0.3)z^{-1}) \tag{8}$$

The noise variance is $\sigma_e^2 = 0.06324$ (SNR=15dB). The equalizer dimension was set to $m = 1$ and the equalizer decision delay was $\tau = 0$ same as in Chen et al. [11]. The ELM-CRBF and CMRAN equalizers are trained using 1000 samples at 15dB SNR. The CRBF equalizer is trained using 10000 samples with 50 hidden neurons. For testing, 1000 000 test data of different SNRs were used and the symbol error rate was evaluated.

In the simulation, we use 40 hidden neurons for ELM-CRBF equalizer and CMRAN equalizer builds 49 hidden neurons at the end of training process. Fig. 1 shows the equalizer output distribution for all the equalizers. CRBF equalizer can't obtain satisfying performance with more training data and hidden neurons. While ELM-CRBF, CMRAN and Bayesian equalizers can separate the output data into four groups clearly. Fig. 2 shows the comparison of the symbol error rate (SER) between Bayesian, CRBF, CMRAN and ELM-CRBF equalizers. We

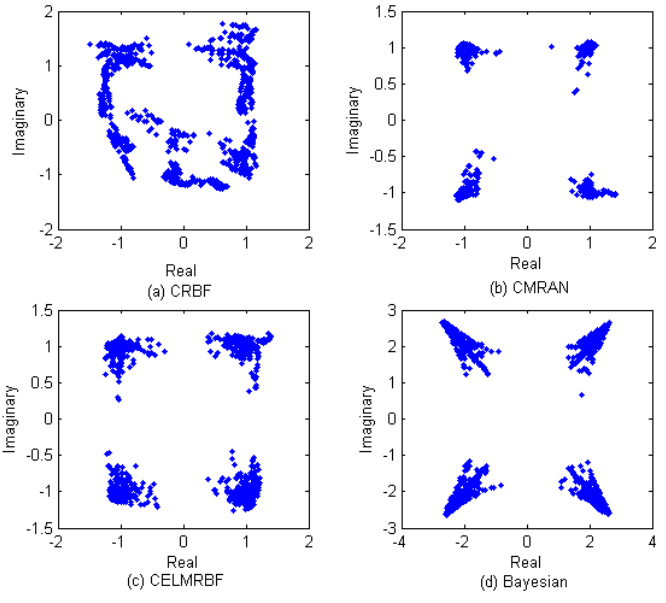


Fig. 1. Eye diagram of the equalizer outputs: (a)CRBF (b) CMRAN (c)ELM-CRBF (d)Bayesian

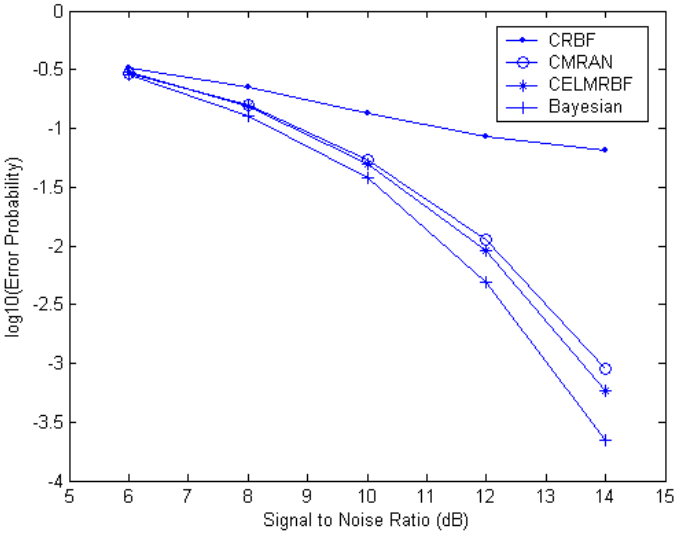


Fig. 2. Error probability for CRBF, CMRAN, ELM-CRBF and Bayesian

can see that ELM-CRBF is always better than CRBF and CMRAN but is not as good as Bayesian method, which is not surprising since Bayesian is supposed to be the optimal one.

Table 1 shows the comparison of CPU time of CRBF, CMRAN, ELM-CRBF and Bayesian equalizers to show the equalizer complexity[1]. The result is obtained using a P4/3GHZ personal computer. It can be seen that ELM-CRBF equalizer takes only 0.047s while CMRAN takes 45.67s and CRBF 67.31s for the training.

Table 1. Complexity comparison of different equalizers

Algorithms	Neurons	Number of Data		CPU time	
		training	testing	training(sec)	testing(msec/sample)
ELM-CRBF	40	1000	10^6	0.047	0.004
CMRAN	49	1400	10^6	45.67	0.165
CRBF	50	10^4	10^6	67.31	0.185
Bayesian	64	–	10^6	–	0.159

4 Conclusions

In this paper, extreme learning machine with complex-valued RBF activation function is proposed to solve complex-valued channel equalization problem. The hidden neuron parameters of SLFNs are randomly chosen and Moore-Penrose generalized inverse is used to determine the output layer weights. To compare with some existing RBF neural network (CRBF and CMRAN), the ELM-CRBF equalizer outperforms than these RBF equalizers in terms of SER and training speed.

References

1. Huang, G.B., Chen, Y.Q., Babri, H.A.: Classification Ability of Single Hidden Layer Feedforward Neural Networks. IEEE Transactions on Neural Networks **11** (2000) 799–801
2. Cha, I., Kassam, S.A.: Channel Equalization Using Adaptive Complex Radial Basis Function Networks. IEEE Journal Selected Areas in Communications **13** (1995) 122–131
3. Jianping, D., Sundararajan, N., Saratchandran, P.: Communication Channel Equalization Using Complex-Valued Minimal Radial Basis Function Neural Networks. IEEE Transactions on Neural Networks **13** (2002) 687–696
4. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme Learning Machine: A New Learning Scheme of Feedforward Neural Networks. In: Proceedings of International Joint Conference on Neural Networks (IJCNN2004), Budapest, Hungary (25-29 July, 2004)
5. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme Learning Machine: Thoery and Applications. Neurocomputing (in press) (2006)
6. Huang, G.B., Siew, C.K.: Extreme Learning Machine: RBF Network Case. In: Proceedings of the Eighth International Conference on Control, Automation, Robotics and Vision (ICARCV 2004), Kunming, China (6-9 Dec, 2004)

[1] Source codes of ELM can be downloaded from http://www.ntu.edu.sg/home/egbhuang

7. Huang, G.B., Siew, C.K.: Extreme Learning Machine with Randomly Assigned RBF Kernels. International Journal of Information Technology **11** (2005)
8. Huang, G.B., Zhu, Q.Y., Mao, K.Z., Siew, C.K., Saratchandran, P., Sundararajan, N.: Can Threshold Networks Be Trained Directly? IEEE Transactions on Circuits and Systems II **53** (2006)
9. Li, M.B., Huang, G.B., Saratchandran, P., Sundararajan, N.: Fully Complex Extreme Learning Machine. Neurocomputing **68** (2005) 306 –314
10. Huang, G.B., Chen, L., Siew, C.K.: Universal Approximation Using Incremental Constructive Feedforword Networks with Random Hidden Nodes. to appear in IEEE Transactions on Neural Networks (2006)
11. Chen, S., Mclaughlin, S., Mulgrew, B.: Complex-Valued Radial Basis Function Networks, Part II: Application to Digital Communications Channel Equalization. Signal Processing **36** (1994) 165–188

Nonlinear Channel Equalization Using Concurrent Support Vector Machine Processor

Jae Woo Wee[1], Tae Seon Kim[2], Sung Soo Dong[3], and Chong Ho Lee[4]

[1] Korean Intellectual Property Office, Dunsan-dong,
Seo-gu, DaeJeon, 302-701, Korea
wizet2000@empal.com
[2] Catholic University of Korea, Bucheon, 420-743, Korea
tkim@catholic.ac.kr
[3] Yongin-Songdam College, Yongin, 449-710, Korea
ssdong@ysc.ac.kr
[4] Inha University, Yonghyun-dong, Nam-gu, Incheon, 402-751, Korea
chlee@inha.ac.kr

Abstract. We developed a high-speed concurrent support vector machine (CSVM) processor for real-time nonlinear channel equalization. All phases of the recognition process, including kernel computing, learning, and recall of the support vector machine (SVM) are performed on a single chip. The concurrent operation of this CSVM using a parallel architecture of elements allows it to achieve high speed. The hardware-friendly kernel adatron (KA) SVM learning algorithms are embedded on a chip. The results of the nonlinear channel equalization obtained by the KA algorithm are compared with those obtained by the quadratic programming (QP) method. The CSVM using the KA learning algorithm is designed and implemented using the FPGA chip. The CSVM processor performs 20% faster than the existing SVM processors.

1 Introduction

Nonlinear channel equalization is a major issue in digital communications, because it affects the transmitted sequence by causing both linear (inter-symbol interference) and nonlinear (amplifiers and converters) distortions. The support vector machine (SVM) has shown promise in the area of nonlinear equalization or nonlinear detection [1]. The objective of this study is to design and implement a nonlinear channel equalizer based on the support vector machine. The equalization performances obtained using the SVM and KA learning technique are verified, and a fast hardware architecture is presented based on parallelism, a shared bus and distributed memory. The circuit is implemented using hardware description language (HDL). We chose the FPGA chip for the hardware realization of our channel equalizer.

The application of the SVM to nonlinear channel communication systems was first proposed by Sebald *et al.* [1], who compared the equalization performance between the SVM and Artificial Neural Network (ANN). According to [1], the SVM shows similar or superior performance as compared to the ANN. They trained an SVM using

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 120 – 127, 2006.

the QP method, which is computationally intensive. In this study, to avoid the high computational cost of training the SVM associated with its solving a quadratic optimization problem, we use a fast and simple procedure known as the Kernel Adatron (KA) algorithm [2]. This procedure leads to exactly the same solution, but with an exponential rate of convergence in the number of iterations.

Nonlinear channel equalization requires specialized hardware, owing to the intensive and real-time computations involved. Various hardware implementations of the SVM have recently been proposed by many researchers based on parallel architectures, such as that used in the ANN chip. These hardware architectures of the SVM have been published, and have involved the development of both analog and digital hardware [3], [4], [5]. The Kerneltron [3], which internally performs analog and massively parallel kernel computation, successfully deals with real-time applications involving object detection and recognition. This analog hardware performs the recall processes on-chip, leaving the learning process off-chip. A digital hardware version of the SVM (DSVM) [4], [5] proposes a hardware-friendly SVM learning algorithm and performs the learning process on-chip. However, this digital hardware performs the kernel computation off-chip. The concurrent SVM (CSVM) processor is the first SVM processor to conduct all three processes of the SVM on a single chip.

2 Support Vector Machine

The SVM expresses the classification or regression output in terms of a linear combination of examples in the training data, in which only a fraction of the data points, called 'support vectors', have non-zero coefficients. The support vectors thus capture all the relevant data contained in the training set.

Quadratic programming in original SVM training algorithm is not suitable for hardware implementation, due to its complexity and large memory consumption. Thus we use kernel adatron (KA) [2] learning algorithms can be implemented on a silicon chip since these algorithms make use of recursive-updating equations instead of quadratic programming.

KA algorithm uses the gradient ascent routine to maximize the margin in feature space. This algorithm is easy to implement and theoretically guaranteed to converge in a finite number of steps to the maximal margin.

Consider a two-class classification problem with training set $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$, where $\boldsymbol{x}_i \in \Re^d$ and $y_i = \{+1, -1\}$. In its basic form, an SVM classifies the pattern vector, $\boldsymbol{x}$, into class, y, based on the support vectors, $\boldsymbol{x}_m$, and corresponding classes, y_m, as in

$$f(\mathrm{x}) = sign\left(\sum_{m=1}^{M} \alpha_m y_m K(\mathrm{x}, \mathrm{x}_m) + b\right) \tag{1}$$

Here, $\{\alpha_i\}$ are Lagrange multipliers and the kernel function $K(\cdot, \cdot)$ realizes a dot product in the feature space. KA algorithm is described as follows.

{Initialization} Set $\alpha_i = 0$ for all i and select learning rate parameter η
{Learning loop}
repeat
for i =1,…n

$$z_i = \sum_{j=1}^{l} \alpha_j y_j K(x_i, x_j), \qquad \Delta\alpha_i = \eta(1 - y_i z_i)$$

if $(\alpha_i + \Delta\alpha_i) > 0$ **then** $\alpha_i \leftarrow \alpha_i + \Delta\alpha_i$ **else** $\alpha_i \leftarrow 0$
end for
{Termination condition}
until (maximum number of iterations reached or

$$\gamma = 0.5 \times [\min_{\{i | y_i = +1\}} (z_i) - \max_{\{i | y_i = -1\}} (z_i)] \approx 1 \ )$$

3 CSVM Architecture

3.1 System Overview

The hardware architecture, which is composed of the Support Vector Elements (SVEs) holding the support vector streams, is shown in Fig. 1. All of the SVEs are connected via a shared data bus, and the parallel operation of the SVEs generates high throughput. The extension of the SVEs using multiple-chips is also possible, when numerous training patterns are involved.

Operating phases of the hardware consists of loading, kernel computation, learning, and recall. The computation of kernel function requires data from one SVE sent to other SVEs via shared data bus and the kernel computations are simultaneously done at all SVEs. The kernel computations are performed faster than those done sequentially by n times. In the learning phase, alphas and bias value are generated in the Learning Element (LE) and the updated parameters are sent to corresponding SVEs. The sign of margin that determines classes of the test data is computed in parallel during recall phase. Also recall phase by parallel operation is faster than by sequential operation by n times.

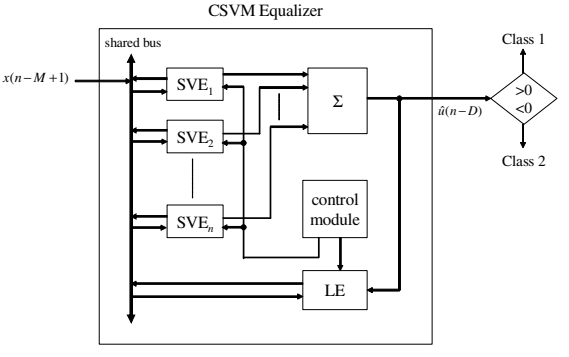


Fig. 1. CSVM architecture for channel equalization

3.2 Building Blocks

The SVE shown in Fig. 2 consists of memory, computing module and interface with shared data bus. The two's complementer is the substitution of a multiplier, because the selection signal, y, is either 1 or –1. When an SVE sends data to other elements, the signal, send_en, of SVE_i becomes active high. Three state buffers are used to prevent the sending of conflicting data over the shared bus. The values of y_j and α_j are saved in the registers, by setting the enable signals yj_en and alpha_en to active high when the requested values have arrived at the requesting register from memory. The value contained in a register can be maintained, by setting the enable signal to active low.

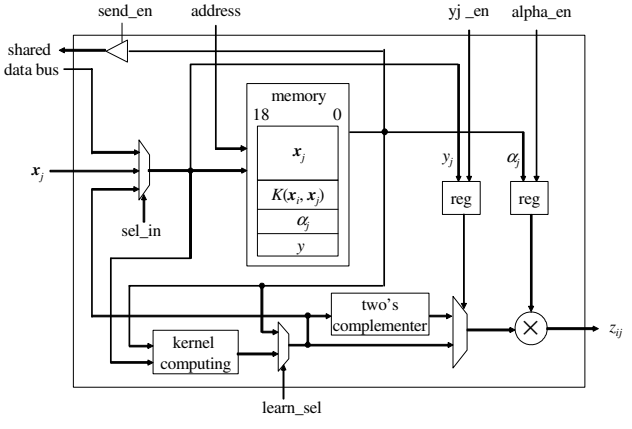


Fig. 2. Support Vector Element

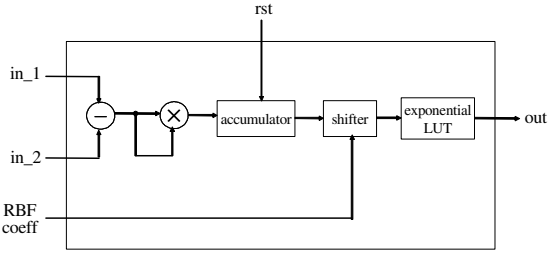


Fig. 3. Kernel Computing Module

The Kernel Computing Element (KCE) shown in Fig. 3 includes the RBF kernel functions: $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(- c \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2)$, where $c = 1/2\sigma^2$. The multiplication of the coefficient, c, is performed by a shifter, since good results are obtained by using only those coefficients that are multiples of 2. A lookup table (LUT) generates the exponential function of the RBF kernel. Eliminating the inputs of $\exp(-f(x))=0$ from the LUT saves additional memory space. The appropriate number of address bits for the LUT is determined by experiment.

The kernel values computed during the kernel computing phase are used for the learning phase. The KA learning algorithms are described in chapter 2. The parameters required by the LE are sent to the SVE_i via the shared data bus. When the termination condition of the learning phase is satisfied, the parameters obtained are saved in the memory of SVE_i.

4 Results

4.1 Channel Equalization Problem

Band-limited communications channels driven at high data rates often display intersymbol interference (ISI) due to their dispersive time response [6]. All of the unknown nonlinear effects of the components involved(transmitter, channel and receiver) are modeled as finite-impulse response (FIR) filters, plus a Gaussian distributed noise, e, with zero mean and variance, σ_e^2, as in the following equations.

$$\tilde{x}(n) = \sum_{k=0}^{N} h_k u(n-k), \quad \hat{x}(n) = \sum_{p=0}^{P} c_p \tilde{x}^p(n), \quad x(n) = \hat{x}(n) + e(n) \tag{2}$$

where the transmitted symbol sequence, $u(k)$, is assumed to have statistically independent and equally probable values from {+1, -1}, h_k are the channel coefficients and $\{c_p\}$ are the polynomial coefficients.

The task of the equalizer is to obtain an estimate of $u(t - D)$, denoted by $\hat{u}(t - D)$, given the channel outputs, $\boldsymbol{x}(t)$, where D is the time delay of the signal transmitted through the physical channel. To facilitate this task, we form the channel output vector

$$\mathrm{x}(t) = [x(t), x(t-1), ..., x(t-M+1)]^T \tag{3}$$

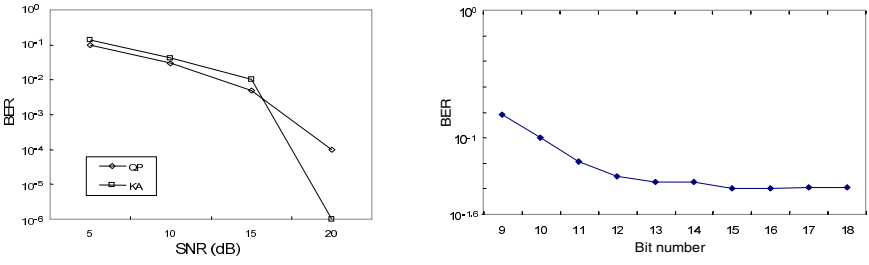
where M is the equalizer order, and we wish to compute $\hat{u}(t - D)$ given $\boldsymbol{x}(t)$. Varying the delay, D, causes the performance of the equalizer to vary, because the correlation between $u(n - D)$ and $x(n)$ changes with D [1]. Since the channel has an ISI of length N and the equalizer has a feed-forward delay of length $M - 1$, the equalizer output is dependant on $M + N - 1$ channel inputs.

The first model involves the nonlinear channel $\hat{x}(n) = \tilde{x}(n) - 0.9\ \tilde{x}^3(n)$, $\tilde{x}(n) = u(n) + 0.5\ u(n-1)$, additive white Gaussian noise of variance $\sigma_e^2 = 0.2$, and SVM parameter $M = 2$ and $\sigma^2 = 0.5$ with the RBF kernel. We refer to these models with $D = 0$ and 2 as Models 1 and 2, respectively. The results are an average of ten trials with 500 training samples and 2400 samples in the test set.

We compared our results with those obtained in [4], and found that the results obtained using the KA were superior to those in this study. The CSVM with KA learning gives rise to error probabilities of 17.75% and 3.95% for Models 1 and 2, respectively. The results obtained with Model 2 have lower errors than the 4.2% error reported in [4] and those obtained with Model 1 are similar.

In the second example, we perform a comparison between the bit error rate (BER) of the KA with the RBF kernel and the BER of the QP, as a function of the SNR for

the channel $\hat{x}(n) = \tilde{x}(n) + 0.2\ \tilde{x}^2(n)$, $\tilde{x}(n) = 0.3482\ u(n) + 0.8704\ u(n-1) + 0.3482\ u(n-2)$, $M = 3$, $D = 1$, $\sigma^2 = 1$. The number of training samples is again 500, whereas the number of trials is 10 with 2000 test samples. As shown in Fig. 4(a), the KA method results in a similar BER to that obtained in [1] using the QP method for an SNR of 5 to 10dB. Above 15dB, the BER of the KA outperforms that of the QP. Note that the RBF kernel obtains better results in the learning process than the polynomial kernel used in [1] for various SNRs.



(a) (b)

Fig. 4. (a) Average BER for the KA learning algorithm vs. QP method. (b) BER with respect to the number of bits in the fraction part.

The relationship between the error performance and the number of bits in the fraction part is given in Fig. 4(b). We found from software simulations that increasing the length of the fraction of data beyond 12 bits does not improve the performance of the equalizer. The BER of the equalizer dramatically increases when this length is lower than 12 bits.

4.2 Implementation on FPGA

The CSVM is designed using Verilog HDL and implemented on a Xilinx Virtex2 FPGA XC2V6000. This chip contains 6M system gates, 144 18×18 multiplier blocks and 144 18-Kbit selected-RAM blocks for a total capacity of 2592 Kbits of RAM. The improved placement and routing obtained on this chip is due to the use of dedicated multiplier blocks and RAM blocks.

Table 1 shows the number of circuit components of the DSVM and CSVM obtained by the synthesis report. The CSVM of 18 data widths is designed for 32 training samples. The CSVM, which has the kernel computing ability, requires a few more components than the DSVM, which is not able to compute the kernel values on a single chip. The number of processing cycles for each phase after being placed and routed in the FPGA chip is shown in Table 2. In this table, learning processing cycles are required at every learning iteration and recall processing cycles are required for one test sample. The kernel computing phase requires more processing cycles than the learning phase when the data size, n, is more than 9 and the data dimensions, d, of both phases are the same.

Table 1. HDL synthesis report of DSVM and CSVM

SVM processor	Registers	Counters	Mux	Tristates	Decoders	Add/Sub	Comps	Shifters
DSVM	213	1	210	20	1	165	34	0
CSVM	279	3	161	33	0	124	92	42

Table 2. Processing cycles for each phase

Phase	Processing cycles
Loading	$n\,(d+1)+7$
Kernel computing	$n\,(d+9)$
Learning	$18\,n$
Recall	8

It takes a mere 4 ms for the completion of all of the phases of the SVM, when the learning process is terminated at 200 iterations using a clock cycle time of 28 ns. In a comparison of the learning processing time, the CSVM was found to perform faster than the DSVM. The processing times for the learning phases of the CSVM and DSVM were 115000 and 140000 clock cycles, respectively.

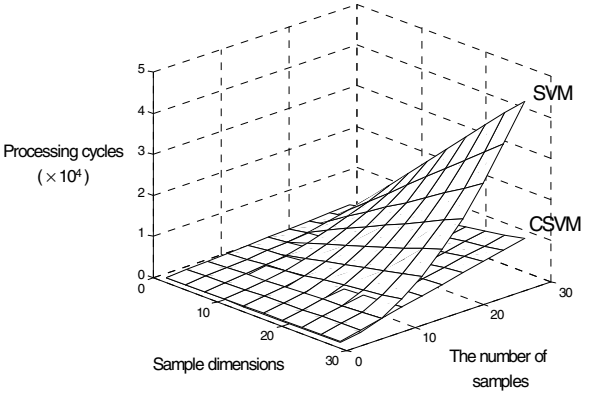


Fig. 5. Processing cycles using sequential SVM process vs. CSVM

Fig. 5 shows the number of processing cycles of the kernel computing phase as the number of samples and sample dimension increase. The exponentially increasing curved surface indicates the number of processing cycles required for the kernel computing phase without using the hardware. In contrast, the CSVM chip allows the

computing time to increase linearly, as shown in the lower surface of Fig. 5. Note that the CSVM with the parallel structure and concurrent operation provides outstanding performance for larger scale problems.

5 Conclusion

In this paper, we proposed the SVM processor, which is suitable for nonlinear channel equalization and is realized using the FPGA chip. The comparison between the KA and QP algorithms shows that the KA has a faster and simpler learning ability than the QP method, while affording equivalent performance. Its parallel and on-chip kernel computing architecture makes the CSVM extremely fast. The intensive kernel computation is quickly performed on the hardware module, together with the learning and recall operations of the SVM. Whereas the computational complexity of the SVM with the KA increases quadratically, that of the CSVM chip increasing linearly. The CSVM, which is the first SVM processor to perform all of the phases of the SVM, is suitable for on-line and stand-alone applications, such as a communication equalizer.

The area complexity problem arises when the data to be processed includes too many samples needing to be classified. One way to reduce the chip area would be to replace the processing architecture by one involving serial processing. By doing so, a considerable amount of chip area could be saved, but the processing time would increase and would be unacceptable in most applications. Another way to decrease the required chip area is to lower the number of data bits in the decimal fractions.

References

1. Sebald, D. J., Buchlew, J. A.: Support Vector Machine Techniques for Nonlinear Equalization. IEEE Trans. Signal Processing 48 (2000) 3217-3226
2. Friess, T.T., Cristianini, N., Campbell, C.: The Kernel-adatron Algorithm: A Fast and Simple Learning Procedure for Support Vector Machines. 15th International Conference on Machine Learning, Wisconsin, USA (1998) 188-196
3. Genov, R., Cauwenberghs, G.: Kerneltron: Support Vector "Machine" in Silicon. IEEE Transactions on Neural Networks 14 (2003) 1426 -1434
4. Anguita, D., Boni, A., Ridella, S.: A Digital Architecture for Support Vector Machines: Theory, Algorithm and FPGA Implementation. IEEE Transactions on Neural Networks 14 (2003) 993-1009
5. Anguita, D., Boni, A., Ridella, S.: Digital Kernel Perceptron. Electronics Letters 38 (2002) 445-446
6. Proakis, J. G.: Digital Communications. 3rd edn. McGaw-Hill, New York (1995)

Recursive Complex Extreme Learning Machine with Widely Linear Processing for Nonlinear Channel Equalizer

Junseok Lim[1], Jaejin Jeon[2], and Sangwook Lee[3]

[1] Dept. of Electronics Eng., Sejong University,
Kunja, Kwangjin, 98,143-747, Seoul, Korea
jslim@sejong.ac.kr
[2] Samsung Electronics Co.,Ltd.
[3] LG Electronics Inc.

Abstract. Recently, a new learning algorithm for the feedforward neural network named the complex extreme learning machine (C-ELM) which can give better performance than traditional tuning-based learning methods for feedforward neural networks in terms of generalization and learning speed has been proposed by Huang et al. In this paper, we propose a new widely linear recursive C-ELM algorithm for nonlinear channel equalizer. The proposed algorithm improves its performance especially in case of real valued modulation such as BPSK and PAM. The computer simulation results demonstrate the improvement in performance achievable with the proposed equalization algorithm.

1 Introductions

In high-speed digital communication systems, equalizers are used very often at receivers to recover the original symbols from the received signals. Real-valued neural network models such as feedforward neural networks, radial basis function (RBF) networks and recurrent neural networks have been successfully used for solving nonlinear channel equalization problems as neural networks are well suited for nonlinear classification problems [1]. Complex-valued neural networks have attracted considerable attention in channel equalization applications in the past decade. Recently, a new learning algorithm for single-hidden-layer feedforward neural network (SLFN) named the complex extreme learning machine (C-ELM) has been proposed by Huang et al. [2]. Unlike traditional approaches (such as BP algorithms), which may face difficulties in manually tuning control parameters (learning rate, learning epochs, etc), C-ELM avoids such issues and reaches good solutions analytically. The learning speed of C-ELM is extremely fast compared to other traditional methods. The simulation results in [2] have shown that the C-ELM equalizer is superior to CRBF, CMRAN and CBP equalizers in terms of symbol error rate (SER) and learning speed. However, C-ELM equalizer still has room to improve its performance in real value modulation and complex channel.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 128–134, 2006.

In recent, a number of papers about real value modulation and complex channel were published [3],[4]. Those derived the improper property in real value modulation signals and presented communication applications where accounting for the improper nature of signals was rewarded with significant performance gains. Widely linear method was proposed to effectively handle the improper signals [3]. The exemplary communication applications with the improper nature are PAM, BPSK, GMSK and OQAM, etc.

In this paper, we propose a new widely linear recursive C-ELM algorithm for nonlinear channel equalizer. The proposed algorithm improves its performance especially in case of real valued modulation such as BPSK and PAM. This paper is organized as follows. Section 2 summarizes C-ELM. Section 3 summarizes improper signals and widely linear processing. Section 4 proposes a recursive C-ELM equalizer with widely linear processing. The performance comparison of C-ELM with the proposed C-ELM equalizers shows in Section 5. Discussions and conclusions are given in Section 6.

2 Complex Extreme Learning Machine (C-ELM) Algorithm

Given a series of complex-valued training samples ($\mathbf{z}$(i),d(i)), i=1,2,. . . ,N, where $\mathbf{z}(\mathrm{i}) \in \mathrm{C}^p$ and $\mathrm{d(i)} \in \mathrm{C}^1$, the actual outputs of the single-hidden-layer feedforward network (SLFN) with complex activation function $\mathrm{g}_c(\mathrm{z})$ for these N training data is given by

$$\sum\_k = 1^{\tilde{N}} \beta_k g_c(\mathbf{w}_k \mathbf{z}_i + b_k) = o_i, \quad i = 1, \cdots, N,, \tag{1}$$

where column vector $\mathbf{w}_k \in \mathrm{C}^p$ is the complex input weight vector connecting the input layer neurons to the kth hidden neuron, $\beta_k \in \mathbf{C}^1$ the complex output weight vector connecting the kth hidden neuron and the output neurons, and $\mathrm{b}_k \in \mathrm{C}^1$ is the complex bias of the kth hidden neuron. $\mathbf{w}_k \bullet \mathbf{z}_i$ denotes the inner product of column vectors $\mathbf{w}_k$ and $\mathbf{z}_i$. $\mathrm{g}_c(\mathrm{z})$ is a fully complex activation function. The above N equations can be written compactly as

$$\mathbf{H}\beta = \mathbf{o}, \tag{2}$$

and in practical applications the number $\tilde{N}$ of the hidden neurons is usually much less than the number N of training samples and $\mathbf{H}\beta \neq \mathbf{d}$, where

$$\mathbf{H}(\mathbf{w}_1, \cdots, \mathbf{w}_{\tilde{N}}, \mathbf{z}_1, \cdots, \mathbf{z}_{\tilde{N}}, b_1, \cdots, b_{\tilde{N}}) = \begin{bmatrix} g_c(\mathbf{w}_1 \mathbf{z}_1 + b_1) & \cdots g_c(\mathbf{w}_{\tilde{N}} \mathbf{z}_1 + b_{\tilde{N}}) \\ \vdots & \ddots \vdots \\ g_c(\mathbf{w}_1 \mathbf{z}_N + b_1) & \cdots g_c(\mathbf{w}_{\tilde{N}} \mathbf{z}_N + b_{\tilde{N}}) \end{bmatrix}, \tag{3}$$

$$\beta = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_{\tilde{N}} \end{bmatrix}, \quad \mathbf{o} = \begin{bmatrix} o_1 \\ \vdots \\ o_N \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} d_1 \\ \vdots \\ d_N \end{bmatrix}, \tag{4}$$

The complex matrix **H** is called the hidden layer output matrix. As analyzed by Huang et al [2] for fixed input weights $\mathbf{w}_i$ and hidden layer biases b_i, we can get the least squares solution $\hat{\beta}$ of the linear system $\mathbf{H}\beta$=**d** with minimum norm of output weights b, which usually tend to have good generalization performance. The resulting $\hat{\beta}$ is given by

$$\hat{\beta} = \mathbf{H}^{+}\mathbf{d}, \tag{5}$$

where complex matrix $\mathbf{H}^{+}$ is the Moore–Penrose generalized inverse of complex matrix H. Thus, ELM can be extended from the real domain to a fully complex domain in a straightforward manner. The three steps in the fully complex ELM (C-ELM) algorithm can be summarized as follows:

Algorithm C-ELM: Given a training set N = $\{(\mathbf{z}(i)\ ,\ d(i))|\ \mathbf{z}(i) \in C^p,\ d(i) \in C^1,\ i = 1, \ldots ,N\}$, complex activation function $g_c(z)$, and hidden neuron number $\tilde{N}$;

1. Randomly choose the complex input weight wk and the complex bias b_k, k $= 1, \ldots , \tilde{N}$.
2. Calculate the complex hidden layer output matrix **H**.
3. Calculate the complex output weight β using formula (5).

3 Improper Signals and Widely Linear Processing

A complex process $x[.]$=$x_R[.]$+$j\mathrm{x}_I[.]$, $x_R[.]$, $\mathrm{x}_I[.] \in R$, is said to be wide-sense stationary (WSS) if the quadrature components $x_R[.]$ and $\mathrm{x}_I[.]$ are jointly WSS, i.e., the means $E\{x_R[k]\}$and $E\{x_I[k]\}$ are constant and all auto- and crosscorrelations of $x_R[.]$ and $\mathrm{x}_I[.]$ depend only on time difference but not on absolute time. This is equivalent to a constant complex mean $E\{x[k]\}$, and a complex autocorrelation $\phi_{xx}[m] \equiv E\{x[k+m]x^*[k]\}$ and pseudoautocorrelation

$$\phi_{xx^*}[m] \equiv E\{x[k+m]x[k]\}, \tag{6}$$

which both depend only on the time difference. Furthermore, a complex WSS process is referred to as proper if its pseudoautocorrelation, which may be interpreted as crosscorrelation between the processes $x[.]$ and $x^*[.]$, vanishes identically

$$\phi_{xx^*}[m] = 0\ \forall m \in Z, \tag{7}$$

On the other hand, an improper (rotationally variant, noncircular) process exhibits a nonvanishing pseudoautocorrelation [4]. Linear estimation of a stochastic process a[.] via an improper complex observation x[.] can be improved by also utilizing x*[.], i.e., x[.] as well as x*[.] are filtered with filters $f_1[.]$ and , $f_2[.]$ respectively, and the results are linearly combined. Applying this strategy of WLP (Widely Linear Processing) [5]—also denoted as conjugate linear filtering in earlier work [6]—is equivalent to joint optimum linear processing of the real and imaginary part of the observation. On the other hand, if the observation

and the random variable to be estimated are proper, WL estimation yields the same performance as conventional complex linear estimation [7].

In neural network, it is essential to consider the nonlinear effects on the output values of their neurons. In [8], S. Oh and Y. Lee have proven that the correlation coefficient between two jointly Gaussian random variables decreases under the element-wise continuous nonlinear transformations that can be approximated to piecewise linear functions. Therefore, improper property is effective in C-ELM.

4 Widely Linear RLS Based ELM Equalizer

In the following, we discuss C-ELM adaptation strategies for the proposed widely linear structures. First, we derive a recursive least square (RLS) algorithm for adaptation based on WLE. For this, we introduce a total coefficient vector $\beta_{wl}(k) = \left[\beta_1^T(k)\ \beta_2^T(k)\right]^T$where $\beta_1(k)$and $\beta_2(k)$are defined similarly as in Section 3, but now depend also on discrete time k. The error signal of WLE is obtained from $e(k) = \mathbf{h}_{wl}^T(k)\beta_{wl}(k) - d(k)$, where the total received vector is given by $\mathbf{h}_{wl}(k) = [\mathbf{h}(k)\ \mathbf{h}^*(k)]and\mathbf{h}(k) = \left[g_c(\mathbf{w}_1\mathbf{z}_1 + b_1) \cdots g_c(\mathbf{w}_{\tilde{N}}\mathbf{z}_N + b_{\tilde{N}})\right]^T$. Applying the RLS in [9] to the problem at hand yields

$$\mathbf{\Psi}(k) = \frac{\mathbf{P}(k-1)\mathbf{h}_{wl}(k)}{1 + \mathbf{h}_{wl}^T(k)\mathbf{P}(k-1)\mathbf{h}_{wl}(k)}, \tag{8}$$

$$\beta_{wl}(k) = \beta_{wl}(k-1) + \mathbf{\Psi}(k)e(k), \tag{9}$$

$$\mathbf{P}(k) = \mathbf{P}(k-1) - \mathbf{\Psi}(k)\mathbf{h}_{wl}^T(k)\mathbf{P}(k-1). \tag{10}$$

This procedure handles double sized dimension of (2) because of using augmented vector, $\mathbf{h}_{wl}$(k). Using the initialization $\beta_2(0) = \beta_1^*(0)$ with arbitrary $\beta_1(0)$and the result in [10], we can reduces the complexity of the above procedures as follows.

$$\mathbf{\Phi}(k) = \frac{\tilde{\mathbf{P}}(k-1)\mathbf{h}(k)}{1 + \mathbf{h}^T(k)\tilde{\mathbf{P}}(k-1)\mathbf{h}(k)}. \tag{11}$$

$$\beta_1(k) = \beta_1(k-1) + \mathbf{\Phi}(k)\left[\mathbf{h}_{wl}^T(k)\beta_{wl}(k) - d(k)\right]. \tag{12}$$

$$\tilde{\mathbf{P}}(k) = \tilde{\mathbf{P}}(k-1) - \mathbf{\Phi}(k)\mathbf{h}^T(k)\tilde{\mathbf{P}}(k-1). \tag{13}$$

$$\beta_2(k) = \beta_1^*(k). \tag{14}$$

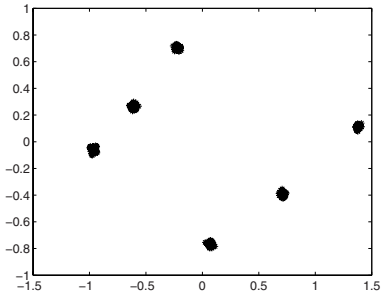
The above procedure handles the same dimension of (2).

5 Performance Evaluation

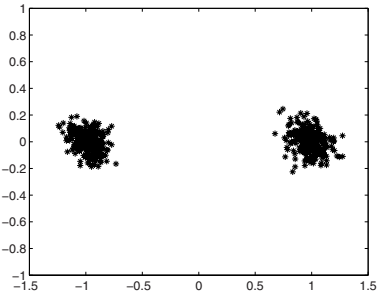
In this section, a well-known complex nonminimum-phase channel model introduced by Cha and Kassam [11] is used to evaluate the performance of the proposed widely linear recursive least square C-ELM equalizer. The channel model is of order 3 with nonlinear distortion for BPSK and 4-PAM signaling. The channel output z(n) (which is also the input of the equalizer) is given by

$$\begin{aligned} z(n) &= o(n) + 0.1o(n)^2 + 0.05o(n)^3 + v(n), \quad v(n) \sim N(0, 0.01) \\ o(n) &= (0.34 - j0.27)s(n) + (0.87 + j0.43)s(n-1) + (0.34 - j0.21)s(n-2)\text{'} \end{aligned} \tag{15}$$

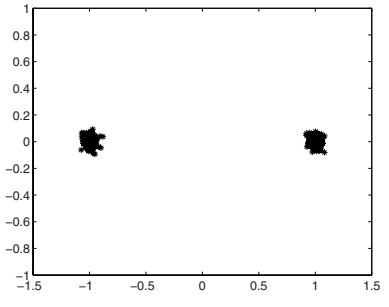
where N(0, 0.01) means the white Gaussian noise (of the nonminimum-phase channel) with mean 0 and variance 0.01. The order of equalizer is chosen as 12. BPSK,{±1}, and 4-PAM, {±1, ±3}, symbol sequence s(n) is passed through the channel. The fully complex activation functions of both C-ELM and the proposed algorithm are chosen as arcsinh(x), where x = $\mathbf{w} \bullet \mathbf{z}$ + b. Both the input weight vectors $\mathbf{w}_k$ and biases b_k of the C-ELM and the proposed one are



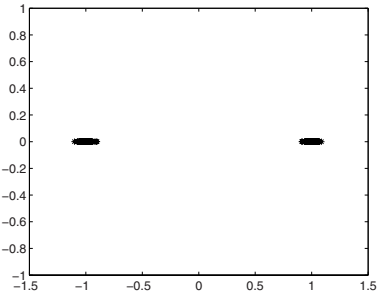
(a) constellationof received signals



(b) constellation of outputs of C-ELM equalizer



(c) constellation of outputs of C-ELM equalizer with double sized order



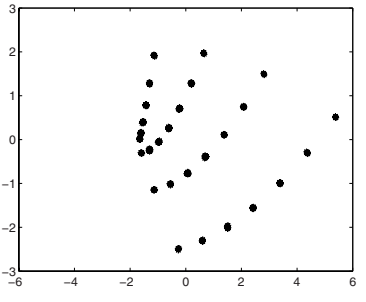
(d) constellation of outputs of the proposed equalizer

Fig. 1. Equalization results in BPSK signaling

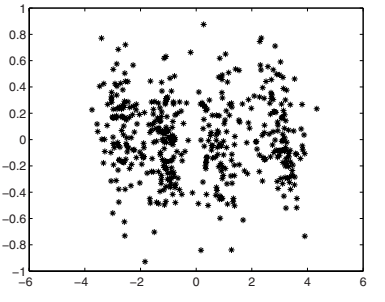
randomly chosen from a complex area centered at the origin with the radius set as 0.1. The hidden neuron numbers of C-ELM and the proposed algorithm are set to 50.

For the performance comparison, we show the results of the three equalizers, normal C-ELM equalizer, normal C-ELM equalizer with the double sized neurons, and the proposed equalizer. They are trained with 1000 data symbols at 16 dB SNR.

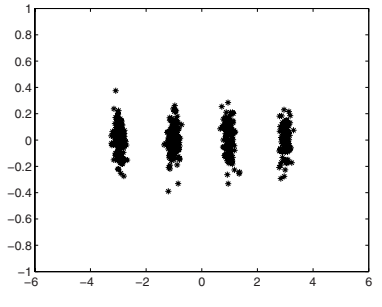
Fig. 1 shows results in BPSK modulation. Fig. 1(a) shows the constellation of received signals. It is severely distorted from the nonlinear channel. Fig. 1 (b) is the result from the normal C-ELM. Fig. 1 (c) shows the result from double size neuron C-ELM equalizer. Fig. 1(d) is the result from the proposed algorithm. These clearly show that the proposed algorithm dramatically improves C-ELM. Comparison between Fig. 1 (b) and (c) says that the performance improvement comes not from double sized neuron but from the improper property. Fig. 2 shows results in 4-PAM modulation. As observed from Fig. 2, the performance of the proposed algorithm is much close to ideal constellation, even though the normal C-ELM degrades its performance.



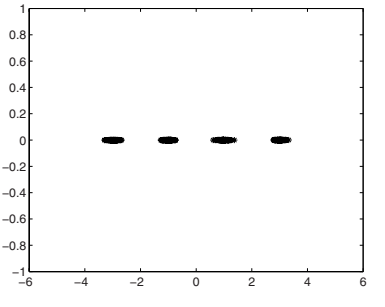
(a) constellation of received signals



(b) constellation of outputs of C-ELM equalizer



(c) constellation of outputs of C-ELM equalizer with double sized order



(d) constellation of outputs of the proposed equalizer

Fig. 2. Equalization results in 4-PAM signaling

6 Discussions and Conclusions

In this paper, we have proposed a new widely linear recursive C-ELM algorithm for equalizer. The proposed algorithm has improved its performance especially in case of real valued modulation such as BPSK and 4-PAM. This algorithm needs almost the same complexity. The simulation results have shown that the proposed algorithm outperforms the C-ELM in equalizer with BPSK and 4-PAM.

References

1. Huang, G.B., Chen, Y.Q., Babri, H.A.: Classification Ability of Single Hidden Layer Feedforward Neural Networks. IEEE Trans. Neural Network 11 (2000) 799–801
2. Huang, M., Saratchandran, G.P., Sundararajan, N.: Fully Complex Extreme Learning Machine. Neurocomputing 68 (2005) 306–314
3. Gelli, G., Paura, L.A., Ragozini, R.P.: Blind Widely Linear Multiuser Detection. IEEE Trans. Comm. Lett. 4 (2000) 187–189
4. Neeser, F.D., Massey, J.L.: Proper Complex Random Processes with Applications to Information Theory. IEEE Trans. Information Theory 39 (1993) 1293–1302
5. Picinbono, B., Chevalier, P.: Widely Linear Estimation with Complex Data. IEEE Trans. Signal Processing 43 (1995) 2030–2033
6. Brown, W.M., Crane, R.B.: Conjugate Linear Filtering. IEEE Trans. Information Theory 15 (1969) 462–465
7. Picinbono, B.: On Circularity. IEEE Trans. Signal Processing 42 (1994) 3473–3482
8. Oh, S., Lee, Y.: Effect of Nonlinear Transformations on Correlation Between Weighted Sums in Multilayer Perceptrons. IEEE Trans. Neural Networks 5 (1994) 508–510
9. Haykin, S.: Adaptive Filter Theory, 3rd ed. Prentice-Hall, Upper Saddle River, NJ (1996)
10. Gerstacker, W.H., Schober, R., Lampe, A.: Receivers with Widely Linear Processing for Frequency-Selective Channels. IEEE Trans. Communication 51 (2003) 1512–1523
11. Cha, I., Kassam, S.A.: Channel Equalization Using Adaptive Complex Radial Basis Function Networks. IEEE J. Sel. Area. Comm. 13 (1995) 122–131

A Study on the Detection Algorithm of QPSK Signal Using TDNN

Sun-Kuk Noh[1] and Jae-Young Pyun[2]

[1] Dept. of Radio Mobile Communication Engineering,
Honam University
[2] Dept. of Information Communication Engineering,
Chosun University
nsk7070@honam.ac.kr

Abstract. Mobile communications and digital wireless communications are requested high frequency use-rates, more efficient data transmission with limited signal power, frequency band. As multiple users share the same frequency in the mobile communications environment, the spectrum efficiency is getting higher. Moreover, as the effect of the velocity of the mobile object and the terrain surroundings get higher, the digital modulation method is required that the character of linear constant amplitude. In this paper, to restore simply and correctly the received signal of quadrature phase shift keying (QPSK) signal in digital wireless communications, we suggest and simulate an algorithm for detection of QPSK signal using time delay neural networks (TDNN). As the results of simulation, the suggested method is confirmed that the phase information of the QPSK signal is recovered simply and correctly in the mobile communications and digital wireless communications.

1 Introduction

Mobile communications and digital wireless communications have the non-linear characters because of the affect of fading or interference by multiple paths. As multiple users share the same frequency in the mobile communications environment, the spectrum efficiency is getting higher. Moreover, as the effect of the velocity of the mobile object and the terrain surroundings get higher, the digital modulation method is required that the character of linear constant amplitude. As fading, noise and frequency reuse-rates are increased, the receiving signal is highly affected by the same or nearby channels interference of the call quality is rapidly getting worse.

One of the plans to solve these problems, the theory of the forward error correction(FEC) and automatic repeat request(ARQ)[1,2] which would apply in the communications department moving to correct the fault, to carry the effective lightings and to improve the S/N rate by downing the affects of the fading, but they have many problems because of the technical limits. Thus, it is a problem that to restore the received signal simply and correctly under this inferior environments.

The quadrature phase shift keying(QPSK) signal which is excellent in determine-ing spectrum efficiency in digital modulation signal[1,2] and the time delay neural

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 135 – 143, 2006.

networks(TDNN) that the class of artificial neural networks(ANN) of the Single layer perceptron structural which consist of the connection intensity, bias, neuron, non-linear transfer function [3-12].

In this paper, to improve the problem, we suggest an algorithm for detection of QPSK signal by using the TDNN, and find symbol error rates of the received signal according to the raised-cosine filter's roll-off factor α by simulation. As the results of simulation, we confirm that the utility of suggested method for detection of QPSK signal in the Mobile communications and digital wireless communications.

2 QPSK

2.1 QPSK Signal and Systems

QPSK is the way in which it detects according to the information of the digital signal distinguish 4 phase the phase of the high frequency carrier. QPSK signal must be able to show the four distinguished states of the spectrum's high degree of efficiency. Table 1 shows binary output and 4 output phase.

Table 1. Truth table of QPSK signal

binary output		QPSK phase
Q	I	
1	1	45^O
1	0	135^O
0	0	225^O
0	1	315^O

In Fig. 1, digital signal $S_i(t)$, QPSK signal $S_{QPSK}(t)$ and restored dual data signal $S_0(t)$ are equal to equation (1), (2) and (3) individually.

$$S_i(t) = Bipolar \quad NRZ \tag{1}$$

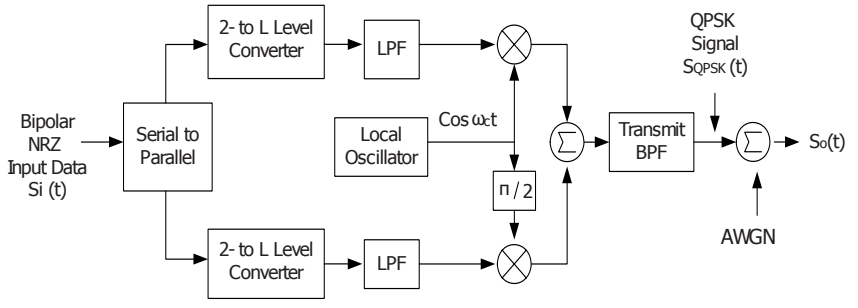
$$S_{QPSK}(t) = b\sqrt{2}\left[a_i \cos \omega_c t + b_i \sin \omega_c t\right] \tag{2}$$

$$S_o(t) = b\sqrt{2}\sum_{i=1}^{\infty} \cos(\omega_c t + \varphi_i) h(t - T_{Si}) + n(t) \tag{3}$$

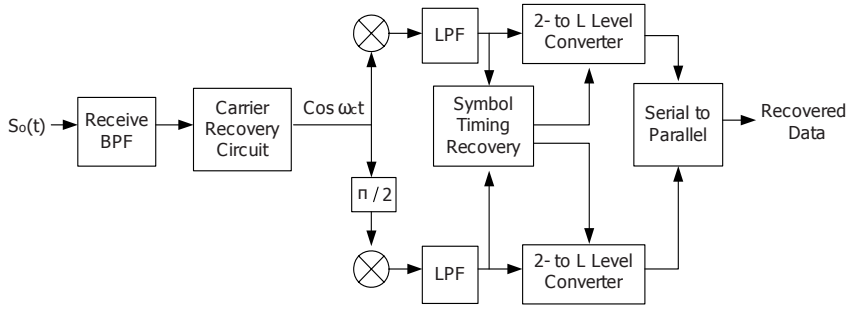
a_i : amplitude of i th $\cos \omega_c t$, b_i : amplitude of i th $\sin \omega_c t$
Ts : symbol period , ω_c : carrier frequency , i : sampling factor
Tsi : i th sampling time , Φi : 45^o, 135^o, 225^o, 315^o , h(t) : impulse response
n(t) : AWGN

As the transmitting channel has limited frequency band-width, nonreturn to zero (NRZ) signal results in the cause of an error by inter symbol interference (ISI)

between transmitting intervals. Therefore, to minimize ISI between NRZ signals, wave must be formed. It is possible to do this only when we can restrict frequency band-width effectively by using raised-cosine filter. This filter is made from determining the roll-off factor α, and the general expression of impulse response h(t) & H(ω) are identical to equation (4), (5). Fig. 2 shows that the impulse response h(t) and transfer function H(ω) of raised-cosine filter by means of value α.



(a) QPSK transmitter



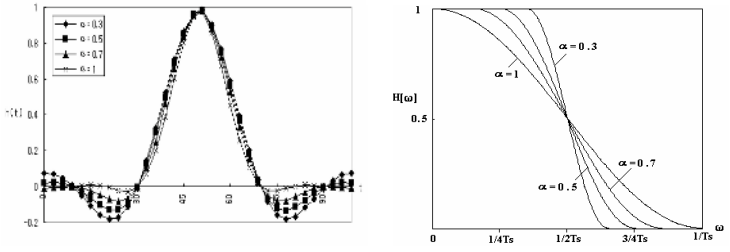
(b) QPSK receiver

Fig. 1. System of QPSK signal

$$h(t) = \frac{\sin \omega_m t}{\omega_m t} \frac{\cos \alpha\omega_m t}{1-(2\alpha\omega_m t/\pi)^2} \tag{4}$$

$$H(\omega) = \begin{cases} \frac{T_S}{2}\left[1+\cos\left(\frac{\pi}{2\alpha}\left(\frac{\omega}{\omega_m}-(1-\alpha)\right)\right)\right] \\ 0 \end{cases} \tag{5}$$

$$\begin{cases} 0 \le |\omega| \le (1-\alpha)\omega_m \\ (1-\alpha)\omega_m \le |\omega| \le (1+\alpha)\omega_m \\ |\omega| > (1+\alpha)\omega_m \end{cases}, \quad \omega_m = \frac{\pi}{T_S}, \quad 0 \le \alpha \le 1$$



(a) h(t) (b) H(ω)

Fig. 2. h(t) & H(ω) of raised-cosine filter

2.2 Error Rates

As the noise that was added during the interval of transmitting signal is AWGN, multi-code cluster noise power N_o is σ^2. The symbol error rates of QPSK signal Ps(e) correspond to equation (6).

$$P_s(e) = erfc\ \sqrt{SNR}\left[1 - \frac{1}{4} erfc\ \sqrt{SNR}\right] \cong erfc\ \sqrt{\frac{E_S}{2N_O}} \tag{6}$$

$$SNR = \frac{E_S}{2N_O} = \frac{b^2}{2\alpha^2} \cdot$$

3 Suggestion of Detection Algorithm Using TDNN

The block diagram of detection algorithm is suggested in Fig. 3 by using the neural network TDNN and the functional major block consists of an intermediate frequency detector, envelope detector, TDNN and decoder.

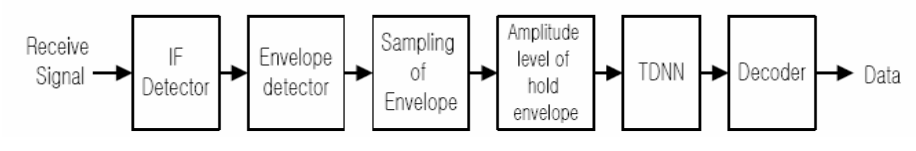


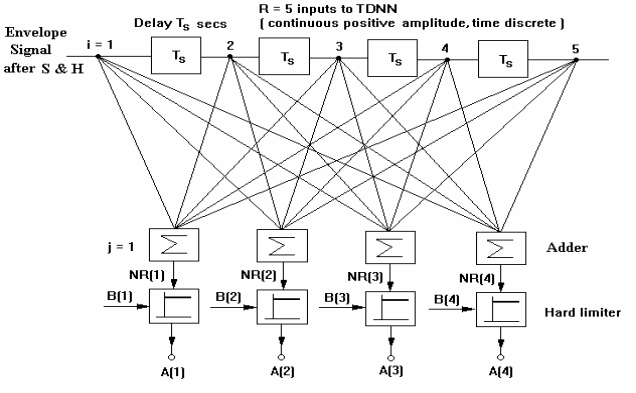
Fig. 3. Diagram of detection algorithm using TDNN

To detect the QPSK signal, received signal is primarily detected with intermediate frequency at the envelope detector. Sample the envelope signal of the pre at the sampler as integer times as the fixed symbol time interval T*s*. Moreover, draw the envelope that holds them and get the amplitude at the symbol period as the input at the TDNN. The TDNN converts the symbol into a bit by the decoder with judging the

correct phase state according to the input amplitude level. The TDNN is a single layer perceptron that consists of 4 neurons to detect the QPSK signal and this is as in Fig. 4.

As the TDNN uses the input data which consists of the data points made with 5 discretely sampled data from envelope of the time domain wave and the TDNN output gets the value of only 1 or 0, each input constructs a net corresponding to them. These samples convert into 4 types of output states from A(1) to A(4), and each output detects one state among the 4 carrier phase states. Neuron 1 outputs the signal which has the phase information of 45°, Neuron 2 outputs the signal which has the phase information of 135°, Neuron 3 outputs the signal which has the phase information of 225°and Neuron 4 outputs the signal which has the phase information of 315°.

The input values that were gathered from the learning are symbols sustaining time T*s*, roll-off factor α and the amplitude level of the data that is sampled from the envelope signal that has the parameters of the phase states of the carriers which are shown as neuron 1, 2, 3, and 4.



S = 4 binary output : A (i) = 1 if j th carrier state detected, or 0 else

Fig. 4. TDNN for detection of QPSK signal

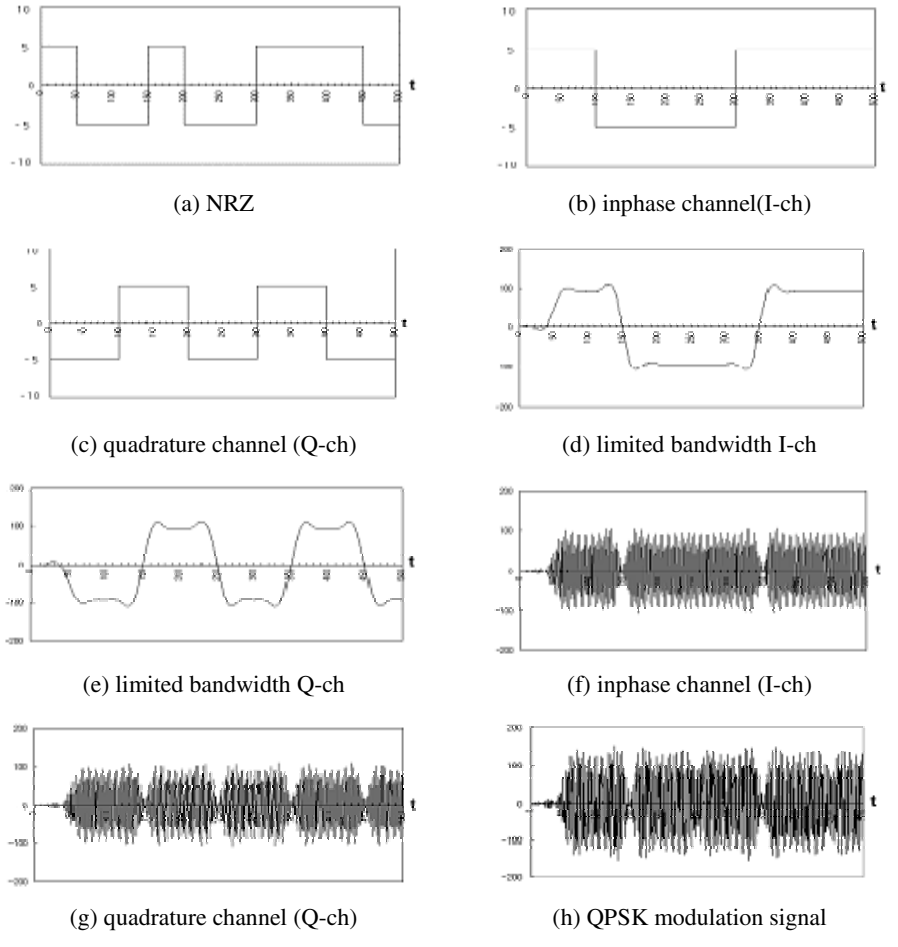
4 Simulation and Analysis

To simulate the modulation of the QPSK signal and the detecting process, the program was made by the language C, and the frequency applied was 6.9㎒.

To pulse shaping of the NRZ signal, the roll-off factor α was set at 0.3, 0.5, 0.7 and 1. Thereafter, I have got the amplitude levels of the values that have phase information, and then input the levels to the TDNN and classified them with 4 phases.

4.1 Modulation of the QPSK Signal

The signal process which went through the QPSK baseband modulation produced the 10 random NRZ signals as in Fig 5.



(a) NRZ (b) inphase channel(I-ch)

(c) quadrature channel (Q-ch) (d) limited bandwidth I-ch

(e) limited bandwidth Q-ch (f) inphase channel (I-ch)

(g) quadrature channel (Q-ch) (h) QPSK modulation signal

Fig. 5. Modulation of QPSK signal

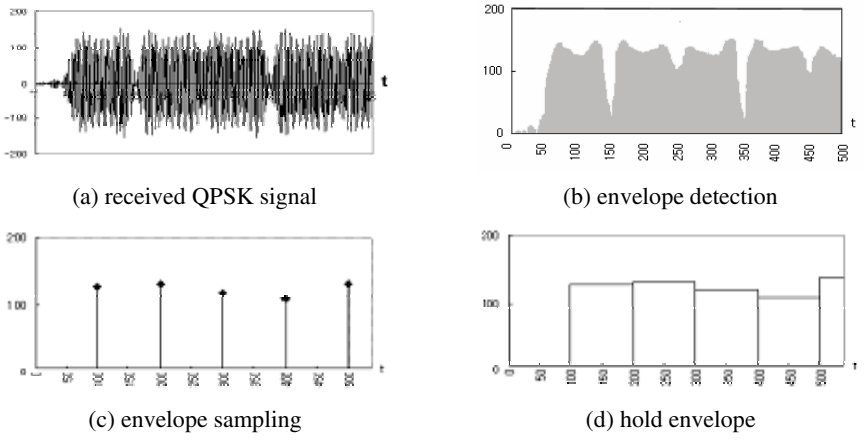
4.2 Detection of the QPSK Signal

The signal process that restores the received QPSK signal with a TDNN detecting algorithm is as in Fig 6.

The amplitude level and the phase of the hold envelope according to the roll-off factor α at the receiving signal management of the QPSK are as in Table 2.

Considering the noise(AWGN), given ± deflection to the amplitude level of the hold envelope of Table 2 and then inputting the value to the TDNN, and, at the TDNN, distinguish the 4 phase of 45°, 135°, 225°and 315°after classifying the phases according to the input and this will determine NRZ signal.

When the α=0.5, the W, B and epochs which are produced by the single layer perceptron learning by using MATLAB are as in Table 2. The inputs are classified by



(a) received QPSK signal (b) envelope detection

(c) envelope sampling (d) hold envelope

Fig. 6. Detection of QPSK signal

Table 2. Amplitude level of hold envelope

symbol \ α	0.3	0.5	0.7	1.0	phase
1	119.62	120.95	122.49	122.25	135。
2	127.5	130.5	132.34	131.85	315。
3	112.68	116.16	119.21	118.77	225。
4	91.71	92.75	94.47	94.15	45。
5	120.3	121.31	122.39	122.28	135。

Table 3. W, B, epochs

MULTI-NEURON CLASSIFICATION for QPSK SIGNAL
TDNN is trained with the perceptron rule to properly classify 5 input vectors into 4 regions. (α= 0.5)
Final Network Values:
W = - 40. 7605 85.3167 -46.9867 47.3025 -41.7936
82.0729 -170.0045 91.8188 -93.3141 83.4340
-242. 1037 -260.0241 -232.0506 -185.1610 -242.9924
0.2020 -0.4857 0.3920 0.2669 0.3997
-164.5462 338.5872 -182.4705 187.2329 -167.0151
B = 1.5251
-1.9478
-1.8922
0.1760
3.6593
Trained for 762 epochs.
Training took 221111 flops.
Average of 290 flops/epoch.
Network classifies : Correctly.

4 phase(45°, 135°, 225°and 315°). With a maximum 1,000 running, classify the accurate phase domain after the 762nd running learning, and the W and bias B at that time will be in Table 3.

4.3 Results Analysis

Fig. 7 shows the results of simulation of the QPSK signal using TDNN, thus they show the symbol error probability according to the SNR when the roll-off factor α of the raised-cosine filter is 0.3, 0.5, 0.7 or 1.

When the SNR was 10 dB, if the α=0.3, 0.5, 0.7 and 1, each symbol error probability was 1.41×10^{-3}, 1.23×10^{-3}, 9.35×10^{-3}, 6.22×10^{-3}, and when the SNR was 15dB, if the α=0.3, 0.5, 0.7 and 1, each symbol error probability was 9.19×10^{-5}, 7.53×10^{-5}, 5.05×10^{-5}, 2.28×10^{-5}.

By the simulation results, we learned that the Ps(e) is approximate to the optimal value at the lower SNR area independent of the roll-off factor α. However, at the higher SNR area, when the roll-off factor α is less than 0.5, by the effect of the ISI, the performance of the system becomes rapidly lower and the symbol error probability gets higher. As we had expected, when the roll-off factor α is higher than 0.5, namely, more than 50% of the excess band-width is applied, especially 100% of the excess band-width(α=1) is applied, the symbol error rates is reduced at the higher SNR area.

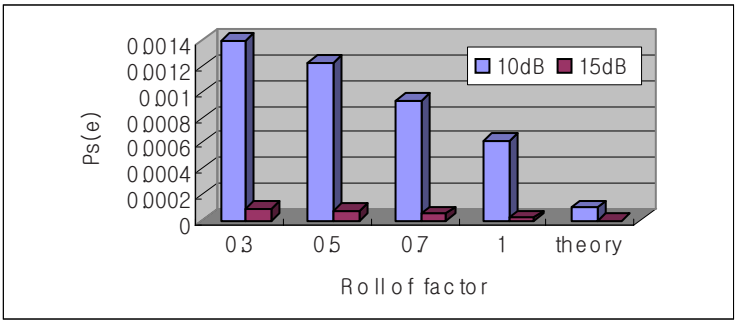


Fig. 7. Symbol error rates of QPSK signal

5 Conclusion

Mobile communications and digital wireless communications have the non-linear characters because of the affect of fading or interference by multiple paths. As multiple users share the same frequency in the mobile communications environment, the spectrum efficiency is getting higher. Moreover, as the effect of the velocity of the mobile object and the terrain surroundings get higher, the digital modulation method is required that the character of linear constant amplitude. As fading, noise and frequency reuse-rates are increased, the receiving signal is highly affected by the same or nearby channels interference of the call quality is rapidly getting worse. Thus, it is a problem that to restore the received signal simply and correctly under this

inferior environments. In this paper, to restore the received signal simply and correctly in the mobile communications and digital wireless communications, we suggested and simulated the algorithm for detection of QPSK signal using TDNN. As the results of simulation, the suggested method was confirmed that the phase information of the QPSK signal was recovered simply and correctly in the mobile communications and digital wireless communications.

References

1. Proakis, J.G.: Digital Communications. New York, McGraw-Hill Book Company (1989)
2. Lee, E.A., Messerschmitt, D.G,: Digital Communication. Norwell, Kluwer Academic Publisher (1992)
3. Diederich, J.: Artificial Neural Networks Concept Learning, IEEE Computer Society Neural Networks Technology Series, IEEE Computer Society (1990)
4. Zurada, J.M.: Introduction to Artificial Neural Systems. St. Paul, West Publishing Company (1992)
5. Kosko, B.: Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence. Englewood Cliffs, Prentice Hall (1992)
6. Lippmann, R.P.: An Introduction to Computing with Neural Nets, IEEE Acoustics, Speech and Signal Processing Magazine 4 (1987) 4-22
7. Farago, A., Lugesi, G.: Strong Universal Consistence of Neural Network Classifiers, IEEE trans. on Information Theory 39(4) (1993) 1146-1151
8. Provence, J. D.: Neural Network Implementation for Maximum-Likelihood Sequence Estimation of Binary Signals in Gaussian Noise, Proceedings of IEEE Int. Conf. on Neural Networks, IEEE (1987) 703-714
9. Hush, D.R, Horne, B.G.: Progress in Supervised Neural Networks; What's New Since Lippmann?, IEEE Signal Processing Magazine 10 (1993) 8-39
10. Bessai, H. J., Loker, K.: Artificial Neural Networks For Smart Detection of Digitally Modula-tion Signals, Proceedings of IEEE Int. Conf. GLOBECOM'04, vol.2 (1994) 1029-1033
11. Hayes, P.V., Uhey, J.R., Sayegh, S. I.: Adaptive FSK Decoding with an Artificial Neural Network, Proceedings of IEEE Int. Tactical Communications Conf., vol.1 (1994) 197-208
12. Rui, J.P., de Figueiredo: Dynamical Functional Artificial Neural Networks(D-FANNS) for Intelligent Signal Processing, IEEE Symposium on Advances in Digital Filtering and Signal Processing, vol.1 (1998) 1-4

The BP Network for Carrier Frequency Offset Estimation in OFDM-Based WLANs

Feng Zhu[1], Yafeng Hu[1], Saiyi Wang[2], and Peng Wei[1]

[1] Dept. of Automation, Tsinghua University, Beijing 100084, China
[2] Dept. of Electronic Engineering, Beijing Institute of Technology, Beijing, China
zhuf03@mails.tsinghua.edu.cn

Abstract. This paper presents an efficient carrier frequency offset (CFO) estimation algorithm for the orthogonal frequency-division multiplexing (OFDM)-based wireless local area networks (WLANs) which adopts the IEEE 802.11 standardization. This algorithm adopts the BP network with the orthogonal projector for the fast adaptive CFO estimation. Numerical results are presented to demonstrate the effectiveness of the proposed algorithm.

1 Introduction

The OFDM-based WLAN system, as specified by the IEEE 802.11a standard [1], uses packet-based transmission. Each packet, as shown in Fig. 1, consists of an OFDM packet preamble, a SIGNAL field, and an OFDM DATA field. The packet preamble is used to determine the channel parameters, including the carrier frequency offset (CFO) and symbol timing.

In [2]-[4], the CFO is estimated based on some repeated OFDM symbols with structures that are different from some of those we consider herein. Recently, a method [5] are particularly tailored to the preamble structure in Fig. 1. However, that method can not estimate CFO iteratively and has the high computation complexity for achieving the accurate result.

In this paper, the BP network is proposed for estimate CFO iteratively and efficiently. This network has a simple structure and only one parameter to be adjusted by the orthogonal projector. Numerical results show the effectiveness of the proposed algorithm with such a BP network.

2 Problem Formation

The packet preamble specified by the IEEE standard (see Fig. 1) consists of 10 identical short OFDM symbols (each containing 16 data samples) and two identical long symbols (each containing 64 data samples). Between the short and long OFDM symbols, there is a guard interval (GI2) of length 32 that constitutes the cyclic prefix of the long symbols.

We consider a frequency-selective fading channel by modeling the channel impulse response as a finite impulse response (FIR) filter with the filter length

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 144–149, 2006.

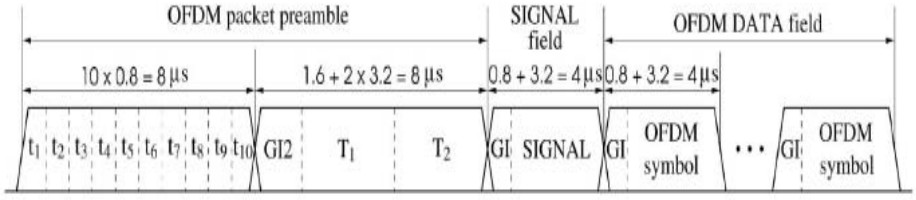


Fig. 1. The packet structure of the IEEE 802.11a standard

assumed to be shorter than a short OFDM symbol. Hence, the first short OFDM symbol is effectively a cyclic prefix for the other nine short OFDM symbols. At an OFDM receiver, apart from a multiplicative complex exponential accounting for the frequency offset, these nine short OFDM symbols will still be identical to each other (note that being affected by the channel-induced intersymbol interference the first OFDM symbol must be discarded). Likewise, due to GI2, the two long OFDM symbols will be identical to each other as well at the receiver. We use the subscripts S and L to differentiate the variables due to the short and long OFDM training symbols, respectively. Let $x_S(m,n)$ ($n = 0, \cdots, N_S - 1$ with $N_S = 16$, and $m = 1, \cdots, M_S - 1$, with $M_S = 10$) denote the nth sample of the mth noise-free short OFDM symbol after removing the cyclic prefix but prior to taking the FFT or estimating and equalizing the channel at the receiver. Note that $x_S(m,n) = x_S(1,n)\exp[j2\pi(m-1)N_S\xi]$ for $m = 1, \cdots, M_S - 1$, with ξ being the CFO. We do not require channel knowledge and treat $x_S(1,n)$ as deterministic unknown. Let $y_S(m,n)$ denote the noisy output at the receiver, i.e., $y_S(m,n) = x_S(m,n) + e_S(m,n)$ where $e_S(m,n)$ is the additive zero-mean Gaussian noise. Let

$$\mathbf{y}_S(n) = [y_S(1,n), \cdots, y_S(M_S - 1, n)]^T \tag{1}$$

and

$$\mathbf{a}_S(\xi) = [1, e^{j2\pi\epsilon}, \cdots, e^{j2\pi(M_S-2)\epsilon}]^T \tag{2}$$

where $(\cdot)^T$ denotes the transpose and $\epsilon = N_S\xi$. Then we have

$$\mathbf{y}_S(n) = \mathbf{a}_S(\xi)x_S(1,n) + \mathbf{e}_S(n) \tag{3}$$

where $\mathbf{e}_S(n)$ is formed from $\{e_S(m,n)\}_{m=1}^{M_S-1}$ in the same way $\mathbf{y}_S(n)$ is formed from $\{y_S(m,n)\}_{m=1}^{M_S-1}$. Similar expressions can be readily obtained for the two long OFDM symbols

$$\mathbf{y}_L(n) = \mathbf{a}_L(\xi)x_L(1,n) + \mathbf{e}_L(n), n = 0, \cdots, N_L - 1 \tag{4}$$

where

$$\mathbf{y}_L(n) = [y_L(0,n), \cdots, y_L(M_L - 1, n)]^T \tag{5}$$

and

$$\mathbf{a}_L(\xi) = [1, \cdots, e^{j2\pi(M_L-1)\mu}]^T \tag{6}$$

with $\mu = N_L\xi$, $M_L = 2$ and $N_L = 64$.

Our problem of interest herein is to estimate the CFO from $\mathbf{y}_S(n)$ and $\mathbf{y}_L(n)$ without knowing the training symbol values or the underlying FIR channel.

3 BP Network with Orthogonal Projector

Taking the short symbols as an example (which could just as well be fit for the long symbols), all of the nine short symbols can be used minimizing the following nonlinear squares (NLS) cost function

$$C(\xi, \{x_S(1,n)\}_{n=0}^{N_S-1}) = \sum_{n=0}^{N_S-1} \|\mathbf{y}_S(n) - \mathbf{a}_S(\xi)x_S(1,n)\|^2 \tag{7}$$

where $\|\cdot\|^2$ denotes the Euclidean norm. Note that the ξ is deterministic unknown, the minimization of (7) is equal to the minimization of every term

$$C(n) = \|\mathbf{y}_S(n) - \mathbf{a}_S(\xi)x_S(1,n)\|^2, n = 0, \cdots, N_S - 1. \tag{8}$$

Minimizing the right side of (8) with respect to $x_S(1,n)$ gives

$$\widehat{x}_S(1,n) = (\mathbf{a}_S^H(\xi)\mathbf{a}_S(\xi))^{-1}\mathbf{a}_S^H(\xi)\mathbf{y}_S(n). \tag{9}$$

Using the (2), (9) is equal to

$$\widehat{x}_S(1,n) = \frac{1}{M_S - 1}\mathbf{a}_S^H(\xi)\mathbf{y}_S(n). \tag{10}$$

Taking (10) into (8), (8) can be written as

$$C(n) = \|\mathbf{y}_S(n) - \widehat{\mathbf{y}}_S(n)\|^2 = \left\|P^{\perp}_{\mathbf{a}_S(\xi)}\mathbf{y}_S(n)\right\|^2 \tag{11}$$

where $\widehat{\mathbf{y}}_S(n) = \mathbf{a}_S(\xi)\widehat{x}_S(1,n)$ and $P^{\perp}_{\mathbf{a}_S(\xi)}$ stands for the orthogonal projector onto the null space $\mathbf{a}_S^H(\xi)$, i.e.

$$\begin{aligned} P^{\perp}_{\mathbf{a}_S(\xi)} &= I - \mathbf{a}_S(\xi)(\mathbf{a}_S^H(\xi)\mathbf{a}_S(\xi))^{-1}\mathbf{a}_S^H(\xi) \\ &= I - \mathbf{a}_S(\xi)\mathbf{a}_S^H(\xi) \end{aligned} \tag{12}$$

with I denoting the identity matrix.

Based on the above discussion, noting the (11) and (12), the BP network is adopted and shown in Fig. 2. In this network, only the parameter ξ are adjusted according to the C in (7) which is also the back propagated error, whereas the connective weights are kept invariant. Then our aim is to train the network and search the parameter ξ by the input signal $\mathbf{y}_S(n)$, $n = 0, \cdots, N_S - 1$.

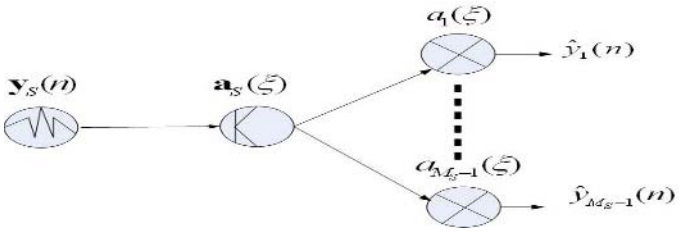


Fig. 2. The structure of BP network.The input $\mathbf{y}_S(n)$ firstly makes inner product with $\mathbf{a}_S(\xi)$. The inner product $\mathbf{a}_S^H(\xi)\mathbf{y}_S(n)$, after multiplying the connective weights which are all the same value $\frac{1}{M_S-1}$, is put into the M_S-1 nodes for multiplying the corresponding factor $a_m(\xi)$, $m=1,\cdots,M_S-1$. The output, $\widehat{\mathbf{y}}_S(n)=[\widehat{y}_1(n),\cdots,\widehat{y}_{M_S-1}(n)]^T$, is the estimate to $\mathbf{y}_S(n)$.

For convenience, we search parameter $\epsilon = N_S\xi$ instead of ξ. According to the delta rule, the adjustment of ϵ is

$$\delta\epsilon = -\mu\frac{\partial C}{\partial \epsilon} = -\mu\sum_{n=0}^{N_S-1}\frac{\partial C(n)}{\partial \epsilon} \tag{13}$$

where μ is the learning rate. Using the (12), (11) can be simplified as

$$\begin{aligned} C(n) &= \mathbf{y}_S^H(n)P^{\perp}_{\mathbf{a}_S(\xi)}\mathbf{y}_S(n) \\ &= \mathbf{y}_S^H(n)(I-\mathbf{a}_S(\xi)\mathbf{a}_S^H(\xi))\mathbf{y}_S(n). \end{aligned} \tag{14}$$

Let

$$\omega = \exp(j2\pi\epsilon) \tag{15}$$

and using the (1) with (2), the (14) can be written as

$$\begin{aligned} C(n) = -f(\omega) = -[y_1^*(n), y_2^*(n), \cdots, y_{M_S-1}^*(n)] \times \\ \begin{pmatrix} 0 & \omega^{-1} & \cdots & \omega^{-M_S+2} \\ \omega & 0 & & \omega^{-M_S+3} \\ \vdots & & & \vdots \\ \omega^{M_S-2} & \omega^{M_S-3} & \cdots & 0 \end{pmatrix} \begin{pmatrix} y_1(n) \\ y_2(n) \\ \vdots \\ y_{M_S-1}(n) \end{pmatrix}. \end{aligned} \tag{16}$$

From the above equation, we can know $f(\omega)$ is the polynomial of ω which can be expressed as

$$f(\omega) = \sum_{k=1}^{M_S-2}(c_{-k}\omega^{-k} + c_k\omega^k) \tag{17}$$

where

$$c_{-k} = \sum_{i=1}^{M_S-1-k} y_{k+1}^*(n) y_i(n) \tag{18}$$
$$c_k = \sum_{i=1}^{M_S-1-k} y_{k+1}(n) y_i^*(n).$$

It's obvious that $c_{-k} = c_k^*$, which can reduce the computation complexity largely. Differentiate the (17), we have

$$\frac{\partial f(\omega)}{\partial \omega} = \sum_{k=1}^{M_S-2} (-k c_{-k} \omega^{-k-1} + k c_k \omega^{k-1}). \tag{19}$$

Noting

$$\frac{\partial C(n)}{\partial \epsilon} = -\frac{\partial f(\omega)}{\partial \omega} \frac{\partial \omega}{\partial \epsilon}, \tag{20}$$

the adjustment $\delta\epsilon$ in (13) is achieved with (15) and (19).

For searching the CFO ξ, Which is proportion to the ϵ, the short symbols are sent to the BP network periodically until the nonlinear squares (NLS) cost function C in (7) is less than a chosen threshold. It should be emphasized that the learning rate μ in (13) is time variant, i.e

$$\mu(t) = \frac{\mu_0}{t}. \tag{21}$$

The initialization of ϵ can be set according to the experience from the channel.

4 Experiment Result

In this example, the algorithm performance for a frequency-selective fading channel is evaluated by assuming that the packet timing is available. We set $\xi = 0.08/N_S$ and the channel impulse function is chosen as $h = [e^{j2.5}, e^{j0.2}, e^{-j1.5}]$. The mean-squared errors (MSEs) of the CFO estimates are obtained from 100 Monte Carlo trials and are compared with the corresponding Cramér–Rao bounds (CRBs) [3] as a function of SNR, which is defined as the ratio of the average energy per sample to the variance of the additive white Gaussian noise.

Fig.3 shows the MSEs and CRBs of the CFO estimates versus SNR when using only nine short, both nine short and two long OFDM symbols. Observe from Fig.3 that using both the nine short and two long OFDM symbols yields an extra improvement with respect to using nine short symbols. In both the two conditions, the MSE converges to the corresponding CRB with the increasement of SNR.

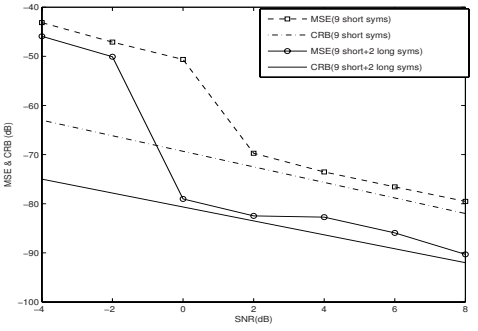


Fig. 3. Comparison of MSE with CRBs as a function of SNR for the OFDM frequency offset estimates with a frequency-selective fading channel

5 Conclusion

In this paper, the efficient algorithm using the BP network for estimate carrier frequency offset in OFDM-Based WLANs is presented. This network has a simple structure and only one parameter to be adjusted by the orthogonal projector. Numerical results for a frequency-selective fading channel show the effectiveness of the proposed algorithm with such a BP network.

Acknowledgement

This work was supported by the Natural Science Foundation of China under Grant 60375004. Feng Zhu and Yafeng Hu are the first coauthors.

References

1. Van, R., Awater, G., Morikura, M., Takanashi, H., Webster, M., Halford, K.: New High-Rate Wireless LAN Standards. IEEE Comm. Magazine **37**(12) (1999) 82-88
2. Moose, P. H.: A Technique for Orthogonal Frequency Division Multiplexing Frequency Offset Correction. IEEE Trans. Comm. **42**(10) (1994) 2908-2914
3. Schmidl, T. M., Cox, D. C.: Robust Frequency and Timing Synchronization for OFDM. IEEE Trans. Comm. **45**(12) (1997) 1613- 1621
4. Negi, R., Coiffi, J.: Pilot Tone Selection for Channel Estimation in a Mobile OFDM System. IEEE Trans. Consumer Electronics **44**(8) (1998) 1122-1128
5. Li, J., Liu, G., Giannakis, G.B.: Carrier Frequency Offset Estimation for OFDM Based WLANs. IEEE Signal Processing Letters **8**(3) (2001) 80-82

The LD-CELP Gain Filter Based on BP Neural Network

Gang Zhang, Keming Xie, Zhefeng Zhao, and Chunyu Xue

College of Information Engineering, Taiyuan University of Technology, Taiyuan 030024, Shanxi, China
tyzhgang@tom.com

Abstract. The recommendation G.728 depends on the Levinson-Durbin (L-D) algorithm to update its gain filter coefficients. In this paper, it is contrasted with BP neural network method. Because quantizer has not existed at optimizing gain filter, the quantization SNR can not be used to evaluate its performance. This paper proposes a scheme to estimate SNR so that gain predictor can be separately optimized with quantizer. Using BP neural network filter, the calculation quantity is only 6.7 percent of L-D method's and its average segment SNR is about 0.156dB higher than G.728. It is also used to evaluate the case that excitation vector is 16 or 20 samples, respectively, the BP neural network algorithm has similarly good result.

1 Introduction

At present, the structure of the gain-shape product codebook has been used to almost all CELP speech coding algorithm. It is very important for LD-CELP how to get a perfect performance of the excitation gain quantizer. After contrasting the fix-coefficient Jayant gain filter [1], G.728 adopts a 10-th order LPC gain filter which coefficients are updated with L-D algorithm [2, 3]. But the literature [2] introduces nothing on comparing their performance. A reason may be because quantizer has not existed at optimizing gain filter, there is nothing used to evaluate its performance. This paper proposes a method to estimate SNR so that gain predictor can be separately optimized with the quantizer. We use BP neural network to replace G.728's LPC filter.The test shows that BP filter's computing quantity is only 6.7 percent of G.728 L-D method's and its average segment SNR is about 0.156dB higher than the later.

2 SNR Estimating

In CELP, searching codebook according to formula

$$D_{min} = \|\boldsymbol{x}(n) - G_j(n)\boldsymbol{H}(n)\boldsymbol{y}_j(n)\|^2 \tag{1}$$

where, $\boldsymbol{x}(n)$ is the target vector, $\boldsymbol{H}(n)$ is the unit impulse respond matrix of the short-term predictor, $G_i(n)$ and $\boldsymbol{y}_j(n)$ are gain and shape codeword, respectively. To be easily understood, let $\boldsymbol{y}_j(n)$ be normalized unit power. The

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 150–155, 2006.

quantization value and the predictor output of excitation gain are denoted as $\hat{G}_i(n)$ and $\sigma(n)$, respectively. In logarithm domain the predictor output is $\log_2 \sigma(n) = \sum_{k=1}^{p} p_k \log_2 \hat{G}_i(n-k)$ and its predicting error $\log_2 g_i(n)$ is

$$\log_2 g_i(n) = \log_2 G_i(n) - \log_2 \sigma(n) \tag{2}$$

or $G_i(n) = \sigma(n) g_i(n)$. Thus we can write (1) as

$$D_{\min} = \sigma^2(n) \left\| \hat{\boldsymbol{x}}(n) - g_i(n) \boldsymbol{H}(n) \boldsymbol{y}_j(n) \right\|^2 \tag{3}$$

where $\hat{\boldsymbol{x}}(n)$ is the target vector adjusted by $\sigma(n)$. Ulteriorly, the minimum of above formula equals the maximum follow

$$\hat{D}_{\max} = 2 g_i(n) \boldsymbol{P}^T(n) \boldsymbol{y}_j(n) - g_i^2(n) E_j(n) \tag{4}$$

In (4),$\boldsymbol{P}(n) = \boldsymbol{H}^T(n) \hat{\boldsymbol{x}}(n)$, $E_j(n) = \left\| \boldsymbol{H}(n) \boldsymbol{y}_j(n) \right\|^2$. Let $\partial \hat{D}_{max} / \partial g_j = 0$, we get the exact value of the excitation gain

$$g_j(n) = [\boldsymbol{P}^T(n) \boldsymbol{y}_j(n)] / E_j(n) \tag{5}$$

Let $Q\{\bullet\}$denote quantization of signal$\{\bullet\}$, then

$$\log_2 \hat{G}_j(n) = Q\{\log_2 g_j(n)\} + \log_2 \sigma(n) \tag{6}$$

As selecting predictor, the signal $\hat{G}_j(n)$ has not existed yet, which had to be replaced with $G_j(n)$, so the SNR will be in optimum state and then nothing is used to judge whether the gain predictor is suitable or not. But it can be estimated [4]. From prediction residue error $\varepsilon_j(n) = G_j(n) - \sigma(n)$ can know

$$g_j(n) = G_j(n) / [G_j(n) - \varepsilon_j(n)] = snr_j(n) / [snr_j(n) - 1] \tag{7}$$

where $snr_j(n) = G_j(n) / \varepsilon_j(n)$ is the ratio of the signal $G_j(n)$ to $\varepsilon_j(n)$ at time n, ulteriorly we have

$$snr_j(n) = g_j(n) / [g_j(n) - 1] \tag{8}$$

From (8) can see that the more $g_j(n)$ is close to 1, the bigger $snr_j(n)$ devotes to SNR. Let snr be an observing value of $snr_j(n)$ set up in advance and Δ is its level in dB, we can find out the all j of satisfying with $snr_j(n) > snr$. It is $SNR = 10 \lg N^{-1} \sum_{k=1}^{N} snr_j^2(k) \geq 20 \lg |snr| = \Delta$, or

$$10^{\Delta/20} = |snr| \leq (1 - g_j(n)^{-1})^{-1}$$

If $g_j(n) > 1$ then $g_j(n) < (1 - 10^{-\Delta/20})^{-1}$ or at $g_j(n) < 1$ have $g_j(n) > (1 + 10^{-\Delta/20})^{-1}$. Now we consider the signal $\sigma(n)$ got from different predictor, respectively. Selecting a SNR observing level Δ , where the quantization signal $\hat{G}_j(n)$ is replaced by $G_j(n)$, the corresponding range is looked into

$$(1 + 10^{-\Delta/20})^{-1} \leq g_j(n) \leq (1 - 10^{-\Delta/20})^{-1} \tag{9}$$

For different schemes, the percentages of $g_j(n)$ satisfied (9) are respectively calculated. The one with bigger percentage is accepted as better than others.

3 Gain Quantization (GQ)

3.1 Principles

Fig.1 is the GQ block scheme [4, 5]. We sends the 3-bit index $I(n)$ got by quantizing $\log_2 g_j(n)$ to decoder while decodes $I(n)$ to get the local quantized signal $\log_2 \hat{g}_j(n)$. Adding the gain estimation $\log_2 \sigma(n)$ can obtain the local rebuild signal $\log_2 \hat{G}_j(n)$. This signal is input to a gain predictor in order to product the gain estimation $\log_2 \sigma(n+1)$ which is subtracted from the optimum gain value $\log_2 G_i(n+1)$ to have next prediction residual error $\log_2 g_j(n+1)$.

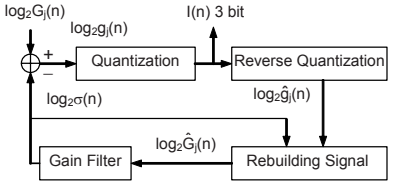


Fig. 1. Gain quantizer block scheme

3.2 The Quantizer Optimum Parameter

By selecting a set of the optimum parameters can get the best quantization result of $\log_2 g_j(n)$. The distribution $p(x)$ of $\log_2 g_j(n)$ is shown in Fig.2. Ignoring sign bit, $\log_2 \hat{g}_j(n)$ is only one of four discrete values $\eta_i (i = 0, \ldots, 3)$. The variance of quantization residue error is

$$\sigma_e^2 = \sum_{i=0}^{3} E\left[(\eta_i - \log_2 g_j(n))^2\right] = \sum_{i=0}^{3} \int_{\xi_i}^{\xi_{i+1}} (\eta_i - x)^2 p(x) dx \tag{10}$$

Selecting the parameters ξ_i and η_i minimizes σ_e^2 , here $\xi_0 = -\infty, \xi_4 = +\infty$. Let $\partial\sigma_e^2/\partial\xi_i = \partial\sigma_e^2/\partial\eta_i = 0$ get

$$\eta_i = [\int_{\xi_i}^{\xi_{i+1}} x p(x) dx] / [\int_{\xi_i}^{\xi_{i+1}} p(x) dx] \tag{11}$$

$$\xi_i = (\eta_{i-1} + \eta_i)/2 \tag{12}$$

Formula (11) and (12) show respectively that the η_i's optimum position locates in rectangle center between ξ_i and ξ_{i+1}, the optimum ξ_i is the average of η_{i-1} and η_i. The parameters ξ_i and $\eta_i (i = 0, \ldots, 3)$ can be iteratively found out as follows [4, 5]. Let Δ be the step size of dividing range $[\xi_i, \xi_{i+1}]$. For each Δ, the number f_i of samples $\log_2 g_j(n)$ is counted. Thus we can get the frequency f_i/F (F is total number of them) which is approximately used as $p(x)$ in (11). Let x be the middle value of the step size Δ. If Δ is enough small, we can think the signal $\log_2 g_j(n)$ as uniform distributing. After $\{\xi_i, \eta_i\}$ put initial value we can get the $p(x)$. New $\{\xi_i, \eta_i\}$ are calculated with (11) and (12) iteratively until their values are steady.

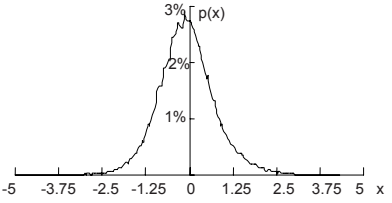


Fig. 2. The distribution $p(x)$ of $\log_2 g_j(n)$

3.3 The BP Neural Network

The data preprocessed with linearity normal scheme are input to a BP neural network. Its framework is denoted as I-H-O which indicates the input nodes I, the hidden nodes H and the output nodes O. The BP network 5-4-7-1 denotes that its structure is 5 input nodes, 4 first hidden layer nodes, 7 second hidden layer nodes and 1 output node. To make up its structure, we use firstly one hidden layer which number of nodes should be as few as possible, and then increase by one node time and again until the effect is satisfied or increase another hidden layer. As adopting inequation (9) to evaluate gain filter performance, 0.467372dB is the result of G.728's L-D arithmetic at $\Delta = 10$dB. In Tab.1, several well BP network with fix coefficient are listed out.

Table 1. The BP neural network

Weight	$\Delta = 10$dB	$\Delta = 20$dB	$\Delta = 40$dB
10-5-1	0.461361	0.147223	0.0147601
9-3-1	0.488967	0.161131	0.0161372
4-10-1	0.497861	0.167168	0.0169031

4 Speech Coding Test

For 5 samples of excitation gain, the quantization intervals and the quantization levels of the BP network are listed in the Tab.2. Then we execute a evaluate test with 15 minutes signals about 7 millions of speech samples. The test result sees Tab.3. The BP network filter's speech coding performance is about 0.156dB

Table 2. The BP network 4-10-1 quantizer

$\lvert\log_2 g_j(n)\rvert$	$\lvert I(n)\rvert$	Quantizer output
$[2.19897, +\infty)$	3	2.86077
$[1.0263, 2.19897)$	2	1.53716
$[-0.050015, 1.0263)$	1	0.51543
$(-\infty, -0.050015)$	0	-0.4154

Table 3. Speech coding effect

Gain Filter	Average Segment SNR
G.728's L-D	18.4506
4-10-1 BP Network	18.6065

higher than G.728's SNR, which is absolutely accordant with the inequation (9) to evaluate filter.

With respect to the computing complexity, G.728 L-D method is 10 divisions and 1810 additions or multiplications. Without window computing, the 4-10-1 structure of the BP network with fix coefficient takes only 122 additions or multiplications without division which is about 6.7 % of G.728 L-D's.

5 16 or 20 Dimension Vector

Researching the case of 16 or 20 samples per excitation vector is very important to reduce rate. For 16 and 20 dimension vector, we use (9) to evaluate gain filter performance. Tab.4 is the result of G.728's L-D method. Tab.5 and Tab.6 are these of BP network filter, respectively. From them, we can see that the performance of BP network filter is also the best one in this case. According to the section 4, we believe reasonably that speech coding effect of BP network filter is also excelled than L-D method for 16 or 20 dimension vector.

Table 4. The L-D method evaluate results

L-D	$\Delta = 10$dB	$\Delta = 20$dB	$\Delta = 40$dB
16Dim	0.58858	0.210464	0.021398
20Dim	0.612049	0.225571	0.023004

Table 5. The evaluate result 16 Dim. vector

BP filter	$\Delta = 10$dB	$\Delta = 20$dB	$\Delta = 40$dB
3-4-1	0.626763	0.224912	0.0227424
3-8-1	0.626170	0.225001	0.0167740
3-7-5-1	0.631193	0.226982	0.0231033

Table 6. The evaluate result 20 Dim. vector

BP filter	$\Delta = 10$dB	$\Delta = 20$dB	$\Delta = 40$dB
2-8-1	0.647425	0.241631	0.0246328
2-9-1	0.647417	0.241656	0.0246716
2-5-10-1	0.644981	0.241927	0.0247044

6 Results and Conclusions

According to the method of judging excitation gain filter performance proposed in this paper, the effects of the BP network filter are all surpassing G.728 L-D method for whichever of 5, 16 or 20 dimension excitation vector. This Conclusion is still existence when it is used in speech coding of 5 samples excitation vector. The lowest complexity of the fix coefficient BP network with 4-10-1 structure is only 6.7 percent of computing quantity of L-D method and its average segment SNR is about 0.156dB higher than G.728.

Acknowledgment

Project Supported by National Natural Science Foundation of China (NSFC) (60372058, 60374029) and Shanxi Province Nature Science Fund (20041046).

References

1. Jayant, N.S.: Adaptive Quantization with a One-word Memory. Bell Syst. Tech. J. 52 (1973) 1119–1144
2. Chen, J.H.: A Robust Low-Delay CELP Speech Coder at 16 kb/s. Advances in Speech Coding. Kluwer Academic Publishers, Boston (1991) 25–35
3. CCITT: Recommendation G.728. Coding of Speech at 16kbit/s Using Low-delay Code Excited Linear Prediction. Geneva (1992)
4. Xue, C.Y.: The Optimization and Choosing of the Gain Filter of the LD-CELP Algorithm. Taiyuan University of Technology Master's Dissertation. (2005)
5. Zhang, G., Xie, K.M., Zhang, X.Y., Huangfu, L.Y.: Optimizing Gain Codebook of LD-CELP. Proc. IEEE ICASSP, Vol. 2. Hong Kong (2003) 149–152

A Neural Network Based Application Layer Multicast Routing Protocol

Peng Cheng, Qiong-Hai Dai, and Qiu-Feng Wu

Department of Automation, Tsinghua University,
Beijing, 100084, P.R. China
chengpeng00@mails.tsinghua.edu.cn

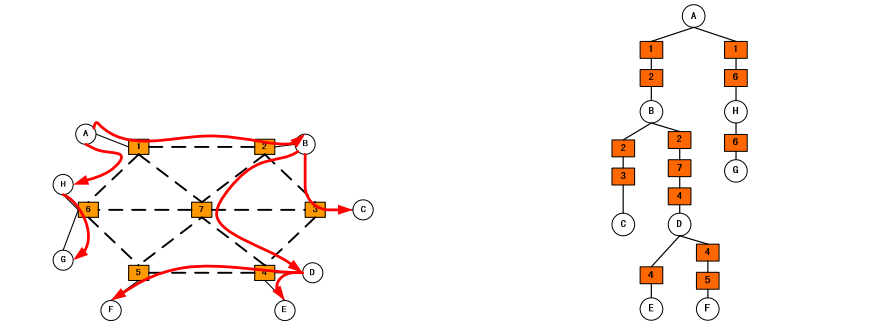
Abstract. Application layer multicast routing problem is a multi-objective optimization problem. This paper reviews the multicast routing problem, analyzes the multi-constraints, especially the importance of network layer load balance. Then a modified SOFM approach is proposed to build a multicast tree according to cost, application layer load balance and network layer load balance. Numerical simulations show that compared with existent routing algorithms, the proposed algorithm improved the performance on all the three metrics.

1 Introduction

Multicast is necessary to satisfy the demand of people for more and more Internet applications, especially multimedia. One host can send the same data to a large number of receivers at the exact same time, which leads to more efficient use of system resources and reduces the bandwidth consumption. The early multicast protocols need the support of network layer equipments and are called IP Multicast (IPM). Though IPM can be an efficient solution, it has not been extensively deployed due to the complexity and the lack of network layer support for multicast in the Internet. Application Layer Multicast (ALM) was then proposed [1] [2]. ALM systems are built on top of a general Internet unicast infrastructure. In ALM systems, hosts are organized, or self-organize, into two logical connected topologies. One topology is called the mesh and the other is called the tree. Each host periodically communicates with its mesh neighbors to maintain the topologys connectivity. One tree or one set of trees are built embedded in the mesh by a routing algorithm. Multicast data is transferred along the branches of the tree.

Fig. 1(a) describes a typical multicast system in which there are 8 hosts, host A, host B, ..., host H. The network layer topology includes 7 routers: router 1, router 2, ..., router 7. In this case, the multicast source is host A and the other seven hosts are receivers. The hosts set composed of all receivers is called multicast group. The multicast data sent from A arrives first arrive at B and H along unicast paths. B and H then replicate the data and transfer them to C, D and G respectively. D also act in the same way as B and H and transfer the data to E and F. The path of data can be summarized as a tree in Fig. 1(b). All

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 156–164, 2006.



(a) Application Layer Multicast Group (b) Application Layer Multicast Tree

□ : *routers*; ◯ : *hosts*; →: *datapath*

Fig. 1. Application Layer Multicast Topologies

the nodes in the tree are hosts and the tree is composed of usual unicast paths. In this tree the source, A, acts as the root.

The multicast trees, whether IPM trees or ALM trees, have similar shape. The difference between two kinds of trees is that the nodes in IPM trees are network routers but those in ALM tress are hosts. Routing algorithms have been studied a lot in IPM protocols. In one of the earliest papers to address the issue of routing to multiple destinations [3], tow cost terms are defined: network cost and destination cost. The former one, which is simply the sum of the costs of all the edges in the tree, is a measure of the utilization of network resources. The later one, which is the sum of the costs of the paths between the source and each destination, is a measure of the average delay experienced by each destination. the Minimum Spanning Tree Heuristic (MSTH) [4] and the Average Distance Heuristic (ADH) [5] are proposed to build static IPM trees. With a view to the issue of Quality of Service, especially for multimedia applications, the concept of Constrained Spanning tree is proposed in [6].Separate functions are used for the cost and the delay as opposed to [3] where the cost and delay functions are the same. The algorithm tries to find the minimization of tree cost given a bounded delay, rather than minimization of the average path delay which is the goal of [3]. A modification of the MSTH was found to have the best performance among the heuristics discussed. The construction of multicast trees that ensure a certain amount of nodal load balancing is discussed in [7]. The MSTH and ADH are modified to integrate link capacities for routing of multimedia multicast streams.

Similar to IPM trees, the performance of ALM trees can be described by cost, delay and load balancing. The common cost and delay constraints include the trees total cost constraint, the trees diameter constraint, the constraint of average delay experienced by all destinations [8], and so on. The load balancing constraints purpose is to present part of hosts and network link from failing by too much load. In [8], MDDL and LDRB are used to describe two different routing algorithms. The focus of MDDL is the optimization of

trees diameter, while that of LDRB is load balancing performance. In [9], network layer load balancing is studied as well as cost and node load balancing. Many popular overlay multicast solutions use some simple geographical rules, such as Compass Routing (CR) [1][10] and Bidirectional Minimal Spanning Tree (BMST) [2], to build a acceptable tree instead of trying to find an optimized tree.

Heuristics based on Neural Networks (NN) have been applied to solve the routing problem in communication networks for a long period. [11] and [12] combine NN with Genetic Algorithms. [13] designs a modified Self Organizing Feature Map (SOFM) algorithm to build the tree and an extra five-layer neural network for new members joining the session.[14] uses random neural network to improve the routing performance. [15] proposes a modified Hopfield neural network to solve delay constrained multicast routing problem. [16] uses a SOFM network for mobile hosts clustering and a Hopfield network to build a tree with a minimum number of links.

In this paper, we present a routing solution for ALM with cost, application layer load balancing and network layer load balancing constraints. The neural network used in this paper is similar to the one in [13] though more constraints are made and the competing process is modified according to routing constrains. The other parts of this paper is organized as following: section II analyzes the ALM system model and routing problem; section III discusses the proposed algorithm in details; the numerical simulation is proposed in section IV. Section V gives out the conclusion and our future plan.

2 System Model

2.1 Network Model

The hosts in ALM group self-organize into a connected mesh which can be summarized as a directed graph $M(H, P, c)$. The mesh M consists of a nonempty set H of $|H|$ hosts and a set P, $P \subseteq H \times H$, of $|P|$ unicast paths connecting pairs of hosts. Each unicast path is associated with a cost function, $c : P \rightarrow R^+$. These paths are element paths that are candidates to build the multicast tree. $H = s \cup D$, s acts as the data source of the group and other hosts belonging to D are the receivers. The mesh is a complete graph when all pairs of hosts are connected by element paths.

With a view to the network layer topology, the overlay multicast group can be summarized into a directed graph, $G(H, N, L, E, c)$. The set N consists of $|N|$ routers and the set E consists of edges connecting pairs of routers, $E \subseteq N \times N$, $E = e_i | i \in N, i \leq |E|$. The set L consists of the links connecting hosts and routers, $|L| = |H|$, $L = l_j | j \in N, j \leq |L|$. All edges and links are associated with the cost function c, $c : E \rightarrow R^+$, $c : L \rightarrow R^+$.

An element path in M can be written as a link-edges-link sequence. The element paths cost is equal to the sum of the links' and the edges cost in the sequence.

2.2 ALM Routing Problem

Given mesh M, the objective of ALM routing algorithm is to find a spanning tree $T(H, F)$, $F \subseteq P$, $T \subseteq M$, subjected to the following constraints.

Cost constraint: a trees cost is the sum of cost of all branches used in the tree. The objective of cost optimization routing is to find a spanning tree $T(H, F)$ of given mesh M, whose cost should be the minimum among all of Ms spanning trees.

Host Load Balance constraint: for a host, the number of its children in the overlay multicast tree should be less than an upper bound, otherwise it may become the bottleneck. For host h, use $d_{max}(h)$ and $d_T(h)$ to denote the degree bound and its actual degree in tree respectively. Define the variable α as,

$$\alpha_T(h) = \frac{d_T(h)}{d_{max}(h)}.$$

The routing problem with Load Balance constraint can be summarized as: find a spanning tree $T(H, F)$ of given mesh M, subject to

- $\alpha_T(h) \leq 1, \forall h \in H$;
- the value of $max_h \alpha_T(h)$ in T is the maximum among all of M's spanning trees.

The first constraint prevents any host from collapse caused by more children than its extreme ability. The second constraint tries to make hosts' interface bandwidth as efficient as possible.

Network Stress constraint: The stress constraint tries to limit the maximum times for one edge to transit the same data. Be analogous to the balance constraint, define $s_{max}(e)$ and $s_T(e)$ as the stress upper bound and the actual stress in tree T of the edge e. The routing problem under balance constraint is: find a spanning tree $T(H, F)$ of given mesh M, subject to

- $\beta_T(e) \leq 1, \forall e \in T$, where $\beta_T(e) = \frac{s_T(e)}{s_{max}(e)}$
- the value of $max_e \beta_T(e)$ in T is the maximum among all of M's spanning trees.

3 Detailed Algorithm

3.1 Output Neuron Space and Input Signals

Denote the multicast tree at time t as $T_t(H_t, F_t)$. In T_t, the parent of host h_i is denoted as $A_t(h_i)$, the children hosts set of host h_i is denoted as $C_t(h_i)$. Output neuron space includes N neurons, denoted as $n_j, j = 1, 2, \ldots, N$. These neurons are also organized to a tree denoted as $\widetilde{T}_t$. In $\widetilde{T}_t$, the parent neuron of n_j is denoted as $\widetilde{A}_t(n_j)$, the children hosts set of host n_j is denoted as $\widetilde{C}_t(n_j)$.

If and only if a neuron is associated with a host in T_t, it is called a Type I neuron. The connection between a pair of Type I neurons in $\widetilde{T}_t$ represents the path between their responding hosts in T_t.

Every neuron has a $|D|$ dimensional weight vector associated with itself. For neuron n_j, the weight vector is denoted as $w_j = [w_{j1}, w_{j2}, \ldots, w_{j|D|}]$. The input signal to the neural network is represented by a series of $|D|$ dimensional vectors. Since the child of a host must be one of its mesh neighbors, the i-th input signal is denoted as x_i, where

$$x_{i,k} = \begin{cases} c(p(i,k)), & \text{host i and host k are mesh neighbours;} \\ 0, & \text{i=k;} \\ +\infty, & \text{otherwise.} \end{cases}$$

3.2 Competitive Learning

During the learning process, the neurons in output space compete to be selected as the winning neuron according to the similarities to the input signal. In most SOFM algorithms, the similarity is evaluated by the distance between the neuron's weight vector and the input signal, especially Euclidean distance. Denote the winning neuron according to input signal x_i as $G(x_i)$. $n_j = G(x_i)$, if $j = arg_{j=1,2,\ldots,N} min \|x_i - w_j\|$.

In the ALM routing problem, the distance metric must be taken as the first term, denoted as

$$g_1(i,j) = K_1 \bullet \|x_i - w_j\| \tag{1}$$

where $i = 1, 2, \ldots, |D|$, $j = 1, 2, \ldots, N$.

However, more terms are necessary to create an overall function to decide which neuron should be selected as the winning one taking into account of the routing constraints.

The input signal x_i is input to the neuron network at time t. Suppose n_j is selected as the winning neuron, it will be associated with h_i. Denote the output neuron topology at time $(t+1)$ as $\widetilde{T}^j_{t+1}$. According to the new connection between n_j and its parent neuron, $\widetilde{A}_{t+1}(n_j)$, in $\widetilde{T}^j_{t+1}$, the multicast tree at time $(t+1)$, T^i_{t+1}, builds a path between h_i and the host associated with n_j's parent neuron.

According to the cost constraint, define the second term in the overall function as:

$$g_2(i,j) = K_2 \bullet \frac{C(T^i_{t+1})}{C(M)} \tag{2}$$

where $C(T^i_{t+1})$ is the sum of all paths' cost in the tree and $C(M)$ is the sum of all element paths' cost.

The values of $max_h \alpha_{T^i_{t+1}}$, $\forall h \in H_t \cup h_i$ and $max_e \beta_{T^i_{t+1}}$, $\forall e \in T^i_{t+1}$ can be calculated. The third and fourth term can be respectively defined as:

$$g_3(i,j) = \begin{cases} K_3 \bullet (1 - max_h \alpha_{T^i_{t+1}}), & \text{if } max_h \alpha_{T^i_{t+1}} < 1; \\ K_3, & \text{Otherwise.} \end{cases} \tag{3}$$

$$g_4(i,j) = \begin{cases} K_4 \bullet (1 - max_e \beta_{T^i_{t+1}}), & \text{if } max_e \beta_{T^i_{t+1}} < 1; \\ K_4, & \text{Otherwise.} \end{cases} \tag{4}$$

according to [0], the fifth term is a penalty in inhibit those Type I neurons which change its matching relationship too often. Denote the changing times for neuron n_j as W_{n_j}, define

$$g_5(i,j) = K_5 \bullet W_{n_j} \tag{5}$$

In the above terms, the values of K_1, K_2, K_3, K_4, K_5 are all positive. The overall function is created by combining the five terms together:

$$g(i,j) = \sum g_l, l = 1,2,3,4,5 \tag{6}$$

The competing process can be summarized as

$$n_j = G(h_i), \tag{7}$$

where $j = arg_{j=1,2,\ldots,N} min(g(i,j))$.

The winning neuron, n_j, and its neighbors will update their weights according to the input signal. Denote the neurons set within ε-hops from n_j in $\widetilde{T}_t$ as $\widetilde{N}_j^{\varepsilon}(t)$, then

$$w_j(t+1) = \begin{cases} w_j(t) + \eta(t)[x_i - w_j(t)], \eta(t) > 0, \text{ if } n_j \in \widetilde{N}_j^{\varepsilon}(t) \ ; \\ w_j(t), \qquad\qquad\qquad\qquad\qquad \text{Otherwise.} \end{cases} \tag{8}$$

3.3 Growing Output Tree

The output neuron space includes only 2 neurons at the beginning. One neuron, n_s, is associated with s, the other is n_s's Type II child. The signal x_i is input to the system at time t, n_j is the winning neuron, $n_j = G(x_i)$. The tree $\widetilde{T}_{t+1}$ is achieved by following rules:

1. If n_j belongs to Type II, both n_j and its parent neuron, $\widetilde{A}_t(n_j)$, create one new Type II child neurons. n_j changes to Type I and is associated with x_i.
2. If n_j belongs to Type I, which means that the neuron has been associated for another host, $G'(n_j)$, n_j will release its matching with $G'(n_j)$, then be associated with h_i. All of its descendants in $\widetilde{T}_t$ will be deleted.
3. If there already exists another Type I neuron associated with h_i, $G(h_i)$, the neuron and its descendants in $\widetilde{T}_t$ will be deleted.

4 Numeric Simulation

4.1 Simulation Environment

The simulation network topology generated by GT-itm [17] is distributed on a 100×100 plant. There are 8 transit routers, 32 stub routers and 21 hosts including the root host. One host is connected to a stub router via a link. An edge's cost is the distance between the pair of routers, all links' cost are set as 1. There are 126 element paths in the mesh totally. The degree upper bound is set as 4 for all hosts. The upper bound of stress is set as 10 for the edges connecting two transit routers, 4 for those connecting two stub routers, 8 for those connecting one transit router and one stub router.

4.2 Simulation Configures

Set initial learning rate $\eta(0) = 0.1$ and $\varepsilon(0) = 1$. Their value are adjusted after 20 steps. K_1 is set as 1, K_2,K_3,K_4,K_5 are equal and are set as the average cost of all element paths, denoted as K.

If all the hosts have been associated to output neurons and the matching relationship do not change in 10 steps or the algorithm has run 200 steps, the algorithm stops.

4.3 Comparison and Analysis

The performance of multicast trees by different routing rules for the same are compared. Besides our algorithm, we build the BMST tree of this topology and all CR tree rooted at each host.

Table 1. Comparison of three ALM routing algorithms

Algorithm	Cost	Host Load Balance	Router Load Balance
Proposed Algorithm	0.1320	0.0150	0.1250
CR	0.1527	0.2475	1
BMST	0.2060	0.4000	0.2500

For each host, run the proposed algorithm 10 times then totally 200 trees are achieved. Table 1 shows the comparison of the performance of the trees by proposed algorithm and those by the CR and BMST. The three columns are respectively the average unitary values of function g_2, g_3 and g_4 of the trees.

The performance of the trees achieved by the proposed algorithm are better than those by BMST and CR on all three metrics:

- On the cost metric: Compared to BMST and CR rules, the average optimization of cost metric achieved by the proposed algorithm is 35.92% and 13.56% respectively.
- On the Host Load Balance metric: Among the total 200 simulations, no host's degree is larger than its degree bound. Only 7 output trees' host load balance metric are not 0. In the other 193 trees there exists at least one host whose degree is equal to its degree bound. The average value of the 200 trees are less the the value of BMST tree and the average value of CR trees, which means the proposed algorithm uses the network resource more efficient.
- On the Router Load Balance metric: Similar to the host load balance metric, no router's degree is larger than its degree bound, and there exists at least one router whose degree is equal to its degree bound in 181 trees. No CR trees are satisfied with the Router Load Balance constraint, which means that there always exists some routers whose degree is larger than its bound in each CR tree. These routers may be the potential bottle neck of the whole system. Compared with BMST, the router load balance metric is also improved obviously by the proposed algorithm.

5 Conclusion and Future Work

Multicast routing problem has been studied for almost two decades. ALM routing are similar to IPM while:

1. Both problems are trying to find a spanning tree;
2. The general constraints are similar: cost, load balance, and so on.

But the difference between them is obvious. ALM routing must take network layer performance into account as well as application layer performance though few protocols pay attention to this.

In this paper, ALM routing problem is summarized as a multi-objective optimizing problem and we propose a new approach based on NN to solve it. The results of numerical simulations show that the proposed algorithm optimizes the overall performance for a given multicast mesh, especially on the application layer load balance and network layer load balance metrics.

Future work will be done along two directions: test and optimize the algorithm by adjusting parameters; test the algorithm in larger multicast topology.

Acknowledgement

This work is supported by the key project of NSFC (No. 60432030).

References

1. Liebeherr, J., Nahas, M., Si, W.: Application-layer Multicasting with Delaunay Triangulation Overlays. IEEE Journal on Selected Areas in Communications **20**(8) (2002) 1472-1488.
2. Pendarakis, D., Shi, S., Verma, D., and Waldvogel, M.: ALMI: An Application Level Multicast Infrastructure. Proceedings of 3rd Usenix Symposium on Internet Technologies and Systems (2001) 49-60.
3. Bharath-Kumar, K., Jaffe, J.M.: Routing to Multiple Destinations in Computer Networks. IEEE Trans. Commun. **31**(3) (1983) 343-351.
4. Kou, L., Markowsky, G., Berman,L.: A Fast Algorithm for Steiner Trees. Acta Informat. **15** (1981) 141-145.
5. Rayward-Smith, V.J.: The Computation of Nearly Minimal Steiner Trees in Graphs. Int. J. Math. Educ. Sci. Tech. **14**(1) (1983) 15-23.
6. Kompella, V.P., Pasquale, J.C., Polyzos, G.C.: Multicast Routing for Multimedia Communication. IEEE/ACM Trans. Networking **1**(3) (1993) 286-292.
7. Tode, H., Sakai, Y., Okada, H., and Tezuka, Y.: Multicast Routing Algorithm for Nodal Load Balancing. Proc. of IEEE INFOCOM **3** (1992) 2086-2095.
8. Shi, S.Y., Turner, J.S.: Routing in Overlay Multicast Networks. Proc. of IEEE INFOCOM **3** (2002) 1200-1208.
9. Cheng, P., Dai, Q.H., Wu, Q.F.: An Application Layer Multicast Routing Algorithm Based on Genetic Algorithms. Proc. of the 8th International Conference on Telecommunications **2** (2005) 413-418.
10. Kranakis, E., Singh, H., Urrutia, J.: Compass Routing on Geometric Networks. Proc. of 11th Canada conference of Computational Geometry (1999).

11. Hamdan, M., El-Hawary, M.E.: Hopfield-Genetic Approach for Solving The Routing Problem in Computer Networks. Proc. of IEEE Canadian Conference on Electrical and Computer Engineering **2** (2002) 823 - 827 .
12. Pan, D., Du, M., Wang, Y., and Yuan, Y.: A Hybrid Neural Network And Genetic Algorithm Approach for Multicast QoS Routing. Proc. of ISNN (2004) 269-274.
13. Xia, Z., Li, P., Yen, I-L.: A Neural Network Based Approach for Overlay Multicast in Media Streaming Systems. Proc. of IPDPS (2004).
14. Gelenbe, E., Ghanwani, A., Srinivasan,V.: Improved Neural Heuristics for Multicast Routing. IEEE Journal on Selected Areas in Communications **15**(2) (1997) 147-155.
15. Pornavalai, C., Chakraborty,G., Shiratori,N.: A Neural Network Approach to Multicast Routing in Real-Time Communication Networks. Proc. of International Conference on Network Protocols (1995) 332-339.
16. Vijay Kumar, B.P., Venkataram, P.: Reliable Multicast Routing in Mobile Networks: a neural-network approach. IEE Proc.-Commun. **150** (2003) 377-384.
17. Zegura, E.W., Calvert, K., Bhattacharjee, S.: How to Model an Internetwork. Proc. of IEEE Infocom (1996) 594-602.

A Neural Network Decision-Making Mechanism for Robust Video Transmission over 3G Wireless Network

Jianwei Wen, Qionghai Dai, and Yihui Jin

Department of Automation, Tsinghua University, Beijing 100084, China
wenjw02@mails.tsinghua.edu.cn

Abstract. This paper addresses the important issues of error control for video transmission over 3G, considering the fact that wireless video delivery faces the huge challenge of the high error rate and time variability in wireless channel. This paper proposes a real world statistics based event-trigger bit error rate (BER) model, which can describe and handle the time-varying wireless channel error characteristics better. Moreover, a recurrent neural network is employed to decide the state transfer as a mechanism. Simulation results and comparisons demonstrate effectiveness and efficiency of the proposed method in term of visual performance and transmission efficiency over a variety of wireless channel conditions.

1 Introduction

The explosive development of the video coding technique and the wireless communications, especially the third generation (3G) wireless network with up to 384 kbps outdoor and 2 Mbps indoor bandwidth for multimedia services, makes it reasonable for delivering video over the wireless channel [1].

However, wireless video delivery faces several challenges, such as high error rate, bandwidth variation, and so on[2]. To cope with the errors during the multimedia transmission, lots of approaches are proposed[3]. As is known, automatic retransmission request (ARQ) and forward error correction (FEC) are two basic error control mechanisms. Furthermore some hybrid ARQ/FEC was proposed to optimize the parity data length to get the tradeoff between the error correction and network traffic efficiency. Some traditional decision-making techniques are used in this error control scheme, such as so-called *Finite Memory Method* and *Iterative Forgot Factor Method* and so on. However, these technique are so coast that couldn't match the real channel conditions well.

In recent years, recurrent neural networks are widely studied in [4, 5], because of their immense potentials of application perspective. The hopfield neural networks and cellular neural networks are typical representative recurrent neural networks among others, and have been successfully applied to signal processing, especially in image processing, and to solving nonlinear algebraic and transcendental equations.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 165–170, 2006.

2 BER Model for Wireless Channel

Wireless video delivery is limited by many unpredictable environment factors. The bit error rate (BER) is the main parameter that depicts the wireless channel character. Furthermore, for every given BER there is an optimal redundancy (R) for every packet. In general, the BER is supposed to be a fixed average value for most error control techniques, just as some two-state Markov model and Gilbert model [6, 7]. Therefore, these error control mechanisms are designed to get as close as possible to the goal by setting the system parameters corresponsively. However, the BER in wireless channel is high and varies over time. Much different from the fore Markov model, our work addresses the additional challenge of modeling traces with time-varying error statistics.

For a given video source, during the process of transmission the BER is supposed not to vary over time, that is, the BER is ber_0 between the time of t_0 and t. However, the BER of wireless channel varies with time rapidly. For example, the BER is ber_1 between the time of t_0 and t_1, and the BER is ber_2 between the time of t_1 and t, here $ber_1 < ber_0 < ber_2$ and $t_0 < t_1 < t$. According to ber_0 the so-called optimized R is got as r_0. Obviously, r_0 is not really the global optimized R for the whole transmission process. When time is between t_0 and t_1, the channel bandwidth is wasted because of $ber_1 < ber_0$. On the other hand, when time is between t_1 and t, the transmission is not robust because of $ber_0 < ber2$. For simplicity, many error control techniques set the BER as the worst case (i.e. ber_2 in this sample) to ensure the transmission robustness at the cost of bandwidth resource. When the interval between t_0 and t_1 is very small relative to that between t_1 and t, that approximate method is acceptable. Otherwise, the waste of bandwidth would badly impact the visual performance.

The essential reason for wasting the transmission efficiency is that the strategy is activated by the transmission events while not by the variation events of the wireless channel BER. Therefore, a BER state transfer model is presented to serve the novel adaptive error resilient scheme. A two BER state transfer model is shown as fig.1.

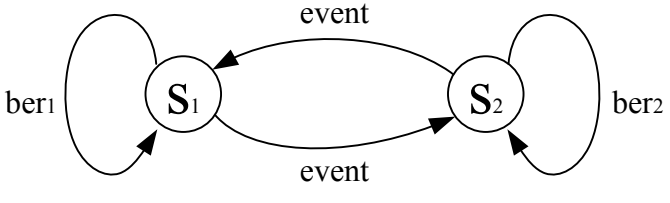


Fig. 1. Two BER state transfer model

S_1 and S_2 represent two different channel states separately. At state S_1, the average value of the BER is ber_1, the variance of the BER is smaller to a given threshold value just like 0.05. The BER is different when the state is different.

The event is the statistics character of the feedback messages. There is a decision-making mechanism in the encoder to monitor if the state is changing. And then, some corresponding actions are taken to estimate the current channel state and get the optimal R. In next section, a hopfield neural network decision-making mechanism in the feedback-based error control scheme would be proposed.

3 Decision-Making Mechanism Based on Recurrent Neural Network

Wireless video delivery has to consider the impact of the high and time-varying bit-error rate in wireless channel. For the channel-adaptive hybrid ARQ/FEC techniques, if the original message length is K, then after adding extra parity data of length R, the codeword is of length N. Obviously, when N is given, the larger the value of R, the more errors it would correct. So choosing an appropriated R to trade-off error correction and network traffic should be considered carefully.

Let *ber* and *per* represent the bit error probability in the channel and the packet error probability separately, let N represents the number of the length in the packet, and let r represents the length of the parity data in one packet. Then it is thought the packet has an error when the number of the error bits in one packet is bigger than $r/2$, and then sends the NACK message to the encoder. When the *ber* is given, the *per* would be got by the following equation.

$$per = 1 - \sum_{i=0}^{r/2} C_N^i ber^i (1 - ber)^{N-i} \tag{1}$$

And then, the optimized r_{opt} could be gained through the relations (2).

$$r_{opt} = min\{r | per \leq perobj\} \tag{2}$$

where *perobj* represents the expected value of the system depended on the capability of the mobile terminal.

In section 2, it has been analyzed that there are several states for the wireless channel. These states are called stable state with constant BER. The process transferring from one stable state to another state is called the temporary hop state. Let $d(n)$ represents the decision-making point at the nth moment. $d(n) = 1$, says that the channel is in the temporary hop state, while $d(n) = 0$ says that it is in the stable state. Let $r(n)$ represents the redundant data length at nth moment. Obviously, when $d(n) = 1$ that is the channel is in hop state, the $r(n)$ should be adjusted to match the real wireless channel BER. Otherwise, when $d(n) = 0$ that is the channel is in stable state, the $r(n)$ should be hold on because the error in the transmission is caused by noise disturbance. So $d(n)$ is the key parameter in this algorithm.

Generally, *Finite Memory Method* and *Iterative Forgot Factor Method* are two basic method used to get $perobs(n)$ as the threshold values of $d(n)$.

$$d(n) = \begin{cases} 0, perobs(n - rttn) \leq perobj \\ 1, perobs(n - rttn) > perobj \end{cases} \tag{3}$$

where *perobs* represents the packet error rate observed in the encoder based on the feedback message. n represents the current moment of the packet being compressed, *rttn* represents the number of packets transmitted within the round trip time.

3.1 Finite Memory Method

For the calculation of the *perobs*, a general method is that the hits for Stat. is limited to some finite quantity, which is the so-called *Finite Memory Method*, just as follow.

$$perobs(n - rttn) = \frac{1}{N} \sum_{i=0}^{N-1} m(n - rttn - i) \tag{4}$$

where N represents the total number of the sampled packets, and $m(i)$ represents the feedback message of the packet transmitted at the ith moment, the value of 1 and 0 represent the message NACK and ACK.

3.2 Iterative Forgot Factor Method

The *Finite Memory Method* mentioned above doesn't work well for ignoring the new information. An *Iterative Forgot Factor Method* is used to cope with that issue.

$$\begin{aligned} perobs(n - rttn) = (perobs(n - rttn - 1) \times \beta + \\ \tfrac{1}{N} m(n - rttn)) \times \tfrac{1}{1+\beta} \end{aligned} \tag{5}$$

And the first value would be set as

$$\begin{aligned} perobs(n - rttn) = (perobs(n - rttn - N) \times \beta^N + \\ \tfrac{1}{N} \sum_{i=0}^{N-1} m(n - rttn - i)\beta^i) \times \tfrac{1-\beta}{1-\beta^{N+1}} \end{aligned} \tag{6}$$

where β is a decay factor within 0.95 and 1, and the value of β decides the pace of tracing the varied *berobs* in the channel.

3.3 Hopfield Decision-Making Mechanism

The key parameter β in the *Iterative Forgot Factor Method* is fixed for any given application. Furthermore, it is worth to note that the natural relation between *perobs* and *d*. *perobs* comes from the feedback message m, d is decided by the value of *perobs*, while d affects m objectively after an adjustment on r. The Hopfield Decision-Making Mechanism proposed utilized the feedback messages directly without the use of *perobs*, proposed as follow:

$$\begin{cases} v(n) = \sum_{i=0}^{N-1} w_i m(n - i) - perobj \\ d(n) = sgn\{v(n)\} \\ m(n + 1) = P\{d(n - rttn + 1)\} \end{cases} \tag{7}$$

where the first value of m comes from the history data, function P represents the process of adjustment on r and the video transmission over wireless channel.

4 Simulation Results and Comparisons

Some common test conditions for wireless video services have been presented in [8]. Two of them for packet-switched streaming services over 3G are given in Table I. It is assumed that the wireless environment varied between the pattern 1 and the pattern 2. So the *ber* changes the value between 2.9e-3 and 9.3e-3.

Table 1. Bit-error patterns

No.	Bit rate	Length	BER	Mobile Speed	Application
1	64 kbit/s	60 s	9.3e-3	3 km/h	Streaming
2	64 kbit/s	60 s	2.9e-3	3 km/h	Streaming

We simulated the proposed algorithm *Hopfield Decision-Making Mechanism* and the *Iterative Forgot Factor Method*. The parameters were chosen as follows. $N = 255$bytes, $n_p = 6000$, $t_{max} = 5$, $N = 1000$, $\beta = 0.95$, $perobj = 0.05$. According to the given varying trend of BER and the definition in this paper, the actual optimized value of R would change between 20bytes and 50bytes.

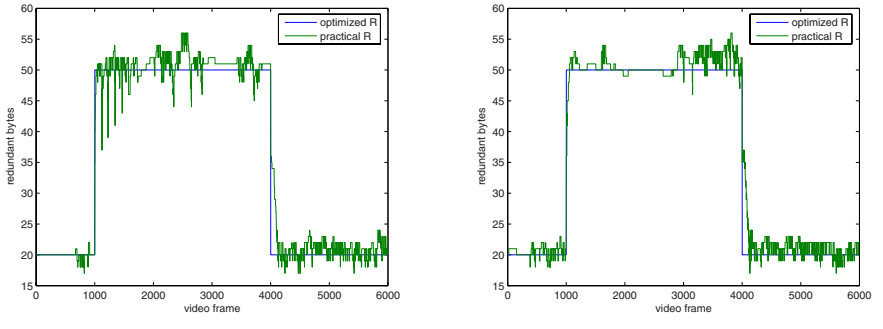


Fig. 2. The contrast of the redundancy length for every packet between the *Iterative Forgot Factor Method* and the *Hopfield Decision-Making Mechanism*

Fig. 3. The contrast of the visual performance between the *Iterative Forgot Factor Method* and the *Hopfield Decision-Making Mechanism*

For the *Iterative Forgot Factor Method*, the average value of R is 32.573 bytes, the average times for retransmission is 0.053 and the transmission efficiency γ is 0.7656. While for the *Iterative Forgot Factor Method*, the average value of R is 28.497 bytes, the average times for retransmission is 0.043 and the transmission efficiency γ is 0.8426. The results of the simulations are given as Fig.2 and Fig.3, which show that the proposed technique performs better than the traditional method.

5 Conclusion

To the best of our knowledge, this is the first paper to utilize a recurrent neural network decision-making mechanism for robust wireless video delivery. Theoretical performance analysis and simulation results demonstrate the effectiveness of the proposed algorithm. The future work is to design more accurate error models for wireless channel and predictive control techniques for delay.

Acknowledgement

This work is supported by the key project of National Natural Science Foundation of China (Grant No. 60432030).

References

1. Mohr, W.W., Onoe, S.: The 3GPP Proposal for IMT-2000. IEEE Commun. Mag. 37(2) (1999) 72-81
2. Zhang, Q., Zhu, W., Zhang, Y.: Channel-adaptive Resource Allocation for Scalable Video Transmission over 3G Wireless Network. IEEE Trans. Circuits Syst. Video Technol. 14(8) (2004) 1049-1063
3. Wang, Y., Zhu, Q.-F.: Error Control and Concealment for Video Communication: A Review. Proceedings of the IEEE 86(5) (1998) 974-997
4. Cao, J., Wang, J.: Global Asymptotic Stability of Recurrent Neural Networks with Lipschitz-continuous Activation Functions and Time-Varying Delays. IEEE Trans. Circuits Syst. I 50(1) (2003) 34-44
5. Zeng, Z., Wang, J., Liao, X.: Global Exponential Stability of a General Class of Recurrent Neural Networks with Time-Varying Delays. IEEE Trans. Circuits Syst. I 50(7) (2003) 1353-1359
6. Bolot, J., Fosse, S., Towsley, D.: Adaptive FEC-based Error Control for Internet Telephony. Proc. Infocom'99 (1999)
7. Balakrishnan, H., Katz, R.: Explicit Loss Notification and Wireles Web Performance. Proceedings of the IEEE Globecom Internet Mini-Conference (1998)
8. Stockhammer, T., Hannuksela, M.M., Wiegand, T.: H.264/AVC in Wireless Environments. IEEE Trans. Circuits Syst. Video Technol. 13(7) (2003) 657-673

A Multilevel Quantifying Spread Spectrum PN Sequence Based on Chaos of Cellular Neural Network

Yaqin Zhao, Nan Zhao, Zhilu Wu, and Guanghui Ren

School of Electronics and Information Technology, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China
{yaqinzhao, wuzhilu, rgh}@hit.edu.cn

Abstract. A novel multilevel quantifying spread spectrum PN sequence based on the chaos of Cellular Neural Network (CNN) is proposed in this paper. The chaotic sequences are created from a CNN with three cells and multilevel quantified to be the PN sequences for the spread spectrum communication systems (SSCS). And then better sequences are filtered out in terms of the equilibria points, self-correlation and mutual-correlation of the chaotic PN sequences. These PN sequences can provide more enhanced multiple access capacity and robustness to the noises than conventional *m*- and Gold sequences because of the sensitivity to the initial conditions of the chaotic sequences and the good dynamical performance of the CNN. The filter processing helps the SSCS to resist on the rake declination and the interference from other users. The experiment results show that the proposed chaotic PN sequences are much better than the conventional sequences for SSCS.

1 Introduction

The spread spectrum communication techniques used in direct sequence-code division multiple access (DS/CDMA), secure communication and so on, has received much attention due to its advantages for providing enhanced multiple access capacity and robustness to the noises. As the information carriers, the pseudorandom or pseudo noise (PN) sequences are utilized to spread the signal bandwidths, and to recover the narrowband messages at the receiver.

Conventional PN sequences are represented by the class of maximal length (*m*-) sequences generated by linear feedback shift registers. However, the number of *m*-sequences may be insufficient for DS/CDMA systems with a large number of users. Now, the sensitivity to the initial conditions of the chaotic sequences has been the main cause for exploiting chaos in SSCS with a rich set of PN sequences in the literatures [1-3], because the sequences generated in this manner diverge to different trajectories in a few iterations even though their initial conditions differ by <1%. At present, the chaotic PN sequences for SSCS with nice statistical properties are almost created by two-level quantifying the real-value sequences, which are often generated by Logistic or Chebyshev mapping with different initial values x_0. However, this two-level quantization will result in great information loss. Additionally, reliable hardware implementations of chaos PN sequence generators realized by piecewise linear

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 171 – 177, 2006.

analogue functions and output quantization is difficult due to manufacturing process variations among different IC production lots, transistor mismatches and noise [4-5].

Cellular neural network (CNN) constitutes a class of recurrent and locally coupled arrays of identical dynamical cells, which can be implemented easily by VLSI and be applied to signal processing problems. And it is also a nonlinear dynamical system in which complex fraction, chaotic and hyperchaotic phenomenon are existed. Especially, the complex dynamical property, intensive randomicity and unpredictability of the chaotic systems are of importance for the secure communication systems [6-7]. In this paper, a new multilevel quantified and two-phase encoded chaotic PN sequence generator based on CNN is proposed for SSCS. And some methods adopted to improve the performance of the PN sequences.

2 Chaos Sequence Generation of CNN

A CNN was composed of regular distributed dynamical cells, in which each cell can communication with its neighbors directly and with the rest indirectly. So the nice continuous time domain characteristic of CNN can meet the needs of real-time digital signal processing. Additionally, the local mutual communication among cells makes it very convenient to realize the CNN with VLSI.

2.1 Chaos of CNN with Three Cells

In recent years, many researchers found that there are complex chaotic phenomena with one positive Lyapunov exponents in the chaotic systems at least. More complex chaotic dynamical behaviors only existed in the CNN systems with three cells or more than. In this paper, a CNN with three cells is adopted to create the chaotic sequences for SSCS considering the hardware implementation complexity and the spread spectrum communication systems` requirements.

The dynamics of the CNN with 3 cells can be described by

$$\begin{aligned} \dot{X}_1 + X_1 &= p_1 f(X_1) - sf(X_2) - sf(X_3) \\ \dot{X}_2 + X_2 &= -sf(X_1) + p_2 f(X_2) - rf(X_3) \\ \dot{X}_3 + X_3 &= -sf(X_1) + rf(X_2) + p_3 f(X_3) \end{aligned}, \tag{1}$$

where $p_1>0$, $p_2>0$, $p_3\geq 0$, $r>0$, $s>0$, the input signal and the current offset are set to none. With different parameters $\{p_1, p_2, p_3, r, s\}$ and the initial values $\{X(0)\}$, we can get the chaotic phenomena from the CNN architecture shown in (1). So we let the parameters $\{p_1, p_2, p_3, r, s\} = \{1.25, 1.1, 1, 3.2, 4.4\}$, and the initial values $\{X(0)\} = \{0.1, 0.1, 0.1\}$. And the Runge-Kutta method is specified to realize the iterative computation with automatic pace adjustments. We define three subsets in R^3 space as

$$\begin{aligned} D_+ &= \{(X_1, X_2, X_3) \,|\, X_1 \geq 1,\ |X_2| < 1, |X_3| < 1\} \\ D_0 &= \{(X_1, X_2, X_3) \,\|\, X_1| < 1,\ |X_2| < 1, |X_3| < 1\} \\ D_- &= \{(X_1, X_2, X_3) \,|\, X_1 \leq -1, |X_2| < 1, |X_3| < 1\} \end{aligned}. \tag{2}$$

Then three equilibria points are

$$\begin{aligned} p_+ &= (1.971,\ 0.7273,\ -0.7107) \in D_+ \\ p_0 &= (0,\ 0,\ 0) \in D_0 \\ p_- &= (-1.971,\ -0.7273,\ 0.7107) \in D_- \end{aligned} \quad . \tag{3}$$

The Jacobi matrix of these equilibria points is the same as the state matrix of the linear self-controlling systems. Then

$$\begin{aligned} J_+ = A_+ &= \begin{bmatrix} -1 & -s & -s \\ 0 & -1+p_2 & -r \\ 0 & r & -1+p_3 \end{bmatrix}, \quad v_+ = \begin{bmatrix} p_1 \\ -s \\ -s \end{bmatrix} \\ J_0 = A_0 &= \begin{bmatrix} -1+p_1 & -s & -s \\ -s & -1+p_2 & -r \\ -s & r & -1+p_3 \end{bmatrix}, \quad v_0 = 0 \\ J_- = A_- &= \begin{bmatrix} -1 & -s & -s \\ 0 & -1+p_2 & -r \\ 0 & r & -1+p_3 \end{bmatrix} = A_+, \quad v_- = \begin{bmatrix} -p_1 \\ s \\ s \end{bmatrix} \end{aligned} \quad . \tag{4}$$

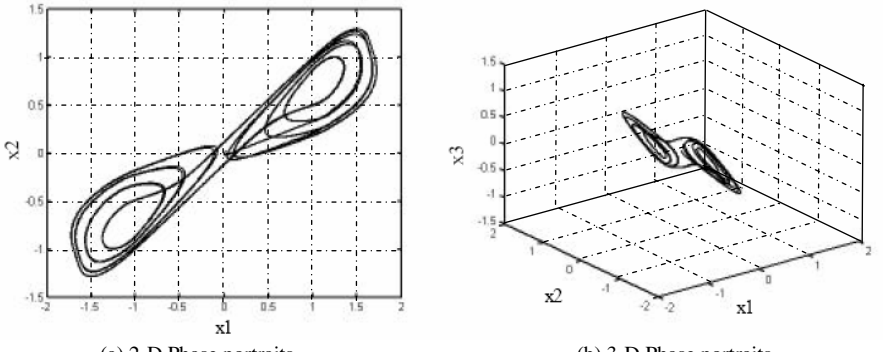
According to the definition of the Lyapunov exponents as (5)

$$\lambda(x_0) = \lim_{n\to\infty} \frac{1}{n} \log \left| \frac{\mathrm{d}f^n(x_0)}{\mathrm{d}x_0} \right| = \lim_{n\to\infty} \frac{1}{n} \log \left| \prod_{i=0}^{n-1} f'(x_i) \right| = \lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} \left| f'(x_i) \right| . \tag{5}$$

So we can get three Lyapunov exponents of the CNN (1) as following

$$\lambda_1 = 0.144, \lambda_2 = -0.001, \lambda_3 = -0.629 . \tag{6}$$

Because the maximal Lyapunov exponent value is more than zero, so we get the chaotic phenomena from the proposed CNN system, which are shown in the 2-D and 3-D simulation phase portraits as Fig.1.



(a) 2-D Phase portraits (b) 3-D Phase portraits

Fig. 1. Phase portraits of the proposed chaotic CNN

2.2 Multilevel Quantifying Chaotic PN Sequences

We adopt the above CNN proposed in section 2.1 to create three chaotic real-value sequence sets $\{X^i(mk): i=1,2,3,\ k=0,1,2,3\cdots\}$ which have been transferred into the range (0, 1) by linear transmissions. Multilevel quantification method is presented to avoid the great information loss during the sequence generating. The whole architecture of the chaotic PN generator based on CNN is shown in Fig.2.

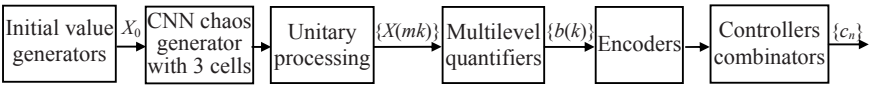


Fig. 2. The chaotic PN generator based on CNN

The chaotic real-value $X(mk)$ sequences can be expressed as

$$X(mk) = 0.x_0(k)x_1(k)x_2(k)\cdots . \tag{7}$$

The former several values $x_0(k)$ $x_2(k)$ $x_2(k)$… should be removed, and the number of the removed values j is often ≥10. Afterwards, the new sequences $\{X'(mk)$**:** k=0, 1, 2,…} is described as

$$X'(mk) = 0.b_0(k)b_1(k)b_2(k)\cdots , \tag{8}$$

where $b_n(i) = x_{n+j}(i)$. Then we quantify the sequence $X'(mk)$ uniformly into 2^{2r} (m≥2r≥1) ranks, and get

$$X_T(mk) = b_0(k)b_1(k)b_{2r-1}(k) , \tag{9}$$

in which, $b_l(i)\in\{0, 1\}$, $l\in\{0, 1,\ldots, 2r\text{-}1\}$ and

$$\sum_{l=0}^{2r-1} 2^{-(l+1)} b_l(k) < X'(mk) < \sum_{l=0}^{2r-1} 2^{-(l+1)} b_l(k) + 2^{-2r} . \tag{10}$$

Connect each $X_T(mk)$ (i=0,1,…), we will get the multilevel quantified sequence $\{b_n\}_{n=0}^{2N-1}$

$$\{b_n : b_n = b_l(i), n = 2ri + l = 0,1,2,\cdots,2N-1\} . \tag{11}$$

Afterwards, according to the requirements of spread spectrum communication systems, the multilevel quantified sequence $\{b_n\}_{n=0}^{2N-1}$ will be encoded into the specified Chaotic PN sequences $\{c_n\}$. Because there are three chaotic PN sequences from the proposed CNN systems, three quantifiers and encoders are needed to complete the PN sequence creations.

3 PN Sequences Evaluation and Filter

For the SSCS such as an asynchronous DS/CDMA system, the characteristics of PN sequences will have deep influences on the performance of the communication

systems. So in this paper, we advance to filter the chaotic PN sequences in terms of the equilibria points, self-correlation and mutual-correlation.

The equilibrium of the spread spectrum sequences is that the number of "1" should be almost equal to that of "-1", which will affect the restrainability to the carriers of the communication systems. Unequalized PN sequences will disturb the secureness and the robustness of the DS/CDMA systems. For the chaotic sequence $\{u_n\}_{n=0}^{N-1}$, the quantity difference between "1" and "-1" is L

$$L = \sum_{n=0}^{N-1} u_n \ . \tag{12}$$

Because the probabilities of "1" and "-1" in the chaotic PN sequences $\{u_n\}_{n=0}^{N-1}$ are both 0.5, and the mean is 0, and the variation is 1, L is a Gauss distribution whose mean is zero and variation is N. If N is an odd, when $|L|=1$, the sequence is equalized one. If N is an even, when $L=0$, the sequence is equalized one.

The second parameter to filter the PN sequences is the self-correlation function, which will have a deep influence on the ability to resist on the rake declination to the communication systems. The ideal self-correlation function of a PN sequence is a δ function. So we set a threshold for the maximal side lobe of the self-correlation function. And the probability distribution of the maximal side lobe for the chaotic PN sequences with the length N is described as

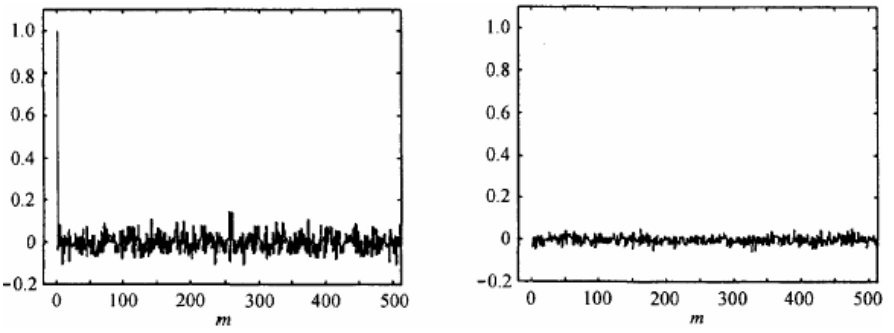
$$p(\max(|R_u(\tau)|) \le k) \approx \begin{cases} \left[2\Phi((k+1)/\sqrt{N})-1\right]^{\left\lceil \frac{(N-1)}{2} \right\rceil}, & k: \text{O/E while } N: \text{O/E} \\ \left[2\Phi((k)/\sqrt{N})-1\right]^{\left\lceil \frac{(N-1)}{2} \right\rceil}, & k: \text{O/E while } N: \text{E/O} \end{cases}, \tag{13}$$

where $\Phi(x) = (1/\sqrt{2\pi}) \int_{-\infty}^{x} e^{-\frac{t^2}{2}} dt$, $k>0$, O is odd, E is even, and $\lceil x \rceil$ represents the minimal integer $\ge x$.

Another important parameter of the PN sequences is the mutual-correlation function, which is more complex to be measured than other two. And the multiple access performance of a DS/CDMA system is decided by it. For a system with K users, the performance of the i^{th} user can be decided by the sum of mean square values of the mutual-correlation between it and others V_i, shown as

$$V_i = \sum_{\substack{k=1 \\ k \ne i}}^{K} \left[\frac{1}{N} \sum_{\tau=0}^{N-1} R_{ki}^2(\tau) + \frac{1}{N} \sum_{\tau=0}^{N-1} \theta_{ki}^2(\tau) \right]. \tag{14}$$

In Fig.3, we can see more ideal self- and mutual-correlation performances of the chaotic PN sequences based on CNN than m- and Gold sequences. The self-correlation function looks very close to the δ function, and the mutual-correlation value is very minimal to zero. Additionally, from the experiments, we find that the correlation performance will be better and better with the increasing of sequence length N, and the architecture pf the CNN system will be more and more complex. So we choose an available sequence length N.



(a) The self-correlation performance (b) The mutual-correlation performance

Fig. 3. The performance of the chaotic PN sequences (N=500)

4 Experiment Results

In order to evaluate the performance of the multilevel quantifying CNN PN sequences, a DS/CDMA system with four users is designed as Fig.4.

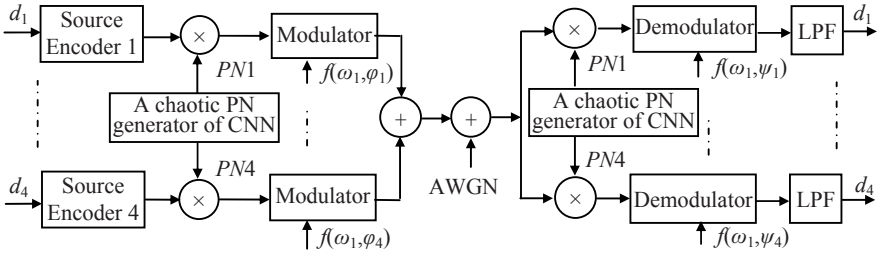


Fig. 4. A SSCS based on the chaotic PN sequences of CNN

In Fig.4, there are four users in the spread spectrum communication system and the PN sequences are all filtered out from the chaotic sequences generated by a three cell CNN proposed in section 2. The quantifying level r in *Eqn*. (10) is set to 1 and 8 respectively, so we can analysis the performance of the multilevel quantifying process. At the same time, we use m-sequences to take the place of the chaotic PN sequences based on CNN to observe the performance improvement of the chaotic PN sequences. The BER curves are shown in Fig.5.

The experiment results show that if the quantifying level r of the chaotic PN sequences is larger, the performance of the spread spectrum communication system will be better at the same SNR. In addition, from the BER curves we can draw a conclusion that the chaotic PN sequences generated by the presented CNN system is more suitable for the spread spectrum communication systems, because the SNR of the chaotic sequences can improved about 2~3dB than m-sequences.

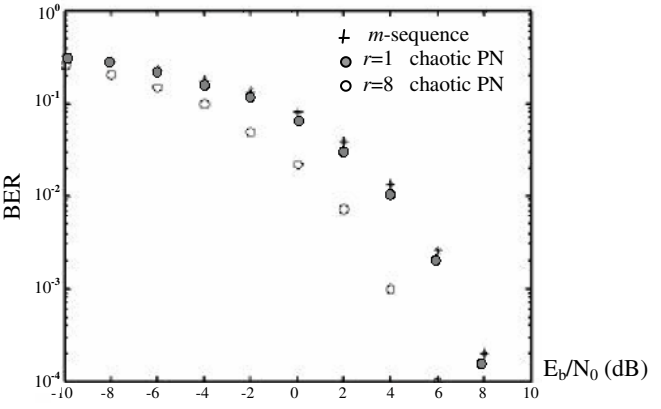


Fig. 5. Performance comparison of the PN sequences

5 Conclusions

In this paper, a novel PN sequence generator based on the cellular neural network for the spread spectrum communication systems is developed. The simulation results show that the performance of the presented multilevel quantifying chaotic PN sequences is much better than that of conventional *m*-sequences and Gold-sequences. And this kind of PN sequences can meet the requirements of the spread spectrum communication systems with large number of users well. Another important advantage of these PN sequence generators is that it is very easy to be realized with very large scale integrated circuits, which is very crucial for their applications in the practical spread spectrum communication systems.

References

1. Heidari-Betani, G., McGuillem, C.D.: A Chaotic Direct-Sequence Spread Spectrum Communication System. IEEE Trans. Communication **42** (1994) 1524-1527
2. Rovatti, R., Setti, G., Mazzini, G.: Chaotic Complex Spreading Sequences for Asynchronous DS-CDMA - Part II: Some Theoretical Performance Bounds. IEEE Trans. Circuits & Systems **45** (1998) 496-506
3. Mazzini, G., Rovatti, R., Setti, G.: Non-Average Performance of Chaos-based DS-CDMA: Driving Optimisation towards Exploitable Maps. IEEE Int. Symp. On Circuits and Systems **1** (2000) 723-726
4. G.Otz, M., Schwarz, W.: Statistical Analysis of Chaotic Markov Systems with Quantised Output. IEEE Int. Symposium on CAS **5** (2000) 229-232
5. Delgado-Restituto, M., Rodriguez-Vazquez, A.: Mixed-Signal Map-Configurable Integrated Chaos Generator for Chaotic Communications. IEEE Trans. CAS **48** (2001) 1462-1474
6. Chua, L.O., Yang, L.: Cellular Neural Networks: Theory, IEEE Trans. on CAS **35** (1988) 1257-1272
7. Yang, T., Chua, L.O.: Chaotic CDMA Communication Systems. International Journal of Bifurcation and Chaos **7** (1997) 2789-2905

An Experimental Hyper-Chaos Spread Spectrum Communication System Based on CNN

Jianye Zhao, Quansheng Ren, and Daoheng Yu

Department of Electronics, Peking University,
Beijing 100871, China
phdzjy@263.net

Abstract. A new hyper-chaos communication system based on cellular neural network (CNN) is proposed in this paper. Hyper-chaos is generated with fifteen-cell CNN, then it's transferred to a digital sequence. The chaotic sequence is better than gold sequence when they are utilized in direct sequence spread spectrum system. Compared with the traditional gold sequence system, there is 3dB improvement in CNN chaotic sequence system when the channel is multi-path channel. Because of the complex dynamic behavior of hyper-chaos, security signal could be transferred through wireless channel. The structure of hardware CNN spread spectrum system is also shown, and the security of CNN communication is analyzed at last.

1 Introduction

Spread spectrum technique is widely used in mobile phone, wireless local network and other consume electronic equipment, and the spread sequence utilized in the system is very important [1]. The chaotic sequence is employed in spread spectrum communication system because of its excellent random characteristic, but it's difficult to get qualified real-time chaotic sequence. In this paper, a new spread spectrum communication system based on CNN is proposed. Real-time chaotic sequence is generated with CNN [2], then the sequence is modulated with DSP modulator. The experimental results confirm the good quality of CNN chaotic sequence.

Neural network is also a good solution to hyper-chaos generator. Such neural network as CNN is very easily realized with analogy circuits, and the performance of these neural network equalizers is also very good. Based on these reasons, CNN hyper-chaos generator for spread spectrum system is designed in this paper, and the simulation confirms the effectiveness of hyper-chaos spread spectrum system.

This paper is organized as follows: Section 2 describes hyper-chaos in CNN system. We discuss the stability of CNN's with opposite-sign template in this section, and the chaotic attractor is shown. In section 3, the random characteristic of chaotic sequence is analyzed. A new spread spectrum system is shown in section 4, and the experimental result is also given. Section 5 shows some concluding remarks.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 178 – 183, 2006.

2 Chaos in CNN

The CNN system can be described as a group of equations:

$$\frac{dV_{Xij}(t)}{dt} = -\frac{1}{R_X}V_{Xij}(t) + \sum_{(k,l)\in N_r(i,j)} d_{ijkl} f_{Ykl}(t) + \sum_{(k,l)\in N_r(i,j)} B_{ijkl} u_{kl}(t) + \frac{1}{\sigma^2} z_{ij} \tag{1}$$

$V_{Xij}(t)$ is the state of cell C(i,j), $f_{Yij}(t) = \frac{1}{2}(|V_{Xij}(t)+1| - |V_{Xij}(t)-1|)$,u is the value of input, z is the offset.

When the characteristic of chaotic behavior is discussed, the *Lyapunov analysis* approach is employed, and cellular neural network is regarded as a dynamic system. When a five cells network is studied, hyper-chaos is generated [3]. Because the offset should be calculated, other input is zero, the CNN equation (1) should be written as :

$$\dot{x}_i = Cx_i + df(x_i) + z \quad for\ i = 1,\ldots,N \tag{2}$$

The connection relationship between number i cell and other cells is:

$$I_i = H_i\left[x_1^T, x_2^T, \ldots, x_N^T\right]^T \quad for\ i = 1,\ldots,N \tag{3}$$

In (3), $I_i \in R^M$ is the connected electric current, $H_i \in R^{M\times MN}$ is matrix that describe the connection relationship of the cells. If we choose the parameter, hyper-chaos will be generated: $C = \begin{bmatrix} -3.2 & 10 & 0 \\ 1 & -1 & 1 \\ 0 & -14.87 & 0 \end{bmatrix}$, $d = \begin{bmatrix} 2.95 \\ 0 \\ 0 \end{bmatrix}$, $z = 0$, $\begin{bmatrix} I_1 \\ I_2 \\ \cdots \\ I_5 \end{bmatrix} = H \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_5 \end{bmatrix}$,

$$H = \begin{bmatrix} -\hat{H} & \hat{H} & 0 & 0 & 0 \\ 0 & -\hat{H} & \hat{H} & 0 & 0 \\ 0 & 0 & -\hat{H} & \hat{H} & 0 \\ 0 & 0 & 0 & -\hat{H} & \hat{H} \\ \hat{H} & 0 & 0 & 0 & -\hat{H} \end{bmatrix},\ \hat{H} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

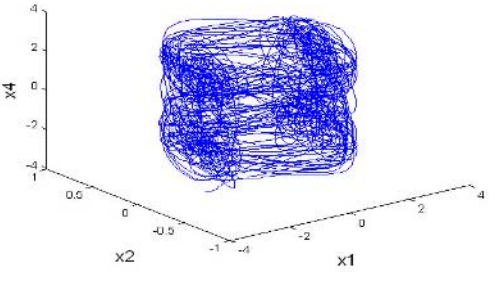


Fig. 1. The CNN hyper-chaos attractor

If CNN includes three cells, the chaos will be generated if the parameters are chosen carefully [4]. There is very complex dynamical behavior in bigger sized network [5], such as fifteen cells CNN. Such dynamical behaviors as hyper-chaos could also be utilized. The utilization of chaos is attractive to wire-less communication, because secure communication can be realized with a synchronizing chaos system, and it's difficult to decipher the sequence. Then CNN hyper-chaos is put into an A/D converter, the digital CNN chaotic sequence is generated.

3 Random Characteristic of CNN Chaotic Sequence

We generate the digital pseudo-sequence with CNN system. Because the dynamic of CNN is very complex, we can get the excellent sequence for spread spectrum. We put CNN chaotic into a simple A/D converter, and we get the digital pseudo-random sequence. The value of chaos is x: if x>0, the corresponding value of binary sequence is *x,* then x=1; if $x \le 0$, the corresponding value of binary sequence is *x,* then x=-1;

The random characteristic can be described as following context:

At first, the mean value of binary sequence is zero. Because each mean value of the cell is zero. After sampled with A/D, and the sampling frequency ($Dt = 1\mu s$) is very fast, the sequence do not change its mean value, the mean value is described as:

$$\bar{x} = \lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} x_i = 0 \tag{4}$$

The auto-correlation function of the CNN chaotic sequence is described as follows

$$ac(m) = \lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} (x_i - \bar{x})(x_{i+m} - \bar{x}) = \lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} x_i x_{i+m} \tag{5}$$

The cross-correlation function of the CNN chaotic sequence is:

$$cc_{12}(m) = \lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} (x_{1i} - \bar{x})(x_{2(i+m)} - \bar{x}) = \lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} x_{1i} x_{2(i+m)} \tag{6}$$

The CNN hyper-chaotic sequence is simulated with TMS320C54 DSK tools, the auto-correlation function is almost the δ function, and the cross-correlation function is almost equal to 0.

4 The CNN Chaotic Spread Spectrum Communication System

In traditional direct sequence spread spectrum (DSSS) system, the Gold sequence is employed. The CNN hyper-chaotic sequence could be employed in DSSS because of its statistical characteristic, the characteristic is similar with that of Gold sequence: the auto-correlation function is almost the δ function, the cross-correlation function is

almost equal to 0,. The CNN hyper-chaotic sequence is sensitive to initial value, so lots of different sequences can be generated. It's attractive to DSSS system. The transmitter of DSSS system is described as:

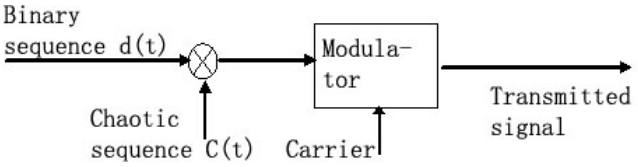


Fig. 2. DS-BPSK Modulator

In Fig.2, C(t) is spread spectrum signal. Usually C(t) is named as direct sequence. Both gold sequence and CNN chaotic sequence are good direct sequences, but the quality of CNN sequence is better.

Usually the signal passes the additive white Gaussian noise channel, so the receiver is described as Fig 3. Fading channel will studied in the following works.

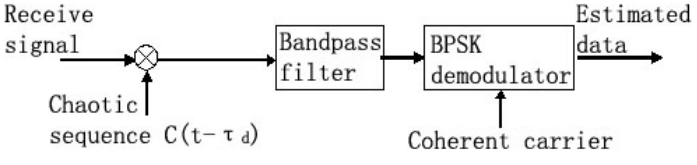


Fig. 3. DS-BPSK demodulator

Different sequences will be generated if the initial value is different, so multi-user could the hyper-chaotic DSSS system. An experimental CNN-DSSS system based on TMS320C5416 DSP is set up, we test two kinds of sequence in our experiment:127 bit Gold sequence, and CNN sequence which is divided into 127 bit segment. The Bit error rate curve is shown in Fig.4. The application of 127 bit is mobile phone system such as IS-95 or CDMA 2000. We also test 31bit sequence, and its application is wireless network such as UWB system. We simulate a multi-path channel, and the channel model that we have used for this simulation is the familiar delay line model as follows:

$$r(n) = \frac{1}{\sqrt{2}}\left[t(n) + t(n-1)\right] + \eta(n) \tag{7}$$

In equation 7, r(n) is the receive PN code, t(n) is the transmitted PN code, and $\eta(n)$ is additive white Gaussian noise. Fig.4 shows the simulation results:

In Fig 4,o represents that DS is Gold sequence; * represents that DS is CNN chaotic sequence. The marks in these two figures are different because the simulation results are gotten in different sequence. Compared the gold sequence system, there is

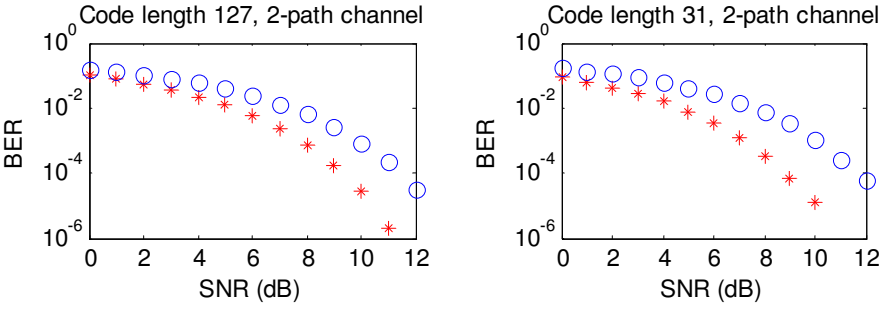


Fig. 4. Bit error rate of DSSS system in multipath channel

almost 3dB improvement in CNN chaotic sequence system when the 31bit sequence is utilized, and there is 2dB improvement when the 127 bits sequence is utilized. The CNN chaotic sequence is excellent in the two-path channel because of its random characteristic. Because CNN have complex dynamic behaviors, it's difficult to reconstruct the phase space of the hyper-chaos, so the security of the system is reliable. We show another figure (Fig 5) when the parameters of hyper-chaos generators are different, the difference is 1×10^{-9}, and the bit error rate is too bad to use. If we want to

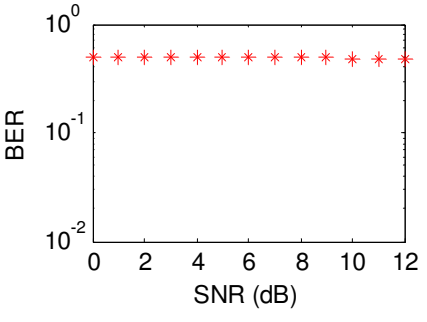


Fig. 5. Bit Error Rate when the parameters are different

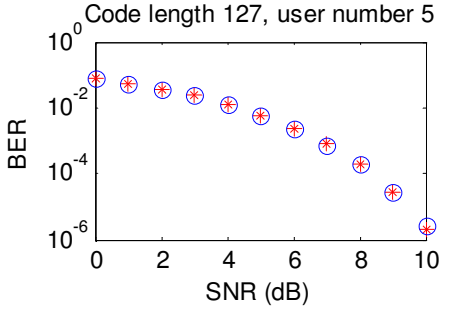


Fig. 6. Bit error rate of DSSS system in multi-user environment

decipher the system, because the chaotic sequence generator contains 15 cell, we must calculate 1.5×10^{10} times at least, it's very difficult. In fact, because the connection relationship is very complex, the calculation is greater than the estimation.

Multi-user characteristic is shown in Fig.6, and the number of user is five. In Fig 6, o represents that DS is Gold sequence; * represents that DS is CNN chaotic sequence. The performance of chaotic sequence is slightly better than that of Gold sequence. Both two kinds of sequence have the best performance because of their cross-correlation characteristic. The cross-correlation characteristic is the most optimum in all kinds of possible sequences, and the difference between them is the disturbance of the noise.

5 Conclusions

In this paper, a new DSSS system based on hyper-chaos is proposed, and an experimental DSSS system is set up, and the DSSS system is set up with TMS320c5416 DSP. Both CNN chaotic sequence and the traditional Gold sequence are tested with the experimental system. The experimental results show that the CNN chaotic sequence is more qualified than Gold sequence when the channel is multi-path channel, and the security of the CNN wireless communication system is also reliable.

References

1. Proakis, J.G.: Digital Communications. McGraw-Hill (1995)
2. Chua, L.O., Yang, L.: Cellular Neural Network: Applications. IEEE Trans. Circuits Syst. 35(1988) 1257-1272
3. Kapitaniak, T., Chua, L.O.: Experiment Hyper-chaos in Coupled Chua's Circuits. IEEE Trans. Circuits Syst. 41 (1994) 499-503
4. Zou, F., Nossek, A.: Bifurcation and Chaos in Cellular Neural Networks. IEEE Trans. Circuits Syst. 40 (1993) 166-173
5. Grassi, G.: Synchronizing High Dimensional Chaotic System via Eigenvalue Placement with Application to CNN. International Journal of Bifurcation and Chaos 9 (1999) 705-711

A Resource Allocating Neural Network Based Approach for Detecting End-to-End Network Performance Anomaly*

Wenwei Li[1], Dafang Zhang[2], Jinmin Yang[2], Gaogang Xie[3], and Lei Wang[2]

[1] College of Computer and Communication, Hunan University,
Changsha 410082, China
liww@hnu.cn
[2] School of Software, Hunan University,
Changsha, 410082, China
{dfzhang, rj_jmyang}@hnu.cn
[3] Institute of Computing Technology, Chinese Academy of Science,
Beijing, 100080, China
xie@ict.ac.cn

Abstract. Automatic detection of end-to-end network performance anomalies is important to efficient network management and optimization. We present an end-to-end network performance anomalies detection method, based on characterizing of the dynamic statistical properties of RTT normality. The experiment on real Internet end-to-end path RTT data shows that, the proposed method is accurate in detecting performance anomalies, it can successfully detect about 96.25% anomalies in the experiment.

1 Introduction

The rapid growth of Internet throughout the workplace has given rise to a discontinuity in expertise of human operators to monitor and manage it. From the point of view of Internet Service Providers, performance management is one of the most important aspects of network management. A crucial problem in performance management is how to detect end-to-end network performance anomalies (degradation) proactively.

Currently, many works attempt to detect network performance anomalies by detecting network traffic anomalies based on such an assumption: the network traffic anomaly will lead to performance anomaly (degradation)[9,15,21]. However, the assumption is only partially true, as there is no inevitable contact between traffic anomaly and end-to-end network performance anomaly. Consequently, the traffic anomaly detection approach has large limitations in detecting end-to-end network performance anomaly. Different from previous researches, we detect end-to-end performance anomalies by using end-to-end path Round Trip Time (RTT) measurement.

* This work was supported by National Natural Science Foundation of China under grant 60273070, 60473031 and 60403031, the National High-Technology Program of China (863) under grant 2005AA121560, the Hunan Provincial Natural Science Foundation of China under grant 05JJ30116.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 184 – 189, 2006.

2 The RANN-Based Detection Method

To detect anomalies of the non-stationary signals, statistical methods and auto regression methods have been developed [2,18]. Considering the diversity of performance of different end-to-end paths, the selection of parameters in these methods is difficult. While the neural network is an ideal approach for solving non-stationary and nonlinear time series problems [16,17], it has been used in many areas such as face recognition [1,3], decision support [4,6,7], image process [8], data forecast [10,11], automatic control [13] and communication [14] etc. Many works have also done to improve the neural network method [19,20,22]. However, due to the time-variable characteristic of the non-stationary signals, attentions must be paid to the size selection of neural networks.

The authors of [5] introduced a Resource-Allocating Neural Network that allocates a new computational unit whenever an unusual pattern is presented to the network. This algorithm is suitable for sequential learning and is based on the idea that the number of hidden units should correspond to the complexity of the underlying function as reflected in the observed signals.

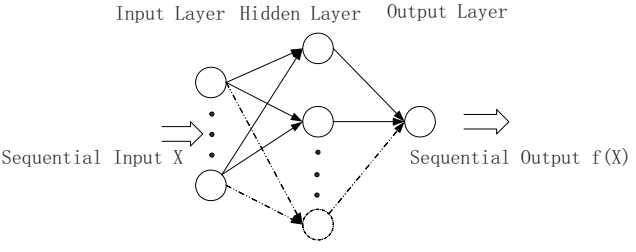


Fig. 1. Structure of the RANN

2.1 Basic Resource-Allocating Neural Network

As shown in Figure 1, the RANN can be simplified as a single hidden layer network whose output response to sequential inputs is a linear combination of the hidden unit responses. The output function can be expressed as follows:

$$f(X) = \alpha_0 + \sum_{i=1}^{I} \alpha_i \phi_i(X) \tag{1}$$

where X is an M-dimensional input vector, α_0 is the bias term, α_i are the weights between the hidden and the output layers, and parameter I is the hidden unit number. Each unit in the hidden layer is fully connected to the inputs and possesses a unit centre. In general, the nonlinear function is selected as a Gaussian function. The RAN hidden unit responses$\ddot{o}_i(X)$ are given by:

$$\phi_i(X) = \exp\{-\frac{1}{\sigma_i^2}\|X - \mu_i\|^2\} \tag{2}$$

where $ì_i$ is the mean of the Gaussian and $ó_i$ is the width of the Gaussian. In order to predict k steps ahead, we start the network without hidden units. The first observation is (X_0, X_k), where X_k is the target output, and in the initial stage the coefficient $\alpha_0 = X_k$. As observations are received the network grows by storing some of them and adding new hidden units. The decision to store an observation (X_n, X_{n+k}) depends on its novelty, as presented by the following two conditions:

$$\left\| X_n - \mu_{ni} \right\| > \varepsilon_n \text{ and } e_n = X_{n+k} - f(X_n) > e_{\min} \tag{3}$$

where $ì_{ni}$ is the nearest stored pattern to X_n in the input space, threshold $å_n$ represents the scale of resolution in the input space, and the value e_{min} is chosen to represent the desired accuracy of the network output. When a new hidden unit is added to the network, the parameters and weights associated with this unit are assigned as follows:

$$\alpha_{i+1} = e_n ;\ \mu_{i+1} = X_n ;\ \sigma_{i+1} = \varphi \left\| X_n - \mu_{ni} \right\| \tag{4}$$

where φ is the overlap factor that determines the overlap of the responses of the hidden units in the input space. The value for the width $ó_{i+1}$ is based on a nearest-neighbor heuristic.

2.2 Dynamic Thresholds Definition

Instead of detecting performance anomalies by a single RTT value that is predicted according to the RANN, a more proper approach is to find a specific interval of the predicted values that is highly likely to contain the actual value. This can be done by specifying some high probability, say $1-â$, and then posing the following problem: Find an interval $[L_{th}, U_{th}]$ such that $P[L_{th} \le X_{n+k} \le U_{th}] = 1-â$. We define the dynamic lower and upper thresholds respectively as L_{th} and U_{th}. For accurate anomaly detection, we define the new interval by using the probability of the absolute prediction errors. First, we define a threshold-sliding window that stores the prediction errors under normal network status. Let us denote the PDF of the random variable in the threshold-sliding window is $g(s)$. Then the interval $(-S, S)$ which gives confidence that the prediction errors will have probability $1-â$ can be calculated by the following expression:

$$\int_{-S}^{S} g(s)ds = 1 - \beta \tag{5}$$

Therefore the dynamic lower and upper thresholds can be defined respectively as follows:

$$L_{th} = f(X_n)/(1+S) ;\ \ U_{th} = f(X_n)/(1-S) . \tag{6}$$

2.3 Performance Anomaly Detection

The network performance anomalies may present different characteristics that vary both in amplitude as well as time duration; so we define the anomaly detection based on the above thresholds, as a combined function of both magnitude and duration. We also define an anomaly violation sliding window with size T $(0 \le T \le T_{ad})$, which will record the anomaly status if the RTT crosses the dynamic threshold. At last the anomaly violation condition is defined as follows:

$$\left|\sum_{t=0}^{T} E(t) * t\right| \geq \sum_{t=0}^{T} k\sigma(t) f(X) \tag{7}$$

$$E(t) = \begin{cases} X_{n+k} - U_{th} & if\ X_{n+k} \geq U_{th} \\ L_{th} - X_{n+k} & if\ X_{n+k} \leq L_{th} \\ 0 & otherwise \end{cases} \tag{8}$$

where $\sigma(t) = \sqrt{\text{var}[Z(t)]}$, $Z(t)$ is the data in the prediction error window at time t, k is a constant that may be determined experimentally. When the RTT crosses the dynamic threshold, the violation magnitude and time will be recorded and an anomaly detection alarm will be sent out if inequality (7) is satisfied.

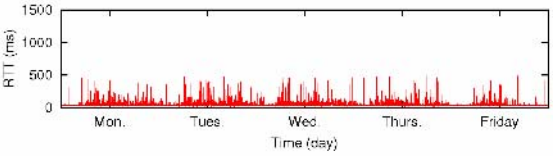


Fig. 2. Time-series plot of original path RTT in a week

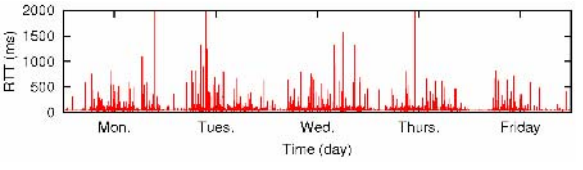


Fig. 3. Time-series plot of modified path RTT in a week

3 Experimental Results

We evaluate the algorithm proposed in this paper using the end-to-end path RTT data measured by AMP project [12]. The end-to-end path is from Boston University to Clemson University, and measuring time is from Sept. 12, 2005 to Sept.16, five weekdays in a week. The time series plot of original RTT data is shown in Figure 2. First, we train the RAN network with the original RTT data, the parameters in the network is chosen as follows:$\mathring{a}_{max}$=1,$\mathring{a}_{min}$=0.01,$\tilde{a}$=0.999, e_{min}=0.01, e'_{min}=0.1,η=0.001, T_{ad}=3,$\hat{a}$=0.03, Q_0=0.0003.

Then we randomly insert some very large RTT values into the original RTT data to simulate network performance anomalies. The time series plot of the modified RTT data is shown in Figure 3. At last, the trained RANN for detecting any performance anomaly checks the modified RTT data. The result is listed in Table 1. The detection rate of the RANN-based method is very high. In the experiment, it reaches 96.25%. And the anomaly detection algorithm has small False Accept Rate (FAR) and False Reject Rate (FRR).

Table 1. Results of the RANN-based performance anomaly detection method

Anomalies	Detection	Detection Rate(%)	FAR(%)	FRR(%)
80	85	96.25	5.88	3.53

4 Conclusions

In this paper, we use the RANN based RTT prediction and dynamic thresholds for RTT anomalies detection. The experiment with real Internet end-to-end path RTT data shows validity of the method. Our ongoing work is centered on extending the method proposed here to automatic detect anomalies of multiple end-to-end paths simultaneously, which is very important for large scale network performance monitoring and management.

References

1. Li, B. C., Yin, H. J.: Face Recognition Using RBF Neural Networks and Wavelet Transform. In: Wang, J., Liao, X. F., Yi, Z. (eds): Advances in Neural Networks-ISNN 2005, Part 2. Lecture Notes in Computer Science, Vol.3497. Springer-Verlag, Berlin Heidelberg New York (2005) 105-111
2. Ho, L., Cavuto, D., Papavassiliou, S.: Adaptive/Automated Detection of Service Anomalies in Transaction WANs: Network Analysis, Algorithms, Implementation, and Deployment. IEEE Journal on Selected Areas in Communications. 18(5) (2000) 744-757
3. Pu, X. R., Yi, Z., Zheng, Z. M., Zhou, W., Ye, M.: Face Recognition Using Fisher Non-negative Matrix Factorization with Sparseness Constraints. In: Wang, J., Liao, X. F., Yi, Z. (eds): Advances in Neural Networks-ISNN 2005, Part 2. Lecture Notes in Computer Science, Vol.3497. Springer-Verlag, Berlin Heidelberg New York (2005) 112-117
4. Sun, C. Y., Feng, C. B.: Neural Networks for Nonconvex Nonlinear Programming Problems: A Switching Control Approach. In: Wang, J., Liao, X. F., Yi, Z. (eds): Advances in Neural Networks-ISNN 2005, Part 1. Lecture Notes in Computer Science, Vol.3496. Springer-Verlag, Berlin Heidelberg New York (2005) 694-699
5. Platt, J.: A Resource-Allocating Network for Function Interpolation. Neural Computation, 3(2) (1991) 213-225
6. Hao, Z. F., Liu, B., Yang, X. W., Liang, Y. C., Zhao, F.: Twi-Map Support Vector Machine for Multi-classification Problems. In: Wang, J., Liao, X. F., Yi, Z. (eds): Advances in Neural Networks-ISNN 2005, Part 1. Lecture Notes in Computer Science, Vol.3496. Springer-Verlag, Berlin Heidelberg New York (2005) 869-874
7. Guo, C. G., Anthony, K.: An Optimal Neural-Network Model for Learning Posterior Probability Functions from Observations. In: Yin, F. L., Wang, J., Guo, C. G. (eds): Advances in Neural Networks-ISNN 2004, Part 1. Lecture Notes in Computer Science, Vol.3173. Springer-Verlag, Berlin Heidelberg New York (2004) 370-376
8. Lu, W., Lu, H. T., Chung, F. L.: Subsampling-Based Robust Watermarking Using Neural Network Detector. In: Wang, J., Liao, X. F., Yi, Z. (eds): Advances in Neural Networks-ISNN 2005, Part 2. Lecture Notes in Computer Science, Vol.3497. Springer-Verlag, Berlin Heidelberg New York (2005) 801-806
9. Manikopoulos, C., Papavassiliou, S.: Network Intrusion and Fault Detection: A Statistical Anomaly Approach. IEEE Communications Magazine, 40(10) (2002) 76–82

10. Han, M., Fan, M. M., Xi, J. H.: Study of Nonlinear Multivariate Time Series Prediction Based on Neural Networks. In: Wang, J., Liao, X. F., Yi, Z. (eds): Advances in Neural Networks-ISNN 2005, Part 2. Lecture Notes in Computer Science, Vol.3497. Springer-Verlag, Berlin Heidelberg New York (2005) 618-623
11. Yang, Y. Q., Cao, J. D., Zhu, D. Q.: A Study of Portfolio Investment Decision Method Based on Neural Network. In: Yin, F. L., Wang, J., Guo, C. G. (eds): Advances in Neural Networks-ISNN 2004, Part 2. Lecture Notes in Computer Science, Vol.3174. Springer-Verlag, Berlin Heidelberg New York (2004) 976-981
12. McGregor, A.., H-W. B: The NLANR Network Analysis Infrastructure. IEEE Communications Magazine, 38(5) (2000) 122-128
13. Sun, F. C., Zhang, H.: Neuro-Fuzzy Hybrid Position/Force Control for a Space Robot with Flexible Dual-Arms. In: Yin, F. L., Wang, J., Guo, C. G. (eds): Advances in Neural Networks-ISNN 2004, Part 2. Lecture Notes in Computer Science, Vol.3174. Springer-Verlag, Berlin Heidelberg New York (2004) 13-18
14. Liu, J., Yuan, D. F., Ci, S., Zhong, Y. J.: A New QoS Routing Optimal Algorithm in Mobile Ad Hoc Networks Based on Hopfield Neural Network. In: Wang, J., Liao, X. F., Yi, Z. (eds): Advances in Neural Networks-ISNN 2005, Part 3.Lecture Notes in Computer Science, Vol.3498. Springer-Verlag, Berlin Heidelberg New York (2005) 343-348
15. Li G., Patrick W.: An Analysis of Network Performance Degradation Induced by Workload Fluctuations. IEEE/ACM Transaction on Network, 3(4) (1995) 433-440
16. Wen, Y. M., Lu, B. L.: A Hierarchical and Parallel Method for Training Support Vector Machines. In: Wang, J., Liao, X. F., Yi, Z. (eds): Advances in Neural Networks-ISNN 2005, Part 1. Lecture Notes in Computer Science, Vol.3496. Springer-Verlag, Berlin Heidelberg New York (2005) 881-886
17. Li, S. W., Li, S. R.: Hopf Bifurcation in a Single Inertial Neuron Model with a Discrete Delay. In: Wang, J., Liao, X. F., Yi, Z. (eds): Advances in Neural Networks-ISNN 2005, Part 1. Lecture Notes in Computer Science, Vol.3496. Springer-Verlag, Berlin Heidelberg New York (2005) 327-332
18. Hajji, H., Far, B. H.: Continuous Network Monitoring for Fast Detection of Performance Problems. Proceedings of International Symposium on Performance Evaluation of Computer and Telecommunication Systems, 2001
19. Zhang, J., Liao, X. F., Li, C. D., Lu, A. W.: A Further Result for Exponential Stability of Neural Networks with Time-Varying Delays. In: Wang, J., Liao, X. F., Yi, Z. (eds): Advances in Neural Networks-ISNN 2005, Part 1. Lecture Notes in Computer Science, Vol.3496. Springer-Verlag, Berlin Heidelberg New York (2005) 120-125
20. Jiang, M. H., Shen, Y., Liu, M. Q.: Global Exponential Stability of Non-autonomous Neural Networks with Variable Delay. In: Wang, J., Liao, X. F., Yi, Z. (eds): Advances in Neural Networks-ISNN 2005, Part 1. Lecture Notes in Computer Science, Vol.3496. Springer-Verlag, Berlin Heidelberg New York (2005) 108-113
21. Juan, M., Garcia-Teodoro, P.: Anomaly Detection Methods in Wired Networks: A Survey and Taxonomy. Computer Communications, 27(3) (2004) 1569-1584
22. Zeng, Z. G., Chen, B. S., Wang, Z. F.: Globally Attractive Periodic State of Discrete-Time Cellular Neural Networks with Time-Varying Delays. In: Wang, J., Liao, X. F., Yi, Z. (eds): Advances in Neural Networks-ISNN 2005, Part 1. Lecture Notes in Computer Science, Vol.3496. Springer-Verlag, Berlin Heidelberg New York (2005) 282-287

Recurrent Neural Network Inference of Internal Delays in Nonstationary Data Network

Feng Qian, Guang-min Hu, Xing-miao Yao, and Le-min Li

Key Lab of Broadband Optical Fiber Transmission and Communication Networks,
University of Electronic Science and Technology of China,
Chengdu 610054, China
qianfeng1978@126.com,
{hgm, yxm, lml}@uestc.edu.cn

Abstract. By applying tomography theory which is highly developed in fields such as medical computerized tomography and seismic tomography to communication network, network tomography has become one of the focused new technologies, which can infer the internal performance of the network by external end-to-end measurement. In this paper, we propose a novel Inference algorithm based on the recurrent multilayer perceptron (RMLP) network capable of tracking nonstationary network behavior and estimating time-varying, internal delay characteristics. Simulation experiments demonstrate the performance of the RMLP network.

1 Introduction

To design, control and manage the network successfully and use its resource effectively, its internal characteristics, such as link delay, loss rate and topology, have to be understood and described accurately. By applying tomography theory which is highly developed in fields such as medical computerized tomography and seismic tomography [4] to communication network, network tomography [1] [2] [3] has become one of the focused new technologies, which can infer the internal performance of the network by external end-to-end measurement. Network tomography can be classified into stationary network tomography [1] [2] and nonstationary network tomography [3]. Stationary network tomography assumes that discrete probability mass functions (pmfs) are non-Time-vary over the measuring period, and maximum likelihood estimator can be adopted to infer internal delay distributions parameter. Literature [1] adopt EM(Expectation Maximization)Iteration to approximate maximum value of likelihood function, literature [2] adopts Multiscale Maximum Penalized Likelihood Estimator(MMPLE)algorithm to approach with a faster speed. Nonstationary network tomography regards pmfs of network queue delay as Time-vary, which better reflects the dynamic nature of the real network. Literature [3] adopts Sequential Monte Carlo (SMC) procedure to track time-vary average delay, and infers delay pmfs. But, bayesian filter performance of particle filter (SMC) and Kalman filter depends on prior information of inference model. Since it is hard to use mathematical model to define

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 190 – 195, 2006.

queue delay in real network, performance of inference in literature [3] will decrease or become instability. RMLP network [7] is a neural network with recursive connection, which make RMLP be able to memory and store, it is suitable to model nonlinear, complicated and dynamic nonstationary network queue delay dynamic performance system. Simultaneity, recurrent neural networks (RNNs) have shown better performance than feedforward neural networks (FNNs), since RNNs approximate infinite impulse response (IIR) filters while FNNs approximate finite impulse response (FIR) filters.

2 Nonstationary Network Delay Tomography

We collect measurements of the end-to-end delays from source 0 to receivers R in the network as shown in Fig 1, and we index the packet pair measurements by $m=1,\ldots,M$. For the mth packet pair measurement, let $y_1(m)$ and $y_2(m)$ denote the two end-to-end delays measured. The ordering 1 and 2 is completely arbitrary. D_k is the random delay that would be encountered by a packet attempting to traverse the link $(f(k),k) \in L$, $D_k \in R_+ \cup \{\infty\}$ .The value $D_k = \infty$ indicates that the packet is lost on the link. The cumulative delay experienced on the path from the root to a node is $Z_k = \sum_{j \succeq k} D_k$ .If $(i,j) \in R$, we shall define $y_1 = Z_i, y_2 = Z_j, Y = \{y_1(m), y_2(m)\}$.

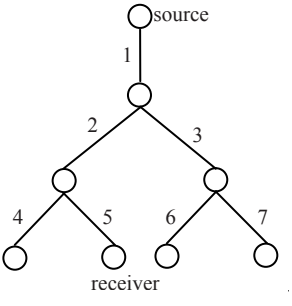


Fig. 1. Tree-structured network topology

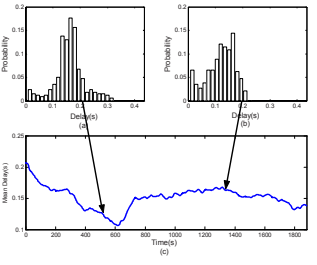


Fig. 2. Time-varying characteristic of link mean delay and delay pmfs

In stationary network, because of delay independence, for finite i, $A_k(i) = \sum_{j=0}^{i} \alpha_k(j) A_{f(k)}(i-j)$,$k \in V \backslash \{0\}$. Based on $y=\{y_1(m),y_2(m)\}$ is measured, we are interested in MLEs of $\alpha_{k,i}$, which is the collection of all delay pmfs. In nonstationary network [7], it is hard to define queue performance which changes with time as shown in figure 2. According to the above measurement model, we can define time-varying delay distribution to probe package m window R:

$$\alpha_{i,j}(R,m) = \frac{1}{R} \sum_{l=m-R+1}^{m} 1_{\{D_i(l)=1\}} \tag{1}$$

Where $D_i(l)$ is the delay which through probe package l experience link (which can not be observed) and $1_{\{z_i(l)=j\}}$ is the display function of the event. $\alpha_{i,R} \equiv \{p_{i,j}(R,m)\}$ is the delay time variety possibility of link i. in practice, the choice of R is based on known or assumed known dynamic of the network. We are interested in the average delay of each link as shown in figure 2(c), because average delay reflect time-varying characteristic. The average delay can be shown by the following equation:

$$\hat{\alpha}_{i,j}(R,m) \coloneqq E_{p(D_{i,m-R+1:m}|y_{1:m})}[\sum_{l=m-R+1}^{m} 1_{\{D_i(l)=j\}}]. \tag{2}$$

3 RMLP Network Estimator

3.1 RMLP Network Architectures

RMLP network framework is adopted in this study. A simple RMLP network is shown in figure 2. The basic form of RMLP network is that all nodes of one level either are recursively connected, or are not recursively connected. Recursive connection means that each node of a certain level is connected with other nodes of the same level by unit delay. These recursive connections make the network be able to memorize, being suitable for modeling dynamic system. Each processing unit of RMLP network can be shown by the following equation:

$$x_{[l,i]}(t) = \sum_{j=1}^{N_{[l]}} w_{[l,j][l,i]} u_{[l,j]}(t-1) + \sum_{j=1}^{N_{[l]}} w_{[l-1,j][l,i]} u_{[l-1,j]}(t) + b_{[l,i]} \tag{3}$$

$$y_{[l,i]}(t) = g_{[l,i]}(x_{[l,i]}(t)) \tag{4}$$

$x_{[l,i]}(t)$ is the internal status value at the node i of level l of time t; $y_{[l,i]}(t)$ is the output at the node i of level l of time t; the sum $b_{[l,i]}$ is the bias of the node, providing the bias of each node in the network. w is the weighed value which connects node i of level l to node j of level l'. Besides, t is discrete time, and the index of the node is $i = 1,....,N_{[l]}$, the index of the sum level is $l = 1,...,L$, $g_{[l,i]}(\cdot)$ is activation function of each node, which can be liner or nonlinear function.

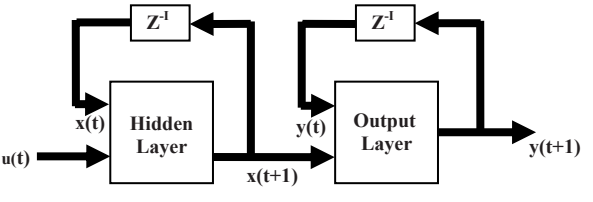


Fig. 3. Recurrent Multilayer Perceptron (RMLP)

The RMLP neural networks are used to construct single-step-ahead Estimators. The value of the inputs at the time t+1 step is predicted using time-series measurements up time t. In other words

$$y(t+1 \mid t) = f(U(t)) \tag{5}$$

where $U(m)$ represents time-lagged values of the time-series, outputs $y(\cdot)$, or transformations of the time-series, inputs $u(\cdot)$, up to time m . That is

$$U(t) \equiv [y(t),..., y(t-n_y+1), u(t),..., u(t-n_u+1)] \tag{6}$$

where n_y and n_u are the maximum number of lags in the outputs and inputs, respectively. The function $f(\cdot)$ can be approximated either by an RMLP.

3.2 Training Algorithms for RMLP Network

As like literature [7], Multi-stream DEKF algorithm [5] which is based on BPTT[6] is used to training RMLP, and it comprises the following five steps: initialization of network, introduction to training data, computation of network weighed values grads,network weighed values' updating and training's Iteration.

1) Initialization: Initialization of RMLP network includes network forward weights, network feedback weighed values and initialization of internal status of the network. They are set as small random value with normal distribution in a certain range.

2) Presentation of training data: Training data which we are interested in are whose period is 10 minutes to the path and link. It is generally inappropriate to select long data sequences to train a recurrent neural network. Each data sequence is truncated into many smaller segments for gradient calculation using standard truncated BPTT**.** A small portion of the training data at the beginning of each data sequence is used to develop network initial states rather than for training network weights (priming operation) [5].

3) Update of network weights: The network weight adjustments are computed by using multi-stream DEKF algorithm in which network weight directives are obtained using standard truncated BPTT as stated above.

4) Update of network weights: The network weight adjustments are computed by using multi-stream DEKF algorithm in which network weight directives are obtained using standard truncated BPTT as stated above.

5) Iteration: The new truncated training data is input into the network to compute RMS. Iteration will not stop until all the training data sequence's accumulated average RMS error reach the minimum or a value which is small enough to be accepted.

4 Simulation Result

1) Experimental Data: To evaluate the performance of RMLP network in a more realistic circumstance, ns2 simulation tools are adopted to simulate the topology shown in figure4 to obtain the data needed by RMLP network training and test.

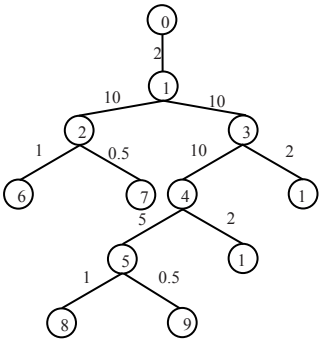


Fig. 4. Tree-structured network topology used for ns simulation experiments

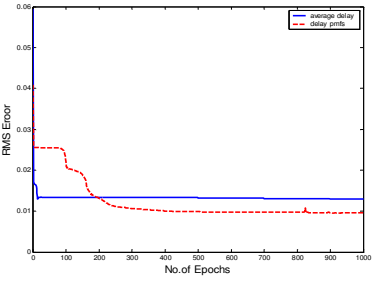


Fig. 5. Mean squared error of link delay pmfs and mean delay

2) Training: The structure of our trained RMLP network is 2-10R-10R-1L, i.e. two inputs, possessing two completely recursive hiding level and a linear output node. Multistream DEKF algorithm is set as follows: node-decoupled EKF is adopted. The number of streams is 10, priming length is 10, and trajectory length is also 200. A truncation depth of BPTT algorithm is 10. Figure 5 shows typical RMS error curves for training the RMLP for prediction of delay pmfs and average delay respectively.

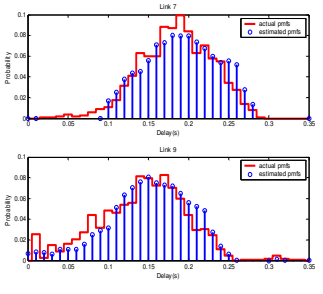


Fig. 6. Delay pmf estimates in nonstationary scenario using RMLP network

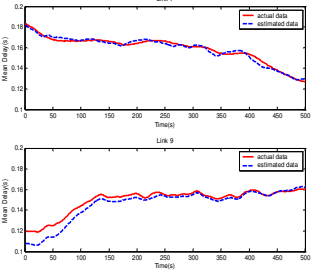


Fig. 7. Tracking of average delay on links 7 and 9 over measurement period

3) Testing: To test performance of the network, link and path data produced by different simulating periods are input. Figure.6 depicts the delay distribution estimates from typical ns simulations. and Figure.7 depicts the mean tracks of the ns experiment. The time-averaged, normalized squared error: $\sum_i |m_i - \hat{m}_i|^2 / m_i^2$, m_i is true value at time i and $\hat{m}_i$ is estimate. Performance comparison of three algorithms for inference delay pmfs and tracking average delay are listed in Table 1 and Table 2, respectively.

Table 1. Performance comparison of three algorithms for inference delay pmfs

	RMLP	SMC	ML-EM
Link7	0.0650	0.4200	1.1500
Link9	0.0591	0.1500	0.5000

Table 2. Performance comparison of three algorithms for tracking average delay

	RMLP	SMC	ML-EM
Link7	0.0094	0.0400	0.1100
Link9	0.0013	0.0400	0.0700

5 Conclusions

Our experiments demonstrate the potential of RMLP network for network delay tomography. We find that very good estimates of the delay pmfs can be obtained from a small number of measurements, and estimates of average delays are very robust, The RMLP network appears to track network behavior reasonably well, as demonstrated by the realistic ns experiments. In addition, recently, such neural network predictors have been implemented on a fixed-point arithmetic digital signal processor (DSP), and their real-time operation has been experimentally verified at video source traffic prediction application [8].Based on this work, we will develop real-time RMLP network estimators for nonstationary network delay tomography.

Acknowledgement

This work is supported by National Science Foundation of China (No.60572092) and Sichuan Youth Science and Technology Foundation (No.04ZQ026-028).

References

1. Liang, G., Yu, B.: Maximum Pseudo Likelihood Estimation in Network Tomography. IEEE Trans. Signal Processing 51 (2003) 2043-2053
2. Tsang, Y., Coates, M. (ed.): Network Delay Tomography. IEEE Trans. Signal Processing 51 (2003) 2125-2136
3. Coates, M.J., Nowak, R.D.: Sequential Monte Carlo Inference of Internal Delays in Nonstationary Data Networks. IEEE Trans. Signal Processing, Vol. 50, No. 2, (2002) 366-376.
4. 4. Li, Y., Lei, X. (ed.): Gas-water Recognition in Carbonate Rock Fracture Systems by use of Neural Network, Tianranqi Gongye/Natural Gas Industry, Vol 20 (2000) 32-36
5. Feldkamp, L. A., Puskorius, G. V.: Training Controller for Robustness: Multi-stream DEKF. IEEE Internationa1 Conference on Neural Network, (1994) 2377-2382
6. Werbos, P. J.: Backpropagation Through Time: What It Does and How To Do it. Proc. IEEE 78 (1990) 1550-1560
7. Li, S: Wind Power Prediction Using Recurrent Multilayer Perceptron Neural Networks. IEEE Power Engineering Society General Meeting, July 2003, Vol. 4, 13-17
8. Harihara, P.: Real-time Implementation of a Neural Networks-Based Motor Speed Filter Using a Digital Signal Processor. M.S. thesis, Texas A&M Univ, TX (2002)

Multiscale BiLinear Recurrent Neural Networks and Their Application to the Long-Term Prediction of Network Traffic

Dong-Chul Park[1], Chung Nguyen Tran[1], and Yunsik Lee[2]

[1] ICRL, Dept. of Information Engineering, Myong Ji University, Korea
{parkd, tnchung}@mju.ac.kr
[2] SoC Research Center, Korea Electronics Tech. Inst., Seongnam, Korea
leeys@keti.re.kr

Abstract. A new wavelet-based neural network architecture employing the BiLinear Recurrent Neural Network (BLRNN) for time-series prediction is proposed in this paper. It is called the Multiscale BiLinear Recurrent Neural Network (M-BLRNN). The wavelet transform is employed to decompose the time-series to a multiresolution representation while the BLRNN model is used to predict a signal at each level of resolution. The proposed M-BLRNN algorithm is applied to the long-term prediction of network traffic. The performance of the proposed M-BLRNN algorithm is evaluated and compared with the traditional MultiLayer Perceptron Type Neural Network (MLPNN) and the BLRNN. The results show that the M-BLRNN gives a 20.8% to 76.5% reduction in terms of the normalized mean square error (NMSE) over the MLPNN and the BLRNN.

1 Introduction

In recent years, various approaches have been proposed for time-series prediction problems. These approaches have been traditionally performed by using linear models and nonlinear models. Classical linear models such as the autoregressive (AR) [1] and autoregressive moving average (ARMA) [2] models have been widely used in practice. Another promising nonlinear model of time-series prediction is the neural network(NN)-based model [3]. As shown in a wide range of engineering applications, the NN-based models have been used successfully for modeling complex nonlinear systems and for forecasting signals.

In this paper, we examine a multiscale neural network architecture for the long-term prediction of a time-series. The multiscale architecture is based on a novel technology developed in the signal processing community called the wavelet transform[4]. The wavelet transform is employed to decompose a signal into a multiresolution representation. By doing this, it provides many important correlation structures, which are hidden in the original signal. The separated BLRNN [5] model which has been proven to be very efficient in modelling a time-series [5, 6, 7] is used to predict a signal at each level of resolution.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 196–201, 2006.

2 Multiresolution Analysis with Wavelet Transform

The aim of multiresolution analysis is to analyze a signal at different frequencies with different resolutions. It produces a high quality local representation of a signal in both the time domain and the frequency domain. Therefore, the wavelet transform is advantageous when dealing with a non-stationary signal like that in time-series data. In order to deal with the time-series prediction, a redundant Haar wavelet transform was proposed. The redundant Haar wavelet transform is based on an *à* trous wavelet transform [8] in which the wavelet scale has the same number of samples as the original signal to provide visually or graphically related information at different scales.

The *à* trous wavelet transform was first proposed by Shensa [8]. In order to calculate the *à* trous wavelet transform, a lowpass filter is used to suppress the high frequency components of a signal and allow the low frequency components to pass through. Then, the scaling function associated with the lowpass filter is used to calculate the average of elements, which results in a smoother signal. The smoothed data $c_j(t)$ at given resolution j, can be obtained by performing successive convolutions with the discrete low-pass filter h,

$$c_j(t) = \sum_k h(k) c_{j-1}(t + 2^{(j-1)}k) \tag{1}$$

where h is a discrete low-pass filter associated with the scaling function and $c_0(t)$ is the original signal. A suitable low-pass filter h is the B_3 spline, defined as $(\frac{1}{16}, \frac{1}{4}, \frac{3}{8}, \frac{1}{4}, \frac{1}{16})$.

From the sequence of the smoothing of the signal, the wavelet coefficients are obtained by calculating the difference between successive smoothed versions:

$$w_j(t) = c_{j-1}(t) - c_j(t) \tag{2}$$

By consequently expending the original signal from the coarsest scale to the finest scale, the original signal can be expressed in terms of the wavelet coefficients and the scaling coefficients as follow:

$$c_0(t) = c_J(t) + \sum_{j=1}^{J} w_j(t) \tag{3}$$

where J is the number of resolutions and $c_J(t)$ is the finest version of the signal. Eq.(3) also provides a reconstruction formula for the original signal.

The discrete *à* trous wavelet transform is not appropriate for a time-series prediction because it uses a symmetric low-pass filter which requires a future data value in the calculation of the wavelet transform. In order to deal with this problem, the redundant Haar wavelet transform is proposed. The redundant Haar wavelet transform is precisely the same as the *à* trous wavelet transform except that in the redundant Haar wavelet transform, the B_3 spline filter is replaced by a simple nonsymmetric filter $(\frac{1}{2}, \frac{1}{2})$.

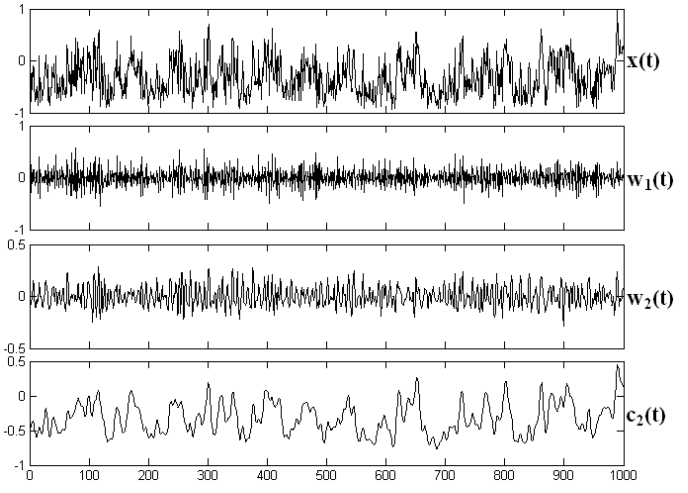


Fig. 1. Example of the wavelet coefficients and the scaling coefficients

The smoothed data $c_j(t)$ or scaling coefficients are calculated as follow:

$$c_j(t) = \frac{1}{2}(c_{j-1}(t) + c_{j-1}(t - 2^{(j-1)})) \tag{4}$$

while the wavelet coefficients are calculated by Eq. (2).

Fig. 1 shows an example of the wavelet coefficients and the scaling coefficients for two levels of resolution over 1,000 samples from a Ethernet network traffic data set. From the top to the bottom are the original signal, two levels of the wavelet coefficients, and the finest scaling coefficients, respectively. As the wavelet level increases, the corresponding wavelet coefficients become smoother.

3 Multiscale BiLinear Recurrent Neural Network

The BLRNN [5] is a simple recurrent neural network, which has a robust ability in modelling nonlinear systems and is especially suitable for time-series data. It has been successfully applied in modelling time-series data [5, 6, 7]. The multiscale BLRNN is a combination of several BLRNN models in which each model is employed to predict the signal at each level of resolution. An example of the multiscale BLRNN-based prediction scheme with three resolution levels is shown in Fig. 2. In this scheme, the prediction of a time-series is separated into three stages. In the first stage, the original signal is decomposed into the wavelet coefficients and the scaling coefficients based on the number of resolution levels. In the second stage, each scale is predicted by a separated BLRNN model using the wavelet coefficients and the scaling coefficients at each resolution level as the inputs. In the third stage, the final prediction result is obtained by reconstructing the prediction output from each BLRNN model using Eq.(3):

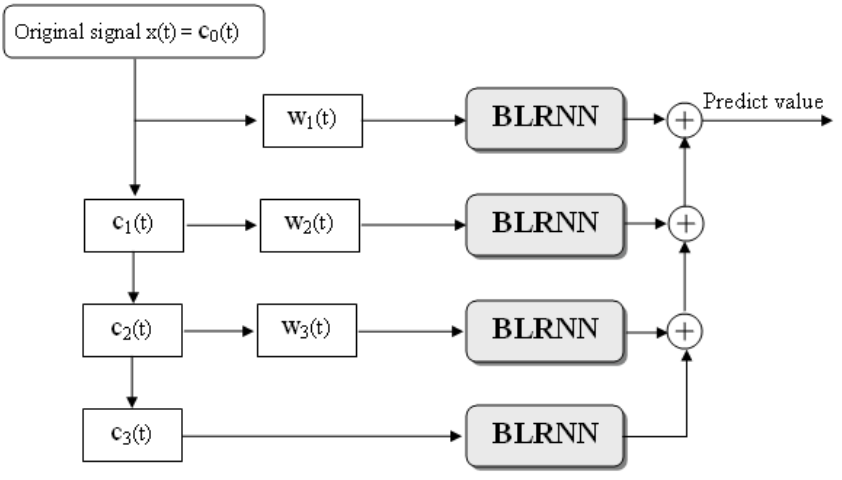


Fig. 2. Prediction scheme of M-BLRNN with 3 resolution levels

$$\hat{x}(t) = \hat{c}_J(t) + \sum_{j=1}^{J} \hat{w}_j(t) \tag{5}$$

where $\hat{c}_J(t)$ represents the predicted values of the finest scaling coefficients, $\hat{w}_j(t)$ represents the predicted values of the wavelet coefficients at level j, and $\hat{x}(t)$ represents the predicted values of the time-series. By adopting the BLRNN in the multiscale architecture, the M-BLRNN can learn the correlation structure and the internal information at each level of resolution hidden in the original signal because the original signal is decomposed into several simpler sub-signals with the form of wavelet coefficients. At each level of resolution, the BLRNN can learn the correlation structure and the internal information of the original signal more efficiently than other neural network architectures such as the MLPNN.

4 Experiments and Results

In this section, we demonstrate the generalization performance and robustness of the proposed M-BLRNN when applied to make a network traffic prediction. Real-world Ethernet traffic data sets collected at Bellcore in August 1989 [9, 10] are used to assess the performance of the proposed M-BLRNN. The above mentioned traffic data is one of the most popular benchmark data that has been used for evaluating network traffic predictors [11]. In this paper, the normalized mean square error (NMSE) [11] is employed to measure the prediction performance of the proposed algorithm.

The Ethernet traffic data sets used in experiments is the network traffic data (i.e., the number of bytes transferred on the network per time unit), which are collected at each 0.01(s) over two normal hours of traffic. They correspond to 10^6 samples of data. The data are then downsampled with the time scale 1(s) and

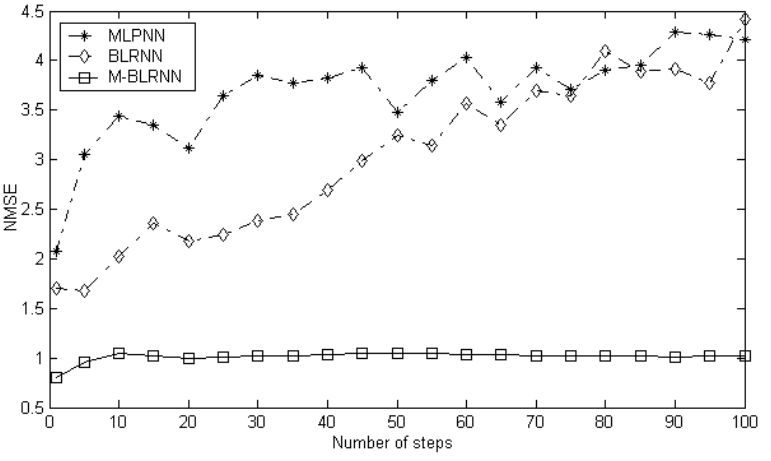


Fig. 3. Prediction performance for Ethernet network traffic data

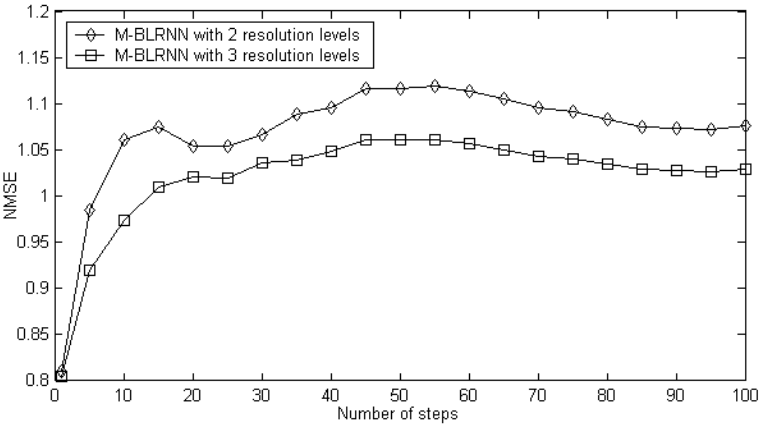


Fig. 4. Prediction performance with different resolutions

are normalized in range of $(-1, 1)$, which is the range of the logistic function. The first 1,000 samples are used for training the proposed M-BLRNN and the other conventional algorithms while the remaining data are used for testing.

Fig. 3 shows the prediction performance versus number of steps for the traditional MLPNN and the BLRNN, and the proposed M-BLRNN in terms of the NMSE. As can be seen from Fig. 3, the proposed algorithm outperforms both the traditional MLPNN and the BLRNN. In particular, the proposed M-BLRNN can successfully predict into the hundred of steps with no degradation in the predictor's performance. The traditional MLPNN and the BLRNN can not do this.

Fig. 4 shows the prediction performance versus number of steps for two different resolution levels of the proposed M-BLRNN. As can be seen from these

results, the prediction performance of the proposed M-BLRNN with three resolution levels is better than that with two resolution levels. In other words, the performance can be improved by increasing the number of resolution levels.

5 Conclusion

In this paper, we have presented a wavelet-based neural network architecture for a time-series prediction. The new architecture is formulated by a combination of the wavelet transform and the separated BLRNN models. The proposed M-BLRNN is applied in the prediction of a complex and challenging network traffic data. The successful prediction, into the hundreds of steps, provides sufficient time intervals for a dynamic bandwidth allocation. Directions for further study of the proposed M-BLRNN include investigating its robustness in modeling a time-series and investigating its application to other practical problems.

References

1. Suzuki, Y.: Prediction of Daily Traffic Volumes by Using Autoregressive Models. Proc. IEEE Conf. on Vehicular Technology 3 (2000) 2103-2107
2. Basu, S., Mukherjee, A., Klivansky, S.: Time Series Models for Internet Traffic. IEEE Joint Conf. on Computer Societies 2 (1996) 611-620
3. Connor, J., Martin, R., Altas, L.: Recurrent Neural Networks and Robust Time Series Prediction. IEEE Trans. on Neural Networks 5(2) (1994) 240-254
4. Mallat, S.G.: A Theory for Multiresolution Signal Decomposition: the Wavelet Representation. IEEE Trans. Pattern Anal. Machine Intell. 11(7) (1989) 674-693
5. Park,D.C.,Yan Zhu: Bilinear Recurrent Neural Network. IEEE ICNN. 3 (1994) 1459-1464
6. Park, D.C., Jeong, T.K.: Complex Bilinear Recurrent Neural Network for Equalization of a Satellite Channel. IEEE Trans. on Neural Networks 13(3) (2002) 711-725
7. Park, D.C., Nguyen, D.H., Hong, S.J., Lee, Y.: Equalization of a Wireless ATM Channel with Simplified Complex BiLinear Reccurent Neural Network. In: Wang, L., Chen, K., Ong, Y.(eds.): ICNC 2005, LNCS Vol. 3610. Springer-Verlag, Berlin Heidelberg New York (2005) 1113-1116
8. Shensa, M.J.: The Discrete Wavelet Transform: Wedding the Trous and Mallat Algorithms. IEEE Trans. on Signal Proc. 40(10) (1992) 2463-2482
9. Leland, W.E., Wilson, D.V.: High Time-Resolution Measurement and Analysis of LAN Traffic: Implications for LAN Interconnection. Proc. IEEE INFOCOM. (1991) 1360-1366
10. Fowler, H.J., Leland, W.E.: Local Area Network Traffic Characteristics, with Implications for Broadband Network Congestion Management. IEEE JSAC. (1991) 1139-1149
11. Alarcon-Aquino, V., Barria, J.A.: Multiresolution FIR Neural-Network-Based Learning Algorithm Applied to Network Traffic Prediction. IEEE Trans. on Sys. Man. and Cyber. PP(99) (2005) 1-13

Bandwidth Prediction and Congestion Control for ABR Traffic Based on Neural Networks

Zhixin Liu, Xinping Guan, and Huihua Wu

Institute of Electrical Engineering, Yanshan Univesity,
Qinhuangdao 066004, Hebei, China
{lzxauto, xpguan}@ysu.edu.cn

Abstract. In this paper, queue length in the buffer is considered as a scale to evaluate the quality of service of communication network. To realize that the system is stable and avoid the data congestion under complex environments, the neural networks predictor and controller are designed, which can predict the bursty available bandwidth for ABR traffic effectively and force the queue level in the buffer to the desired region, respectively. The fairness of different connections is achieved through fair algorithm. The simulations verify the effectiveness of the proposed scheme.

1 Introduction

In Asynchronous Transfer Mode (ATM) Networks, ABR service is the only one that can adopt a feedback mechanism to adjust the sending rate of sources according to the variable network environment. The feedback is realized by a special cell-Resource Management Cell (RM Cell). The source sends RM cells periodically and the destination returns them to source after it received them [1].

Since the priority of ABR is lower than VBR, CBR services, ABR can only use the remained bandwidth after the VBR, CBR utilize enough resources. In order to utilize as much bandwidth as possible and avoid loss of data, the congestion control of ABR is necessary. A number of rate-based ABR traffic management schemes have been developed in literatures. Among these, the ERICAA and its improved scheme, proposed in 1995[2] and 2000[3], are practical for its simplicity. Recently several schemes, such as Smith principle, PID control, inverse adaptive control, were presented to deal with the congestion problems [4-6]. For the network conditions are changing all the time, researchers proposed some methods to predict the traffic of ABR or VBR to get more accurate information of network load. Internet traffic was modeled as Morkov, Bossion or ARMA process. Recently, researchers found networks traffic had long-range dependence character, and a FARIMA model is used to model the traffic flow.

For these models cannot describe the character of network traffic accurately, the intelligent methods were introduced to solve the prediction and control. Literature [7] presented a method to predict the VBR traffic using Fuzzy-Neural networks, literature [8] gave a controller based on minimal resource allocation neural networks, which was well suit for on-line adaptive control of Internet.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 202 – 207, 2006.

In this paper, a traffic predictor and congestion controller are proposed based on Back Propagation (BP) neural networks, With the existence of constant traffic and VBR traffic conditions, the better performance of prediction and control are achieved.

2 Network Model

ATM network is a class of connection-oriented virtual circuit switching networks. Typical transfer process in ATM networks is shown in Fig.1. Before two users on a network can communicate each other, they should inform all intermediate switches about their service requirements and traffic parameters by establishing a virtual circuit. Many circuits can share network resources via store-and-forward packet switching and statistical multiplexing. Along a connection, there may be lots of switches. For simplicity, only one switch is considered, which is the bottlenecked node of the connection.

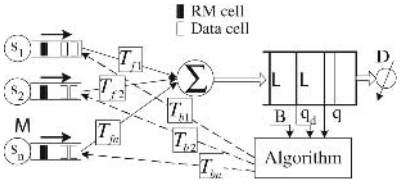


Fig. 1. ATM networks date transmission

The round trip time of the *i*th virtual connection, denoted by T_i, is the sum of forward delay T_{fi} and the backward delay T_{bi}. It consists of propagation, queuing, transmission and processing time. We assume that it is time-varying but bounded parameter, whose upper bound is $T_{\max} := \max\{T_i\}$. We can get the dynamic model.

$$q(k+1) = \mathrm{Sat}_B\left[q(k) + R(k)\cdot T - D(k)\cdot T\right] \tag{1}$$

Where $\mathrm{Sat}_B\{x\}$ is a saturation function, and its lower and upper bound are 0 and B, respectively. In Eq.(1), $q(k)$ and $R(k)$ are respectively the queue length and whole sending rate of VCs at time k, $D(k)$ is the available bandwidth for the ABR service, and B is the buffer size. Then, the open loop system can be represented by

$$q(k+1) = \mathrm{Sat}_B\left\{q(k) + \sum_{i=1}^{d} l_i(k)\, r(k-i) - D(k)\cdot T\right\} \tag{2}$$

where $r(k)$ is the control law and can be written in the ER field in the backward RM cell during the next period and $d = \mathrm{int}\{T_{\max}/T\}$. Here the queue length is viewed as a scale to evaluate the performance of network. If the value of $q(k)$ can be kept at a constant level or in a small range near the desired value, we can say that the avail-

able resource is utilized sufficiently and there is no cell lost. From eq.(2), we can see that the system is a nonlinear and discrete model.

3 Design of Flow Management Scheme

First we consider the structure of the predictor and controller. Specific knowledge of the statistic behavior of input traffic is necessary for every control scheme. The more information of traffic is known, the better performance may be achieved. The traditional model, either short-range dependence process or random process, cannot accurately characterize vary rapid changes in the bit rate variations of the traffic over short time intervals. Even ones can get the predictive value, it may be far away from the real value. So, we propose the predictive method based on neural network. The neural network used here is BP network. Given the past data, the neural network predictor (NNP) can predict the dynamic change in a next step, even several steps. This is to say the neural network can capture the unknown complicated relationship between past and future cell arrival rate of traffic. The architecture of the predictor is shown in Fig.2.

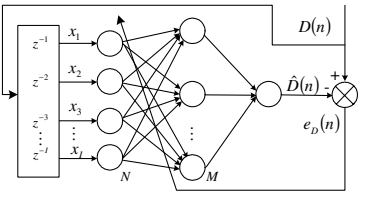


Fig. 2. The architecture of the predictor

Where

$x_i(i=1,2,\cdots,I)$ the arrival rates at time $(n-i)$ respectively, i.e. $x(n-i)$.

$D(n)$ the available bandwidth at time n.

$\hat{D}(n)$ the predictive value of arrival rate at time n.

$e_D(n)$ the error between $D(n)$ and $\hat{D}(n)$ at time n.

N, M the neuron numbers of input layer and hidden layer, respectively.

The active function of the neuron is chosen as Sigmoid function, given by $F[x_i(t)]=\left(1+e^{-x_i(t)}\right)^{-1}$. It need be pointed out that the number I decides the capacity of input information, and it is clear that the larger I is, the more the collected information of past time will be, but it results in the complication of network structure. While, the selection of the number of hidden layer neuron is an opening problem, and it is chosen according to practice.

The structure of congestion controller is similar to the above description. It is a single hidden layer BP neural network. The inputs are predictive available bandwidth, queue length, control law and their stored values in the past time. The block diagram of the complete system is shown in Fig. 3. According to the experiments, the parameters are chosen as shown in Table 1.

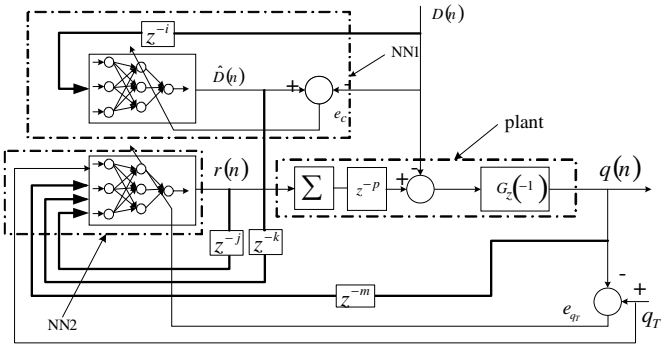


Fig. 3. The block diagram of the system

Table 1. The parameters of the predictive and control neural networks

	NN1	NN2
Parameters	I=4	J=2,K=1,M=2
Architecture	N=4,M=16	N=6,M=8

Next we consider the fairness of different connections. MAX-MIN fairness is a frequently adopted fairness criterion. For a VC with one bottleneck link, its bandwidth allocation is limited by the amount allocated in the bottleneck link. So the VCs bottlenecked at the same link have an equal amount of bandwidth allocation. Our proposed method is based on the framework of ER feedback mechanism. The RM cell is sent from the source every N_{rm} cell with the initial ER value. When it passes by a switch, the associated controller computes the available resource for each source.

We assume that there are K connections bottlenecked here totally. The sending rate $r(n)$ calculated by controller need satisfy the following constraint.

$$r_s(n) = \mathrm{Sat}_{MCR,PCR}\{r(n)\}$$

where, MCR and PCR are lower and upper bound of sending rate.

For each connection bottlenecked here, the absolute/ideal fair rate is

$$r_{fair}(n) = r_s(n)/K$$

Since the existence of estimated error during above phase, the calculated rate may be larger than the ER value stored in the received RM. Then the bottleneck link is not located here. The ER value cannot be changed. In this phase, when the switch receives the backward RM(BRM) cell, the rule of modifying ER field is

$$ER(n) = \min\{ER, r_{fair}(n)\}$$

Thus we get dynamic fair rate allocated to each VC.

4 Simulations

The network topology under consideration here is shown in Fig.4. We assume that the VBR represents the traffic with high priority. It is clear that, for source S_1,S_2,S_3, L_1 is the only bottleneck link.

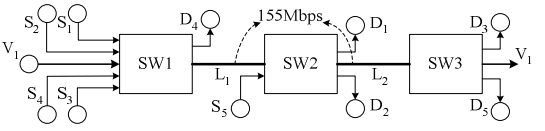
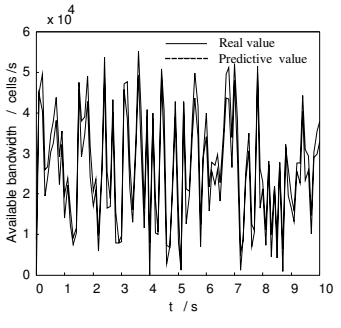


Fig. 4. The topology of ATM networks in simulation

First, we analyze the performance of predictor. Consider a typical kind of traffic flow, Fractional Auto-Regressive Integrated Moving Average process, which is detailed in literature [9]. Now we use the neural network predictor mentioned above to predict the data flow, which has different feature of long-range. Here we choose d=0.2, to examine the approach ability. The results of prediction are shown in Fig.5(a). We see that the error between the predictive value and the FARIMA(p,d,q) series is small and acceptable.



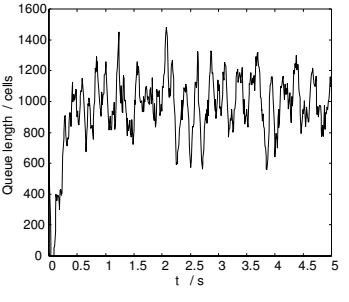


Fig. 5. (a) The predictive result with d=0.2 (b) The queue length of switch 1

Next, we analyze the control performance. The FARIMA process (d=0.2) is viewed as the background traffic. The driven random time series is $15000+N(0,700^2)$. The maximum transfer delay is 50ms, which is in accordance with a wide area networks, and the desired queue length is 1000cell, i.e. $\tau_{\max}=50ms$, $Q_D=1000$ cell. Link 1 is the bottlenecked link for source 1. The queue length at switch 1 is shown in Fig.5(b). The queue length has a certain wave, but it does not overflow the buffer. We can say that the system is stable, and there is nearly no cell lost. It is to be pointed out that the violent oscillation of background traffic or too small buffer size will result the data lost. Once the buffer size and desired queue length are determined according to the practical networks environment, the better quality of service can be gained.

5 Conclusions

The BP neural networks are used in congestion control of ABR service in ATM networks. The predictor and controller are designed to achieve desired performance. The intelligent control method can reduce the response caused by disadvantageous factor, such as varied round trip time, saturation nonlinear feature. The fairness of VCs is achieved using the fair algorithm further. The simulations show the effectiveness of the proposed scheme.

Acknowledgements

The authors grateful acknowledge the financial support of the Outstanding Youth Scientific Foundation of P.R. China under Grant 60525303, the National Science Foundation of P.R. China under Grants 60404022.

References

1. ATM From Management Specification 4.1, AF-TM-0121.0000 (1999)
2. Jain, R., Kalyanaraman, S., Viswanathan, R., Goyal, R.: A Sample Switch Algorithm. ATM Forum/95-0178 (1995)
3. Kalyanaraman, J. R., Fahmy, S.: The ERICA Switch Algorithm for ABR Traffic Management in ATM Networks, IEEE/ACM Trans. on Net. 8(1) (2000) 87-98
4. Tan, L.S., Yin, M.: A Rate-Based PI Congestion Controller for High-Speed Computer Communication Networks, Acta Electron. Sinica 30(8) (2002) 1138-1141
5. Mascolo, S.: Congestion Control in High-Speed Communication Networks Using the Smith's Principle. Automatica 35(12) (1999) 1921-1935
6. Laberteaux, K. P., Charles, E. R., Panos, J. A.: A Practical Controller for Explicit Rate Congestion Control. IEEE Trans. on Auto. Contr. 47(6) (2002) 960-977
7. Fei, X., He, X.: Fuzzy Neural Network Based Traffic Prediction and Congestion Control in High-Speed Networks. J. Comput. Sci & Tech. 15(3) (2000) 144-149
8. Soon, N. H., Sundararajan, N., Saratchandran, P.: ABR Traffic Management Using Minimal Resource Allocation (Neural) Networks. Comp. Com. 25(1) (2002) 9-20
9. Shu, Y.T., Wang, L., Zhang, L.F., *et al.*: Internet Traffic Modeling and Prediction Using FARIMA Models. Chin. J. of Comp. 24(1) (2001) 46-54

Application of Neural Networks in Network Control and Information Security

Ángel Grediaga[1], Francisco Ibarra[1], Federico García[2], Bernardo Ledesma[1], and Francisco Brotóns[1]

[1] Alicante University, P.O. 99 Alicante, E-03080, Spain
angel.grediaga@ua.es, ibarra@dtic.ua.es
http://www.ua.es/tia/
[2] Miguel Hernández University, Avda. de la Universidad, s/n. Elche E-03202, Spain
fedeg@umh.es

Abstract. The increment of intrusions and bad uses in computer systems and internal networks of a great number of companies has caused an increase in concern for computer security. For some time one comes applying measures based on fire walls and in systems of intrusion detection (IDS). In this document we present an alternative to the problem of the IDS based on rules, using two different neural networks, a Multi-Layer perceptron, and a self organizative map. A series of experiments are carried out and the results are shown to be better than others found in the literature.

Keywords: intrusion detection system, neural networks, self organizative map, perceptron.

1 Introduction

Security in computers has been studied as a discipline since 1970. It refers to the measures and controls that protect the information systems against the negation of service and the authorisation absence (accidental or deliberately) to reveal, to modify, or to destroy the information and data systems [1][2]. However, the scientific community dedicated to the computer system security, has noted that technical prevention is not enough to protect the systems, such and like it was proven in the year 2000 with the socalled Distributed attacks Denial of Service (DDoS) [3]. These incidents demonstrated that technical prevention is inadequate, although the true reason is that the developed systems of information are not certain, since they can have cracks in the implementation, as well as shortcomings in the design. Intrusion detection was proposed as a complement of technical prevention.

2 Experimentation Proposal

The proposal to palliate the main limitation of the traditional IDS and their inability to recognize attacks lightly modified regarding the patterns with those

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 208–213, 2006.

that carry out the comparisons, is the use of neural networks. By means of this proposal it is sought to prove that by means of using a neural network, it is possible to carry out the distinction among packages that represent normal flow in a computer network, and packages that represent attacks. Concretely we will carry out the experimentation with a ML perceptron, and with a self organizative map (SOM) [6] as for the type of IDS that we will implement using neural networks, it will be an IDS host.

2.1 Selection of the Data

The first thing that it is necessary to think about when training a neural network is what data will contain the samples that it uses as entrances, so much to train as for checking. This belongs since the most important decision that can betaken, will depend in great measure the distinction capacity that can acquire the neural network works once the training is concluded. For the election of these data, they have taken as reference the data that a traditional IDS uses, concretely 'Snort ', the one which, apart from using numerous data characteristic of the packages TCP, UDP, or ICMP, it also considers the content of the very package. This is since something that outlines a serious problem, in principle that this was the most important fact to distinguish among a dangerous inoffensive and other package, but it is very complicated to introduce the content of the data of the package in a neural network. Finally we have opted to take out four characteristics of the same one in statistical form of probability of the four more frequent characters that appear in it. The data used is the Table 1. In total there are 29 entrances, which have been introduced to the network for each package, but keeping in mind that when the package belongs to a protocol, the data of the other protocols will be represented as nonexistent (for example introducing the fact 0, or any other value that suits). When introducing the previous data to the neural network, they have been normalized between 0 and 1 choosing a maximum value for each item.

2.2 Obtaining of the Data

To obtain the dangerous packages two machines are used, one from which the attacks rushed using the scanner "Nessus" and another from which those packages were captured using the traditional IDS "Snort". In total 433 dangerous packages were obtained. To obtain the inoffensive packages, a real computer network is used, that is to say, where the habitual packages flow in any company or corporation where it seeks to settle an IDS. Finally the study has been carried out using a small departmental net. The fact that the IP address origin and the destination are the same one in all the packages doesn't rebound in the obtained results, since that information isn't kept in mind. Between Windows and Linux 5731 inoffensive packages were obtained. Next the two neural networks, the ML perceptron and the self organizative map, are explained. It is necessary to indicate that because the inoffensive training packages are much bigger in number that the dangerous training packages, when training both neural networks, it

Table 1. Data used

Head IP of the package:	**Head TCP of the package:**
Port origin	Flag 1
Port destination	Flag 2
Protocol	Flag OR
TOS	Flag TO
Size of the head IP	Flag P
Total size of the package	Flag R
Reserved Bit	Flag S
Don't Fragment bit	Flag F
Fragments bit live	Size of the window
Number of options IP	Size head TCP
	Number of options TCP
Head UDP of the package:	**Content Package:**
Size of the head UDP + data (Len)	Percentage of the most frequent datos1
	Percentage of the most frequent datos2
	Percentage of the most frequent datos3
	Percentage of the most frequent datos4
Head ICMP of the package:	
Message type (Type)	
Code of the message (Code)	
Protocol of the original package	

has left introducing alternating a dangerous inoffen-sive and other package, and when these last they have finished, then they went by the network again until all the inoffensive packages finish happening.

3 Multi-layer Perceptron

For our case of ML perceptron, guided to distinguish among inoffensive packages and packages that represent attacks to a host in a computer net, the configuration that we have adopted is the following one: The activation function for each neuron should be continuous and derivable, so the network can store information on the exit of each neuron in a more precise way, only not differing between 0 and 1, that is to say that the exit can take values in the continuous one between 0 and 1. For this end, a sigmoidea function has been used. The number of neurons in the first layer is similar to the number of data that is extracted of each package, in this case 29, and in the exit layer, since we will only distinguish between two possible values, we only use a neuron whose exit will indicate that the analyzed package is inoffensive when it is smaller or the same as 0.5, and will indicate that the analyzed package is dangerous when the exit is bigger than 0.5, since the exit is enclosed between 0 and 1 for the activation function, and the training exits have been introduced as 1 for dangerous packages, and 0 for inoffensive packages. Only a hidden layer has been used. After carrying out diverse tests varying the number of neurons of this layer, we obtained the best results with a

number of 30. The learning rate that has been used is variable. It begins with a value of 0.5, and when the error begins to oscillate (the oscillations are detected when the error ascends 4 times after 8 histories) it diminishes in 0.2 units. If the error continues oscillating, it must descend the value again of a, until it arrives at 0.1. If it continues oscillating it is lowered up to 0.06, and it no longer is lowered more. After proving several values for the moment, the one that had given The half error of each history in the learning, something that will be good to check if the error is lowered in each history, is calculated adding the error of all the samples of the history and dividing it among the number of samples, where the error of each sample is indicated in the equation (1)

$$E(hist) = \frac{1}{N} \cdot \sum_{p-1}^{N} E(p). \tag{1}$$

4 Self Organizative Map

The second proposal of neural networks work that has been proven to distinguish inoffensive packages from dangerous packages is based on unsupervised learning, concretely it is a self organizative map [6]. For our case of recognition of inoffensive packages and dangerous packages of a computer net, where the extracted information of each package is composed of 29 data, a rectangular SOM has been used (toroidal) size 40 x 40, since it has been proven with sizes of 5 x 5, 10 x 10 and 20 x 20, and the error of each history oscillated too much, indicating that clusters was superimposed, and therefore it lacked space so that these groupings are formed without blocks among them. It has also been proven to use sizes of 30 x 30 and 50 x 50, but 50 x stops 50 the results they were the same ones that using a size of 40 x 40, and 30 x stops 30, the results were a little worse. It is also necessary to consider that as much as adult is the size of the map, more will take a long time as much in the training as in the recognition, and this last is something to keep in mind if one wants to implant the system in a computer net, and that it works in real time, the minimum map size has been chosen for the reason that offered better results. The distance function used to measure the difference between a training sample and each neuron of the SOM has been the distance euclidea. As for the coefficients of upgrade of weights (Cp) for the winning neuron, and their neighbours, the following ones have been used, after having proven other values and to see that they didn't work better: Cp for the winning neuron: 0.9 Cp for the neighbouring neurons of level 1: 0.1, Cp for the neighbouring neurons of level 2: 0.005, Cp for the neighboring neurons of level 3: 0.0005]. The error of each sample is measured by the distance to the winning neuron, equation(1).

5 Obtained Results

It is necessary to say that the obtained results have overcome the initial expectations, since the fact of being able to distinguish between inoffensive packages

and dangerous packages without totally looking at the content of data of the package was something that was thought it could only be gotten in a moderate percentage of successes, but the percentage of successes gotten in the test overcame 90%. The first thing that is necessary to decide in this stage is how to measure the results. Since This is a problem bigger than which was thought of in principle, we don't prepare more than 433 dangerous packages with those that to train and to test, and neither you can prove the result of the training of the neural network work installing it in a computer net, since the surest thing is it meets with packages completely different from the packages with those that it has trained in the small network of two machines used in the project. Concretely 80% of the packages has been chosen for training, and 20% for testing, as much for the inoffensive packages as for the obtained dangerous packages. The results will measure them in terms of percentage of successes obtained in the testing packages that will be: inoffensive, dangerous packages, and the entirety of them.

5.1 Results of the Multi-layer Perceptron

Next a summary of the error is shown after each history in the training of the multilayer perceptron that has given better results, since this depends in great measure of the random initialization of the weights. We will indicate the errors of the first 10 histories, and then we will go advancing more quickly, since it allowed to train until the error didn't lower more, and there were 2153 histories (min error= 0,002132). The results as for success Package types approximate number of packages of training Inoffensive packages. Dangerous packages the rate of success of all the test packages, approximately 1233, is of 0.984750. To check that the obtained results have not been fruit of chance when selecting the training packages and the test packages, they have been carried out 3 more trainings, each one with a division of training packages and of different test. The results, after 100 training test are the Table 2.

Table 2. 100 training test results

	Division 1	**Division 2**	**Division 3**
Success inoffensive test packages	0.863755	0.849877	0.947186
Success dangerous test packages	1.000000	0.988372	0.876712
Success in all the test packages	0.872340	0.859004	0.942997

5.2 Results of the Self Organizative Map

Next the same thing will be shown for the training of the self organizative map that has given better results. The configuration of the number of neurons of the map and other parameters are previously the suitable ones. It is necessary to keep in mind that the mensuration of the error is not calculated in the same way that in the ML perceptron, for what the comparison in that sense is not possible. Yes it will be possible to compare the rate of successes. We will indicate the errors of

the first 10 histories, and then we will go advancing more quickly, since it allowed to train until the error didn't lower more, and they were 85 histories (min error= 0.004966).. The rate of success of all the test packages, approximately 1233, it is of: 0.991235. To check that the obtained results have not been fruit of chance when selecting the training packages and the test packages, 3 more trainings have been carried out, each one with a division of training packages and of different test, The results, after 10 histories of training, are the Table 3.

Table 3. 100 training test results

	Division 1	**Division 2**	**Division 3**
Success inoffensive test packages	0.995442	0.996567	0.997567
Success dangerous test packages	0.924051	0.911111	0.891304
Success in all the test packages	0.990646	0.990438	0.990189

6 Conclusions

Managing the detection of attacks is something complicated, since continually new exploits arise and the at-tackers invent new ways to penetrate the systems. The use of the neural network can suppose a great advantage in the detection of these attacks, since they have the capacity to detect attacks that they have not memorized directly. As a final conclusion one can say that the intent of checking if a neural network work could learn how to distinguish inoffensive packages of dangerous packages has been an entire success, still without using the whole information of the content of each package.

References

1. Abrams, M.D., Jadodia, S., Podell, H.J. (eds.): Information Security: An Integrated Collection of Essays. IEEE Computer Society Press, Los Alamitos California (1995)
2. Pfleeger, C.P., Pfleeger S.L.: Security in Computing. 3rd edn. Prentice Hall Inc., New Jersey (2003)
3. CERT: Incident Note IN-99-07: Distributed Denial of Service Tools. CERT Software Engineering Institute, Carnegie Mellon University, Pittsburgh (2001)
4. Kumar, S.: Classification and Detection of Computer Intrusion. PhD thesis, Purdue University, August (1995)
5. Denning, D.E.: An Intrusion-Detection Model. IEEE Trans. Software Engineering SE13 (2) (1987) 222-232.
6. Kohonen, T.: Self-Organization and Associative Memory Springer-Verlag, New York (1987)

Enhancing the Transmission Security of Content-Based Hidden Biometric Data

Muhammad Khurram Khan and Jiashu Zhang

Research Group for Biometrics and Security,
Sichuan Province Key Laboratory of Signal and Information Processing,
Southwest Jiaotong University, Chengdu, Sichuan, P.R. China
khurram.khan@scientist.com

Abstract. This paper presents a secure scheme to enhance the transmission security of content-based hidden biometric data over insecure network. Encryption, coding, spread spectrum modulation, and data hiding techniques are used to improve the security and secrecy of the transmitted templates. Secret keys are generated by the biometric image and used as the parameter value and initial condition of the chaotic maps, and each transaction session has different secret keys to protect from the attacks. Templates are encrypted by chaotic encryption, encoded by BCH encoding method, modulated by chaotic neural network parameter modulation (CNNPM), and then hid into the cover image. Experimental results show that the security, performance, and accuracy of the presented scheme is encouraging comparable with other methods found in the current literature.

1 Introduction

Biometric is defined as measurable physiological and/or behavioral characteristics that can be utilized to verify the identity of an individual. It includes fingerprint, hand geometry, palm print, voice, face, and iris recognition etc [1] [2] [3]. Biometric data is unique, but it does not provide secrecy. Only biometrics is not a panacea for the secrecy of data because it has some risks of being hacked, modified, and reused whenever it is sent over the network, so there is a need to protect biometric data from different attacks. To promote wide spread utilization of biometric techniques, an increased security and secrecy of its data are necessary. Steganography, cryptography, and watermarking can be used to achieve the biometric data security and secrecy [4] [5] [6]. Some schemes related with watermarking and steganography are proposed for the biometric data security. Due to the brevity, we refer readers to [4][5][7][8] for related work. To improve the security and secrecy, this paper presents a novel content-based hidden transmission scheme of biometric templates. Spread spectrum technique is utilized, in which Biometric templates are encrypted by chaotic encryption method and encoded by error correction code. Chaotic neural network parameter modulation (CNNPM) is employed to modulate the encrypted and encoded biometric templates and then, templates are hid into the cover image. Because chaotic trajectory is sensitive to its initial condition and parameter value, and biometric is

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 214 – 223, 2006.

random at each enrollment of the subject [9], so the secret keys are generated from the biometric image and used as the parameter value and initial condition of the chaotic map. Thus, we use biokey i.e. biometrically generated keys as the secret keys of the chaotic map. Our method utilizes discrete wavelet transform (DWT) to hide biometric data, Moreover, detailed experimental and simulation results show that the proposed scheme is effective for hidden biometric templates transmission in terms of noise performance, robustness against attacks, and payload. In addition, its implementation is simple and security is higher than conventional schemes.

2 Proposed Scheme

The proposed model is depicted in fig. 1. In the following subsections, we review all of the proposed modules in detail.

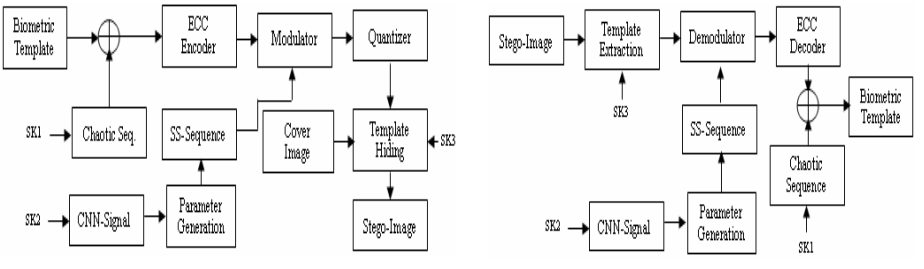
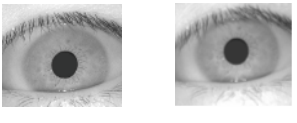


Fig. 1. Proposed Secure Transmission Scheme (a) Client side process; (b) Server side process

2.1 Secret Key Generation

Before secret keys generation, we capture the iris-biometric of user by the CCD camera. Two captured iris images of a person are shown in Fig. 2. We extract the iris features by the process and method described by John Daugman [2], the inventor of the iris recognition system. The statistical properties of iris-biometric are very random and have about 244 degree-of-freedom, means features, to identify a person [2]. The two irises of the same person have different features, which make iris-biometric a good candidate to use for the identification [27]. The pixel value distribution of the captured iris images are different at every instant, because of different posing, illumination effects, distance of a person from camera, and so on, as it can be seen in



(a) (b)

Fig. 2. Iris images of a person captured by CCD camera

Fig.2. Both, figures 2(a) and 2(b) are captured during a short break of few seconds, but the pattern got from the camera is different and gray value distribution is changed due to the eye-movement and other factors [10].

These characteristics make iris-biometric images a good candidate to generate the secret keys for the chaotic map. As the chaotic trajectory is sensitive to its initial condition and parameter value, so these values can be generated from the random pixel value distribution of the captured iris images from the camera. Assume that the iris gray image is X with size $M \times N$. It can be represented as:

$$X = \{x(i, j), 0 \le i < M, 0 \le j < N\}, \quad (1)$$

$$x(i, j) \in \{0,1,...,2^L - 1\}$$

where L is the number of bits to represent a pixel.

In the following steps, we generate secret keys to be used in the encryption process of the iris templates.

1. Randomly select two pixel values from the x-axis and y-axis of the iris image which are p_x and p_y, respectively.
2. Subtract these values and divide by the total gray scale of the image, as shown in the following equation. Note that all captured iris images are gray images.

$$V = \frac{abs(p_x - p_y)}{N} \quad 0 < V \le 1 \quad (2)$$

   where *abs()* returns absolute value of the two subtracted pixel values and N is the number of pixel bits of the iris image.
3. Now, generate the parameter value by the following equation [17]:

$$S_k = \begin{cases} 3 + V & V > 0.57 \\ 3.57 + (0.43 - V) & V \le 0.57 \end{cases} \quad (3)$$

4. We use following equation to generate the initial condition of the chaotic map:

$$p = abs(\lceil V \rceil - V) \quad 0 < p \le 1 \quad (4)$$

   where $\lceil V \rceil$ is the ceiling-value of V and *abs()* returns the absolute value.

2.2 Biometric Template Encryption and Encoding

In this paper, we utilize chaos to encrypt the biometric templates. In our method, for the encryption, we use a chaotic sequence to encrypt the biometric template. We use Logistic chaotic map to generate a sequence of real numbers [7][11][12]:

$$x_{n+1} = g(x_n) = \lambda x_n (1 - x_n) \quad (5)$$

Where $3.57 < \lambda \le 4$, the sequence is non-periodic, non-convergent, and very sensitive to the initial value, which is used as a secret key. Because the stream generated by logistic map is a sequence of real numbers, so the output sequence of equation (5) is quantized into binary stream by the following threshold to perform XOR operation with the biometric template.

$$c(n) = \begin{cases} 1, & x_n \geq 0.5 \\ 0, & x_n < 0.5 \end{cases} \tag{6}$$

The normalized sequence is $c(n) \in \{0,1\}$ and the biometric template is $s(n) \in \{0,1\}$. We use XOR operation to encrypt the template, and the encrypted template is obtained by:

$$q(n) = \sum_{n=1}^{N} s(n) \oplus c(n) \tag{7}$$

where N is size of the biometric template and generated chaotic sequence.

Now after encryption, the encrypted biometric templates are encoded by the error-correction coding (ECC) method which is used to improve the performance of the system in terms of bit error probability. For this purpose, we use Bose–Chaudhuri–Hochquenghem (BCH) codes, which are cyclic linear block codes and allow a large selection of data block length, code rates, alphabet size, and error correction capability [13]. The encoded template is represented by *e(n)*.

2.3 Spread Spectrum Modulation

In this paper, we use a chaotic neural network spread modulation method to generate such noise-like signals. We introduce an effective scheme called chaotic neural network parameter modulation (CNNPM) method [14],[15] to modulate the encrypted and encoded biometric template. The main advantages of chaotic parameter modulation (CPM) technique are its robustness to synchronization errors and potential higher capacity [16].

In the CNNPM process, we use Tent and Chebyshev maps. Tent map, which is a piecewise linear chaotic map (PWLCM), is the simplest kind of chaotic maps from the view point of implementation and realization [17]. Tent map also have uniform invariant density and good correlation property, which is very important in the cryptography and pseudo-random sequence generation [18].

$$b(i) = \begin{cases} x/p & , x \in [0, p) \\ (1-x)/(1-p) & , x \in (p,1] \end{cases} \tag{8}$$

where: p satisfies $0 < p < 1$. The value of p is used as a secret key i.e. SK_2, which is generated in equation (4). The generated sequence by equation (8) can be represented as: $\{b(i)_{i=0}^{N-1}\}$.

To generate the chaotic neural network parameters [15], we perform the following operations:

1. First, we calculate $p(n)$ by:

$$p(n) = \sum_{i=0}^{N-1} 1 - 2b(8N + i) \tag{9}$$

2. The biases θ_i are computed as follows:

$$\theta_i = \sum_{i=0}^{N-1} \frac{2b(8N+i)}{2} \tag{10}$$

3. Now, the parameters $\mu(n)$ for modulation are generated as follows:

$$\mu(n) = \sum_{i=0}^{N-1} \left[(p(n) \times b(i)) + \theta_i \right] \tag{11}$$

After generating bifurcation parameters, we limit them in a manner to ensure that all the values of $\mu(n)$ lie in the chaotic range of Chebyshev map i.e. $\mu(n) \in (1.3,2]$. Chebyshev map is an interesting chaotic sequence generator, whose chaoticity can be easily verified by rigorous mathematical analysis [19]. The sequences generated by Chebyshev map have good statistical properties and are pseudorandom in nature. Chebyshev map is defined by equation (12):

$$z_{n+1} = \cos(\mu \arccos(z)) \quad , z \in [-1,1], \mu \in (1.3,2] \tag{12}$$

The value of μ is generated in equation (11), which is used as the bifurcating parameter of Chebyshev map. Each bifurcating parameter $\mu \in [\mu_{\min}, \mu_{\max}]$ has a unique invariant Ergodic measure for the chaotic map [14]. The sequence generated by equation (12) used for modulation and the modulated template can be represented by $w(n)$.

2.4 Biometric Template Hiding

In DWT-based data hiding, an image is decomposed into four bands; low frequency (LL), high frequency (HH), low-high frequency (LH), and high-low frequency (HL) sub-bands. The lower sub-band is not suitable for data hiding, because it contains the important information of an image and image quality is degraded when the data is embedded. For this reason the modulated biometric template is used as hiding position of LH2, HL2, and HH2 sub-bands. To avoid overlapping of data allocation and modification of pixel value, we select the data hiding position separated with at least one pixel. The same pixel should not be selected as embedding/hiding position in selecting the procedure. If size of the biometric template is large, we can also select the pixels in HL3, LH3 and HH3 for embedding e.g. the template size of multimodal biometric template is large, which can contain face, voice or iris together.

After that, we compute neighboring symbol's mean value of selected pixel by using (13). The pixel value of the data embedding position is $x_{i,j}$, and the mean value of the neighboring value is $m_{i,j}$, then we compute $m_{i,j}$ using the following formula:

$$m_{i,j} = mean(x_{i,j-1}, x_{i-1,j}, x_{i+1,j}, x_{i,j+1}) \tag{13}$$

After computing the mean value, we embed the encrypted iris template into the selected pixels and generate flags ($f_{i,j}$) by the following steps:

If the selected $x_{i,j}$ is bigger than the mean value $m_{i,j}$ and $w_{i,j} = 1$, we change $x_{i,j}$ with $x'_{i,j}$ using (14) and create a flag $f_{i,j}$ that contains value of 0.

$$\begin{aligned} &if\,(x_{i,j} > m_{i,j} and w_{i,j} = 1) then \\ &x'_{i,j} = x_{i,j}(1 - \alpha w_{i,j}) \\ &f_{i,j} = 0 \end{aligned} \tag{14}$$

where α is the data hiding/embedding strength.

If the selected $x_{i,j}$ is bigger than mean value $m_{i,j}$ and $w_{i,j} = 0$, we change $x_{i,j}$ with $x'_{i,j}$ using (15) and create a flag $f_{i,j}$ that contains value of 1.

$$\begin{aligned} &if\,(x_{i,j} > m_{i,j} and w_{i,j} = 0) then \\ &x'_{i,j} = x_{i,j}(1 - \alpha) \\ &f_{i,j} = 1 \end{aligned} \tag{15}$$

If the selected $x_{i,j}$ is smaller than the mean value $m_{i,j}$ and $w_{i,j} = 1$, we change $x_{i,j}$ with $x'_{i,j}$ using (16) and create a flag $f_{i,j}$ that contains value of 2.

$$\begin{aligned} &if\,(x_{i,j} < m_{i,j} and w_{i,j} = 1) then \\ &x'_{i,j} = x_{i,j}(1 + \alpha w_{i,j}) \\ &f_{i,j} = 2 \end{aligned} \tag{16}$$

If the selected $x_{i,j}$ is smaller than the mean value $m_{i,j}$ and $w_{i,j} = 0$, we change $x_{i,j}$ with $x'_{i,j}$ using (17) and create a flag $f_{i,j}$ that contains value of 3.

$$\begin{aligned} &if\,(x_{i,j} < m_{i,j} and w_{i,j} = 0) then \\ &x'_{i,j} = x_{i,j}(1 + \alpha) \\ &f_{i,j} = 3 \end{aligned} \tag{17}$$

These equations describe that the modified value $x'_{i,j}$ is proportional to the original value of the image $x_{i,j}$, which makes the hidden iris-biometric template more robust. The generated flag $f_{i,j}$ is stored in the database and is used in the template extraction procedure. After this process, we take the inverse transform of the host image using inverse discrete wavelet transform (IDWT) to reconstruct the stego-image. Now, the stego-image can be transmitted to the authentication server to perform identification of the claimed identity.

2.5 Template Extraction

On the authentication server side, for the hidden data extraction, we transform the stego-image using second level discrete wavelet decomposition, as it was decomposed in the template hiding/embedding process to extract the data from LH2, HL2 and HH2 sub bands. From the transform image, we find the embedded location of hidden template by using the same key, which was used in the hiding procedure. After then, we compute neighbor's mean value $m_{i,j}$ using the same method described in data hiding process and generate a flag $f'_{i,j}$, and then generate $w'_{i,j}$ using the opposite procedure described in flag generation and template hiding process in subsection 2.4.

After successfully extraction of the biometric template from the received stego-image, we perform demodulation. The biometric templates are demodulated by the same method as described in the subsection 2.3. The demodulated template is represented by $d'(n)$. After demodulation, we decode biometric templates by the same BCH codes as used in encoding process and decoded template is represented by $q'(n)$. Decoded biometric templates are decrypted by using the symmetric secret key, which was used in the encryption process. The same chaotic sequence $c(n)$ is generated by the chaotic decryption system and applied on the decoded template i.e. $q'(n)$. We utilize the following formula for the template decryption:

$$s'(n) = \sum_{n=1}^{N} q'(n) \oplus c(n) \tag{18}$$

At the end, to perform matching of decrypted templates, we compare $s'(n)$ with the original template that is stored in the database. Equation (19) is used for this matching process.

$$M = \frac{1}{N} \sum_{i=1}^{N} s'(n) \oplus s(n) \tag{19}$$

where the Boolean operator (XOR) equals 1, if and only if the two bits are different. Here, N is the size of the template, and $s(n)$ is original template stored in the database. We may define a matching threshold according to the criticality and usage of the system.

3 Experimental Results and Discussions

In this section, we discuss experimental results performed during the research of this work. To evaluate the performance of the proposed method, portions of the research in this paper use the CASIA iris image database collected by institute of automation, Chinese academy of sciences. CASIA iris image database (ver 1.0) contains 756 iris images from 108 eyes [20].

A well designed data-hiding system should fulfill the 'Kerchoff's' principle of cryptography [21]. In our system, we generated secret keys from the iris-biometric

image i.e. *biokey*, because biometric is a potential source of high-entropy and has been suggested as a way to enable strong, cryptographically-secure authentication of a person without requiring them to remember or store traditional cryptographic keys [9]. To perform the experiments, we used *biokey* as the parameter value and initial condition of the chaotic map. The results that the generated keys are very sensitive and a tiny amount of change can abruptly change the behavior of the system. As an example, at the encryption end, we used the secret key S_k =0.345, but when we decrypted the data by a slightly different value of the key i.e. S_k =0.3450000000001, then the recovered data is different. Hence, our encryption module is very sensitive to the secret keys mismatch (10^{-14}). Furthermore, the dynamic secret key i.e. *biokey* generates a sequence of different pseudorandom numbers at each transaction session.

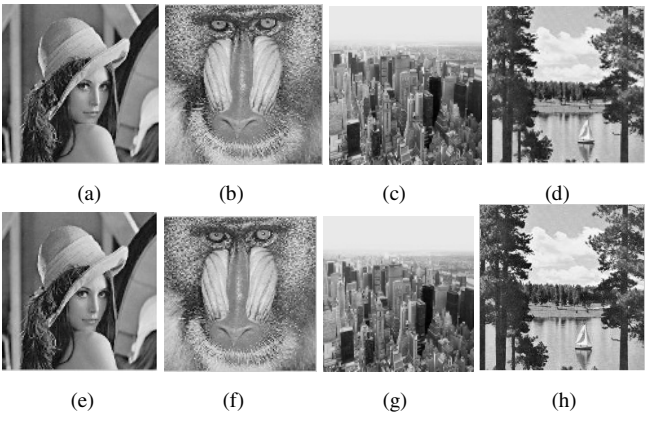


Fig. 3. Sample cover images (a-d) and watermarked stego images (e-h) (New York image courtesy of Patrick Loo, University of Cambridge, others from USC-SIPI)

The embedded iris-biometric template size is 512 bytes, which depends on the feature extraction and image processing algorithms and varies from system to system [2]. We performed experiments on different cover-images as shown in Fig. 3. We used Baboon, Lena, New York, and Sailboat images for the experiments and the extracted iris-biometric data is exactly the same as it was encoded. The hidden data extraction performance in comparison with others is shown in Table 1.

Table 1. Comparison of the proposed method (In %)

Image	Method in [22]	Method in [5]	Our method
Lena	99.57	99.78	99.86
Baboon	91.79	100	99.84
Sailboat	73.87	99.78	99.81
New York	98.49	99.78	99.79

Furthermore, Gunsel et al. [5] and Jain-Uludag [4] didn't apply encryption on the biometric template before or after the embedding/hiding, so their method is susceptible if a hacker becomes successful in extracting the hidden biometric data from the transmitted stego-image. Thus, their scheme is vulnerable to the so-called copy attack [23][24], where any unauthorized person can copy hacked biometric data from the host image and misuse it on purpose.

4 Conclusion

In this work, we have presented a novel scheme to improve the transmission security of content-based hidden biometric templates, in which encryption, encoding, spread spectrum modulation, and data hiding methods are combined to ensure the authenticity, confidentiality, and integrity of the transmitted templates. We generated secret keys from the biometric data and used as the parameter value and initial condition of the chaotic map. We encrypted biometric templates by chaotic encryption, encoded by BCH encoding method, modulated by chaotic neural network parameter modulation (CNNPM), and then hid into the cover image, which makes our system more secure and protected from the copy attack. Due to the excellent time-frequency features and well matching to the human visual system (HVS) characteristics, we used DWT to hide iris templates. Furthermore, we performed a series of experiments to evaluate the proposed system. Moreover, we carried out extensive quantitative comparisons among some existing methods and provided discussions on the overall experimental results.

Acknowledgements

This project is supported by 'Sichuan Youth Science and Technology Funds' under grant number: 03ZQ026-033 and 'Southwest Jiaotong University Doctors Innovation Funds 2005'.

References

1. Anil, K.J. , Pankanti, S., Bolle, R.: Biometrics: Personal Identification in Networked Society. Kluwer, USA (1999)
2. Daugman, J.: High Confidence Visual Recognition of Persons by a Test of Statistical Independence. IEEE Transactions on Pattern Analysis and Machine Intelligence 15 (1999) 1148-61
3. Muhammad, K.K., Zhang, J.: Securing Biometric Templates for Reliable Identity Authentication. ICIC, Lecture Notes in Computer Science, Vol. 3645, part II. Springer-Verlag, Berlin Heidelberg New York (2005) 723-32
4. Anil, K.J. , Umut, U.: Hiding Biometric Data. IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (2003) 1494-8
5. Gunsel, B., Umut U., Tekalp, AM. : Robust Watermarking of Fingerprint Images. Pattern Recognition Elsevier Science Ltd. 35 (2002) 2739-47
6. Cox, I.J. : Digital Watermarking. Morgan Kaufmann Publishers, CA, USA (2002)

7. Yeung, M.M., Pankanti, S.: Verification Watermarks on Fingerprint Recognition and Retrieval. Journal of Electronic Imaging 9 (2000) 468-76
8. Sonia, J.: Digital Watermarking Techniques: A Case Study in Fingerprints and Faces. Proc. Indian Conference on Computer Vision, Graphics, and Image Processing (2000) 139-44
9. Teoh, A.B.J., David, C.L.N., Goh, A.: Personalised Cryptographic Key Generation based on Facehashing. Computers and Security 23 (2004) 606-14
10. Daugman, J., Downing, C.: Epigenetic Randomness, Complexity, and Singularity of Human Iris Patterns. Proc. of the Royal Society, Biological Sciences, B-268 (2001) 1737-40
11. Zhao, D., Chen, G.R., Liu, W.: A Chaos-based Robust Wavelet-domain Watermarking Algorithm. Chaos, Solitons and Fractals 22 (2004) 47-54
12. Zhang, J., Lei, T., Tai, H.M. : A New Watermarking Method Based on Chaotic Maps. Proc. IEEE ICME'04, (2004)
13. Sklar, B.: Digital Communications. Prentice Hall PTR, Upper Saddle River, New Jersey, USA (2001)
14. Siyue, C., Henry L,: Ergodic Chaotic Parameter Modulation with Application to Digital Image Watermarking", IEEE Trans. on Image Processing 14(10) (2005) 1590-1602
15. Lian, S., Chen, G.R., Cheung, A., Wang, Z.Q.: A Chaotic Neural Network based Encryption Algorithm for JPEG2000 Encoded Images. In: Advances in Neural Networks-ISNN 2004, LNCS, Springer-Verlag (2004) 627-632
16. Yu, H., Leung, H.: A Comparative Study of Different Chaos-based Spread Spectrum Communication Systems, Proc. IEEE ISCAS, (III) (2003) 213-216
17. Li, S., Li, Qi., Li, W., Mou, X.Q., Cai, Y.L. : Statistical Properties of Digital Piecewise Linear Chaotic Maps and Their Roles in Cryptography and Pseudo-Random Coding, IMA-C&C, Vol. 2260, LNCS, Springer-Verlag (2001) 205-221
18. Baranovsky, A., Daems, D.: Design of One-dimensional Chaotic Maps with Prescribed Statistical Properties. Int. J. Bifurcation and Chaos 5(6) (1995) 1585-1598
19. Liao, X.F., Li, X., Pen, J., Chen, G.R.: A Digital Secure Image Communication Scheme based on the Chaotic Chebyshev Map, Int. Journal of Communication Systems 17 (2004) 437-445
20. CASIA Iris Database, [online] http://www.sinobiometrics.com (2005).
21. Katzenbeiser, S., Petitcolas, F.A.: Information Hiding Techniques for Steganography and Digital Watermarking. Artech House Books (2000)
22. Umut, U., Gunsel, B., Tekalp, A.M.: Robust Watermarking of Busy Images. Proc. of SPIE Electronic Imaging 2001 Conference, Security and Watermarking of Multimedia Contents III, USA (4314) (2001) 18-25
23. Deguillaume, F., Voloshynovskiy, S., Pun, T.: Secure Hybrid Robust Watermarking Resistant against Tampering and Copy Attack. Signal Processing, Elsevier Science Ltd. (2003) 2133-70
24. Muhammad, K.K., Zhang, J.: Enhancing the Security of Biometric Templates for Trust Worthy Person Authentication. GESTS International Transaction on Computer Science & Engineering 5 (2005) 166-76

Building Lightweight Intrusion Detection System Based on Random Forest

Dong Seong Kim, Sang Min Lee, and Jong Sou Park

Network Security Lab., Computer Engineering Department,
Hankuk Aviation University, 200-1, Hwajeon-dong,
Deogyang-gu, Goyang-city, Gyeonggi-do, 412-791, Korea
{dskim, minuri33, jspark}@hau.ac.kr

Abstract. This paper proposes a new approach to build lightweight Intrusion Detection System (IDS) based on Random Forest (RF). RF is a special kind of ensemble learning techniques and it turns out to perform very well compared to other classification algorithms such as Support Vector Machines (SVM) and Artificial Neural Networks (ANN). In addition, RF produces a measure of importance of feature variables. Our approach is able not only to show high detection rates but also to figure out stable output of important features simultaneously. The results of experiments on KDD 1999 intrusion detection dataset indicate the feasibility of our approach.

1 Introduction

Intrusion Detection System (IDS) plays vital role of detecting various kinds of attacks. The main purpose of IDS is to find out intrusions among normal audit data and this can be considered as classification problem. As new attacks appear and amount of audit data increases, IDS should counteract them. IDS may utilize additional hardware such as network processor, System on Chip (SoC) and Field-Programmable Gate Array (FPGA) [17]. Additional hardware can increase packet capture speed and decrease processing time but it needs more costs and may not enhance detection rates of IDS. In addition to this, as network speed becomes faster, there is an emerging need for security analysis techniques that will be able to keep up with the increased network throughput [10]. Therefore, IDS itself should be lightweight while guaranteeing high detection rates. This paper proposes a new approach to build lightweight IDS. Previous researches have focused on parameter optimization of intrusion detection model and feature selection of audit data. Firstly, the objective of parameter optimization is to regulate parameters of detection model based on various kinds of classification algorithms such as Support Vector Machines (SVM) [5, 6, 12], Hidden Markov Model (HMM) [13], several kinds of Artificial Neural Network (ANN) [4, 18] and so on. For example, Kim and Park [9, 14] and Mukkamala *et al.* [20] have tried to optimize parameters of kernel function in SVM. Secondly, the objective of feature selection of audit data is to figure out relevant features among whole features to decrease processing time of audit data and to improve detection rates. For feature selection, wrapper and filter method have been

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 224 – 230, 2006.

proposed; wrapper method adopts classification algorithms and performs cross-validation to identify important features. Sung and Mukkamala [18] have tried to identify and categorize features according to their importance to detect a specific kind of attacks such as probe, DoS (Denial of Service), Remote to Local (R2L) and User to Root (U2R). They have used backward feature selection method and SVM and ANN as feature selection algorithms. On the other hand, filter method utilizes correlation based approach, which utilizes feature-class and inter-feature correlation. Filter method is more lightweight than wrapper method in terms of computation time and overheads since it is performed independent of classification algorithms. However, filter method has lower detection rates than wrapper method due to its low correlation with classification algorithms. In order to improve these problems, several studies have proposed hybrid approaches which combine wrapper and filter approach. Kim *et al.* [9] have proposed fusion approach to optimize both feature selection and parameter regulation. Park *et al.* [14] have proposed correlation based hybrid approach, which combines filter method with wrapper method through Genetic Algorithm (GA) operation. However, hybrid approaches may inherit both filter and wrapper approach's drawbacks; they sometimes show a little degradation on detection rates with more computations rather than the naive filter method. To cope with them, this paper proposes a new approach to build lightweight IDS based on Random Forest (RF). RF is a special kind of ensemble learning techniques and turns out to perform very well compared to other classification algorithms, including Support Vector Machines (SVM), Artificial Neural Network (ANN) and so on. In addition, RF produces a measure of importance of the feature variables. Proposed approach based on RF is able not only to show high detection rates but also to figure out important feature simultaneously, without any further overheads compared to hybrid approaches [9, 14]. The results of experiments on KDD 1999 intrusion detection dataset indicate the feasibility of our approach.

2 Random Forest

This paper proposes lightweight IDS based on Random Forest (RF). RF is random forests are comparable and sometimes better than state-of-the-art methods in classification and regression [11]. RF is a special kind of ensemble learning techniques [2]. RF has low classification (and regression) error comparable to SVM. Moreover, RF produces additional facilities, especially feature importance. RF is robust concerning the noise and the number of attributes. The learning instances are not selected with bootstrap replication is used for evaluation of the tree, called OOB (out-of-bag) evaluation, which is unbiased estimator of generalization error. RF builds an ensemble of CART tree classifications using bagging mechanism [3]. By using bagging, each node of trees only selects a small subset of features for the split, which enables the algorithm to create classifiers for high dimensional data very quickly. One has to specify the number of randomly selected features (*mtry*) at each split. The default value is *sqrt(p)* for classification where *p* represents number of features. The Gini index [1] is used as the splitting criterion. The largest possible tree is grown and is not pruned. One also should choose the big enough number of trees (*ntree*) to

ensure that every input feature gets predicted several times. The root node of each tree in the forest keeps a bootstrap sample from the original data as the training set. The OOB estimates are based on roughly one third of the original data set. By contrasting these OOB predictions with the training set outcomes, one can arrive at an estimation of the predicting error rate, which is referred to as the OOB error rate. In summary, classification accuracy of RF is high comparable to that of SVM, and it also generates individual features' importance, these two properties help one to build lightweight IDS with small overheads compared to previous approaches. Our proposed approach will be presented in next section.

3 Proposed Approach

The overall flow of proposed approach is depicted in figure 1. The network audit data is consisting of training set and testing set. Training set is separated into learning set, validation set. Testing set has additional attacks which are not included in training set. In general, even if RF is robust against over-fitting problem [2], we use n-fold cross validation method to minimize generalization errors [3]. Learning set is used to train classifiers based on RF and figure out importance of each feature of network audit data. These classifiers can be considered as detection models in IDS. Validation set is used to compute classification rates by means of estimating OOB errors in RF, which are detection rates in IDS. Feature importance ranking is performed according to the result of feature importance values in previous step. The irrelevant feature(s) are eliminated and only important features are survived. In next phase, only the important features are used to build detection models and evaluated by testing set in terms of detection rates. If the detection rates satisfy our design requirement, the overall procedure is over; otherwise, it iterates the procedures. We carry out several experiments on KDD 1999 intrusion detection dataset and experimental results and analysis are presented in next section.

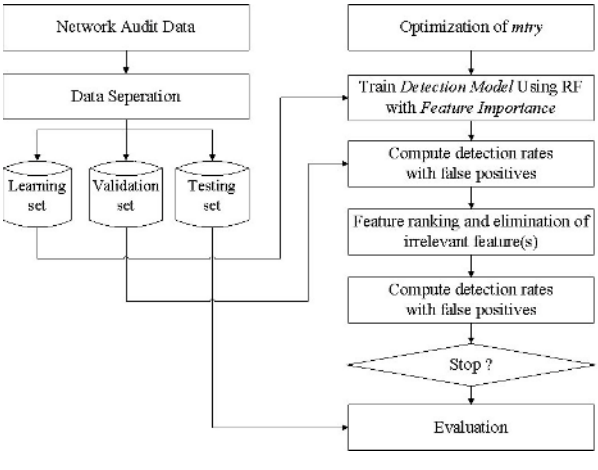


Fig. 1. Overall flow of proposed approach

4 Experiments and Analysis

4.1 Experimental Environments and Dataset

All experiments were performed in a Windows machine having configurations Intel (R) Pentium(R) 4, 1.70 GHz (over 1.72 GHz), 512 MB RAM. We have used RF version (R 2.2.0) in open source R-project [19]. We have used the KDD 1999 CUP labeled dataset [6] so as to evaluate our approach. Stolfo *et al.* [8] defined higher-level features that help in distinguishing normal connections from attacks. The dataset contains 24 different types of attacks that are broadly categorized in four groups such as probes, DoS (Denial of Service), U2R (User to Root) and R2L (remote to local). Each instance of data consists of 41 features which we have labeled as *f1*, *f2*, *f3*, *f4* and so forth. We only used DoS type of attacks, since other attack types have very small number of instances so that they are not suitable for our experiments [15]. The dataset is consists of training set and testing set. We randomly split the set of labeled training set into two parts; learning set and validation set. Learning set is used to adjust the parameters in RF. Validation set is used to estimate the generalization error of detection model. The generalization errors are represented as OOB errors in RF. In order to achieve both low generation errors, in other words, high detection rates, we have adopted 10-fold cross validation with 2000 samples. Testing set is used to evaluate detection model built by learning set and validation set.

4.2 Experimental Results and Analysis

RF has two parameters; the number of variables in the random subset at each node (*mtry*) and the number of trees in the forest (*ntree*). The optimization of two parameters is necessary to guarantee high accuracy of classification, called high detection rates. We regulated optimal value of *mtry* using *tuneRF( )* function in *Random Forest Package*, and it turned out *mtry* = 6. In case of *ntree*, there is not a specific function; we regulated optimal *ntree* value by carrying out experiments with varying *ntree* values from 0 to 350. The optimal *ntree* value can be evaluated referring to the OOB errors (see figure 2). We assume that detection rates are determined by equation "1 – OOB errors". The experimental results for determination of optimal *ntree* values are depicted in figure 2. According to figure 2, detection rate

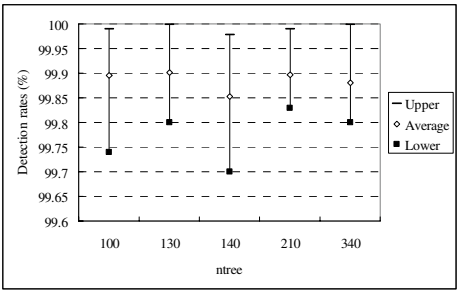


Fig. 2. Detection rates vs. ntree values from 0 to 350

of RF turned out most high and stable when *ntree* = 130 and 340. It spends more training time in case of *ntree* = 340 than in that of *ntree* = 130, and we selected *ntree* = 130. As the result of experiments, we set two optimized parameter values; *mtry* = 6, *ntree* = 130. After determination of these two parameters, we have performed "*feature selection*" according to the result of feature importance. The each feature importance values vary a little bit in each experiment. We ranked individual features in descending order according to the average value of each features as the result of 30 times iteration with 2000 samples. The important features from upper 1st and 5th and their properties are summarized in Table 1.

Table 1. The upper 5 important features and their properties

Feature	Properties	Average importance values
f23	number of connections to the same host as the current connection in the past two seconds	0.3314
f6	number of data bytes from destination to source	0.2961
f13	number of "compromised"' conditions	0.2484
f3	network service on the destination, e.g., http, telnet, etc.	0.2153
f12	1 if successfully logged in; 0 otherwise	0.2065

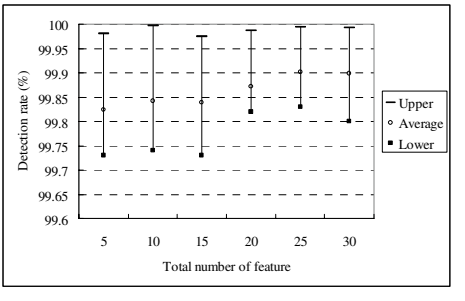


Fig. 3. Detection rates vs. number of features

The order of feature importance ranking varies according to the dataset but these 5 features almost appear in these upper 5 features, it means that these features are intrinsic features. This is comparable results to Kim *et al.* [9] and Sung and Mukkamala's approach [18]. Kim *et. al.*'s approach has showed important 'feature set' but didn't show the individual features' importance. Sung and Mukkamala's approach [18] have ranked feature importance but importance values have a little difference so that it is not applicable to build real IDS. On the other hand, our approach showed not only definitely individual feature importance but also reasonable context information for each important feature. For example, in case of feature *f23*, it represents "number of connections to the same host as the current

connection in the past two seconds" property. For DoS attacks detection, it's one of important metric even applied in preprocessor plug-in in SNORT [16].

As the results of feature selection and elimination, there occurs a little degradation in detection rates but it is marginally small (see figure 3). Experimental results have showed higher detection rates comparable to Kim *et al.*'s fusion approach [9] and Park *et. al* 's hybrid feature selection approach [14]. In summary, both optimal values for *mtry* and *ntree* parameters were determined through parameter optimization, and importance features were figured out through feature selection of audit data. This means that our approach performs both parameter optimization and feature selection while it has higher detection rates, with stable results of important features. These advantages in our approach enable one to model and implement lightweight IDS.

5 Conclusions

Existing studies to build lightweight IDS have proposed two main approaches; parameters optimization of detection models (classification algorithms) and feature selection of audit data. A numbers of researches on parameters optimization have been proposed, which are based on a variety of classification algorithms, including SVM, ANN and so on [5, 6, 9, 12, 14, 18]. The feature selection of audit data has adopted two main methods; wrapper and filter method. The hybrid approaches have been proposed to improve both filter and wrapper method [9, 14]. However, there are still rooms for improvement in terms of detection rates and stable selection for important features. This paper has presented how to build lightweight IDS based on RF, since performance of RF turns out comparable to that of SVM and it also produces feature variables importance. Several experiments on KDD 1999 intrusion detection dataset have been carried out; experimental results indicate that our approach is able not only to guarantee high detection rates but also to figure out important features of audit data.

Acknowledgements

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment).

References

1. Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J.: Classification and Regression Trees. Chapman and Hall, New York (1984)
2. Breiman, L.: Random forest. Machine Learning 45(1) (2001) 5–32
3. Duda, R. O., Hart, P. E., Stork, D. G.: Pattern Classification. 2nd edn. John Wiley & Sons, Inc. (2001)

4. Fox, K. L., Henning, R. R., Reed, J. H., Simonian, R. P.: A Neural Network Approach Towards Intrusion Detection. In Proc. of the 13th National Computer Security Conf., Washington, DC (1990)
5. Fugate, M., Gattiker, J. R.: Anomaly Detection Enhanced Classification in Computer Intrusion Detection. In.: Lee, S.-W., Verri, A. (eds.): Pattern Recognition with Support Vector Machines. Lecture Notes in Computer Science, Vol. 2388. Springer-Verlag, Berlin Heidelberg New York (2002) 186–197
6. Hu, W., Liao, Y., Vemuri, V. R.: Robust Support Vector Machines for Anomaly Detection in Computer Security. In Proc. of Int. Conf. on Machine Learning and Applications 2003, CSREA Press (2003) 168–174
7. KDD Cup 1999 Data.: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
8. KDD-CUP-99 Task Description: http://kdd.ics.uci.edu/databases/kddcup99/task.html
9. Kim, D., Nguyen, H.-N., Ohn, S.-Y., Park, J.: Fusions of GA and SVM for Anomaly Detection in Intrusion Detection System. In.: Wang J., Liao, X., Yi, Z. (eds.): Advances in Neural Networks. Lecture Notes in Computer Science, Vol. 3498. Springer-Verlag, Berlin Heidelberg New York (2005) 415–420
10. Kruegel, C., Valeur, F.: Stateful Intrusion Detection for High-Speed Networks. In Proc. of the IEEE Symposium on Research on Security and Privacy (2002) 285–293
11. Meyer, D., Leisch, F., Hornik, K.: The Support Vector Machine under Test. Neurocomputing. 55 (2003) 169–186
12. Nguyen, B. V.: An Application of Support Vector Machines to Anomaly Detection. Research in Computer Science-Support Vector Machine, report (2002)
13. Ourston, D., Matzner, S., Stump, W., Hopkins, B.: Applications of Hidden Markov Models to Detecting Multi-Stage Network Attacks. In Proc. of the 36th Hawaii Int. Conf. on System Sciences, IEEE Computer Society Press (2002) 334–343
14. Park, J., Shazzad, Sazzad, K. M., Kim, D.: Toward Modeling Lightweight Intrusion Detection System through Correlation-Based Hybrid Feature Selection. In.: Feng, D., Lin, D., Yung, M. (eds.): Information Security and Cryptology. Lecture Notes in Computer Science, Vol. 3822. Springer-Verlag, Berlin Heidelberg New York (2005) 279–289
15. Sabhnani, M., Serpen, G.: On Failure of Machine Learning Algorithms for Detecting Misuse in KDD Intrusion Detection Data Set. Intelligent Analysis (2004)
16. SNORT, http://www.snort.org
17. Song, H., Lockwood, J. W.: Efficient Packet Classification for Network Intrusion Detection using FPGA. In.: Schmit, H., Wilton, S. J. E. (eds.): In Proc. of the ACM/SIGDA 13th Int. Symposium on Field-Programmable Gate Arrays. FPGA (2005) 238–245
18. Sung, A. H., Mukkamala, S.: Identifying Important Features for Intrusion Detection Using Support Vector Machines and Neural Networks. In Proc. of the 2003 Int. Symposium on Applications and the Internet Technology, IEEE Computer Society Press (2003) 209–216
19. The R Project for Statistical Computing, http://www.r-project.org/
20. Mukkamala, S., Sung, A. H., Ribeiro, B. M.: Model Selection for Kernel Based Intrusion Detection Systems, In Proc. of Int. Conf. on Adaptive and Natural Computing Algorithms, Springer-Verlag (2005) 458–461

Intrusion Detection Based on Fuzzy Neural Networks

Ji-yao An[1,2], Guangxue Yue[1,3], Fei Yu[1,2], and Ren-fa Li[1]

[1] College of Computer & Communication, Hunan University, 410082 Changsha, China
jt_anbob@hnu.cn
[2] State Key Laboratory of Information Security,
Graduate School of Chinese Academy of Sciences, 100049 Beijing, China
hunanyufei@163.com
[3] Department of Computer Science, Huaihua University, 418000 Huaihua, China
guangxueyue@yahoo.com.cn

Abstract. A new network intrusion detection system is presented in this paper. The system is skillfully combined with fuzzy technique and neural network which architecture and arithmetic is redesigned. In order to overcome the difficulty of specifying the membership function of rules depending on experiences of experts in multi-dimensional space, fuzzy neural network model is introduced to carry through proper nonlinear division of input/output characteristics of complex system and to generate fuzzy rule sets and added membership relation automatically. The new system architecture adopts the network processor to collect and analyze the data in the low layer of network, and a prototype system is established. This prototype system behaves better ability of intrusion detection and lower rate of distort, and that it has the ability to detect unknown attack.

1 Introduction

The information foundation establishment has become an important support point of the national economy [1], as an important part of the information foundation establishment, information security relates to national live or death, economic development, social stability. Intrusion detection technique is investigative hotspot after the "fire wall", "encrypt" network security etc., is the new generation of the security guarantee technology [2].

The most mature abnormal detection method, which is also provided by initially, is based on rule and expert system. But these methods don't concern for fuzzy feature of intrusion action. For instance, it is possible that some intrusion action cannot be detected because of a little difference between rules in rule database, and some a little difference of normal action often result in undeserved alerts.

In order to increase flexibility of intrusion detection system (for short, IDS) and improve upon mutation between smooth abnormal state and normal state, a new intrusion detection system is presented in this paper, which is based on fuzzy neural network (for short, FNN-IDS); Along with fuzzy reasoning method and neural network technique, the system increases the ability of intrusion detection and decreases the rate of distort.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 231 – 239, 2006.

2 Improvement of Intrusion Detection

The factor how to judge many intrusion actions is commonly complicated, such as run state of system resource、illness data packet or some statistical variables etc. Sometime, to judge whether a network connection is an attack action or not needs various factors, many expert system have anciently been realized only through filtering and matching few intrusion features, such as SNORT or IDES etc. The result is that such system will be provided with higher rate of distort and lower performance [3].

Moreover, intrusion detection technique based on feature matching constantly utilizes a kind of duality sentence to filter and match intrusion feature, and when one factor appears to suspicious value in factor set, decision module directly brings out the conclusion of attack or normal. But in fact, single factor's effect on the result of judge has uncertainty side to great extent. Only colligating the uncertainty of various factors together, the judge is likely exact.

2.1 Import of Fuzzy Concept

Fuzzy neural network is a new network system combined with fuzzy theory and neural network technique, which model is fuzzed in some extent on the base of traditional neural network model. The main feature are shown that there is a fuzzy neural cell in every node, whose value is defined in [0,1], and then expresses its fuzzy extent.

On the basic of single factor, we can obtain related decision vector from its relevant fuzzy set and membership function. Colligating various decision vectors of such single factor together, we can obtain its related intrusion feature set from every type of intrusion action, and then weight suitably every vector in such an intrusion feature set, and thus obtain a fuzzy judge to detect whether a network connection is normal or not.

What's more, making use of the ability of learning in neural network system, we can carry through proper nonlinear division of input/output characteristics of complex system and generate fuzzy rule sets and added membership relation automatically. Apparently, it needs a distributed process method storing all of fuzzy rules, and that their process is parallel. If our sample data are abundance sufficiently, then the run state of network or audited system can be behaved expressly. At the same time, the process to consequence part of rules is a set of mapping relation. All of the work can be rightly competence for neural network.

2.2 Fuzzy Reasoning Model for Intrusion Detection

Fuzzy neural network reasoning model is shown as fig.1, which includes two kind of fuzzy neural network: (1) NNw, utilizing the antecedent of fuzzy rules, fuzzy rule reasoning process treats membership measure of every antecedent of fuzzy rule in the input mode, and weights all of fuzzy rules, and thus realizes the reasoning of fuzzy rules; (2) NN1—NNr, they are alike fuzzy neural network, and realize the consequence mapping of r fuzzy rules, which are trained offline. Acting the membership measure of antecedent of rule as weight from NN_w, acquiring the result of consequent mapping in NN_1—NN_r, and carry through added weight to compute fuzzy reasoning of attack action.

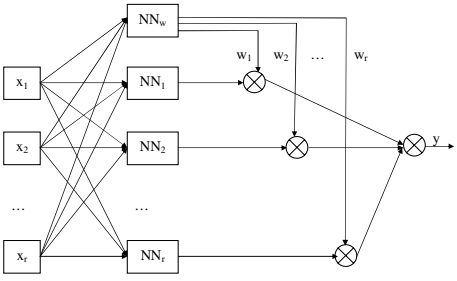


Fig. 1. Fuzzy reasoning model for intrusion detection

Two type of network referred as fig.1, such as NNw and NNs , their fuzzy process and material neural network architecture are concurrent, which is called by a joint name fuzzy neural network (FNN). In brief, the architecture of FNN is shown as fig.2. Abstractly, the weight between input layer and fuzzy layer equals 1; the output of fuzzy layer is the membership measure of input vector which value is defined in (0,1); the weight between fuzzy layer and fuzzy reasoning layer equals also 1; the output of fuzzy reasoning layer is the matching measure of antecedent of various fuzzy rules, which equals the mean of various fuzzy measure; the weight between fuzzy reasoning layer and output layer equals w_{ij}, the total output of output layer is added weight sum of matching measure of various fuzzy rules.

2.3 Detection and Train Step for Intrusion Action

We adopt forward feedback triple-tier BP neural network model in every neural network, which are designed as follows:

Fuzzy rules in above neural network are formed as $IF\ \mathrm{x} \in \mathrm{A}^{\mathrm{s}}\ \ THEN\ \mathrm{y} = \mathrm{F}_{\mathrm{s}}(x)$, where x is input vector, $\mathrm{y} = \mathrm{F}_{\mathrm{s}}(x)$ is output vector, and A^{s} is a kind of fuzzy rule set correspond to fuzzy rule s.

We suppose that training sample data set is $T^s = \left\{(x_i^s, y_i^s) \mid 1 \le i \le n_t^s\right\}$, $1 \le s \le r, n_t^s$, and its check model set is $C = \left\{(x_i^c, y_i^c) \mid 1 \le i \le n_c\right\}$. But, if the rule is uncertainty, and has only many related data, and then we firstly treat these related data to obtain above training sample data, and thus we make use of other a triple-tier BP network model to classify these uncertainty rules. The actual method is described 8 steps as follows:

(1) Remove an input variable from input vector that has little related with output vector. Firstly, we train the triple-tier BP neural network for all training sample data to obtain mean variance ε_m; secondly, we train repeatedly the neural network removed input variable x_p to obtain mean variance ε_{m-1}^p, if $\varepsilon_m \ge \varepsilon_{m-1}^p$, then it is possible that to remove input variable x_p affects hardly the fuzzy reasoning of the fuzzy rules.

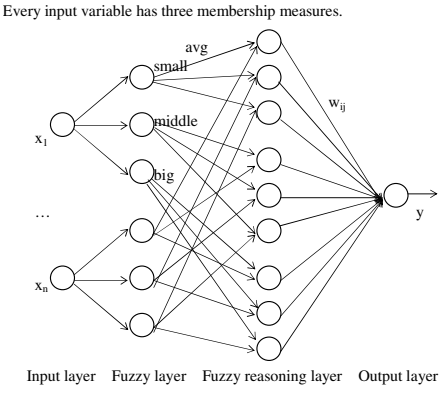


Fig. 2. Model of constructing single fuzzy neural network

(2) After treating as step (1), we can obtain training sample data excepting irrelevant variables, and classify train data set (Train: nt) and check data set (Check: nc):

$$C = \left\{ (x_i^c, y_i^c) \mid 1 \le i \le n_c \right\} \tag{1}$$

(3) And we cluster for above train data set, which are formed as r train data sets:

$$T^s = \left\{ (x_i^s, y_i^s) \mid 1 \le i \le n_t^s \right\}, 1 \le s \le r \tag{2}$$

After completing above steps, we can design fuzzy neural network NN_w and NN_1—NN_r making use of T^s and C respectively.

(4) We select the input vector in T^s as the input vector of NN_w, and establish its target output vector shown as follows:

$$W_i^s = (w_{i1}^s, ..., w_{ir}^s)^T, i = 1, ..., n_t^s, s = 1, ..., r \tag{3}$$

where, $w_{ik}^s = \begin{cases} 1 & (x_i^s, y_i^s) \in A^k \\ 0 & (x_i^s, y_i^s) \notin A^k \end{cases} \quad (k = 1, ..., r)$

We choice the training sample data set $T_w = \left\{ (x_i^s, W_i^s) \mid i = 1, ..., n_t^s, s = 1, ..., r \right\}$ for NN_w, and train the fuzzy neural network utilizing BP method. At last, we figure that the output of the NN is membership evaluation of antecedent of these fuzzy rules, when any vector x in input vector region is an input vector of NN_w.

For every fuzzy neural network $NN_s, s = 1, ..., r$, we design their forward feedback triple-tier BP neural network respectively.

(5) For a forward feedback triple-tier BP neural network NN_s, $T_i^s = \left\{ (x_i^s, y_i^s) \mid 1 \le i \le n_t^s \right\}$ is selected as training sample data, the input vector of

check sample data $x_i^c, i = 1,...,n_c$ is treated as input vector, and y_i^c is acted as target output, we train such neural network model and obtain mean variance ε_m^s.

(6) Suppose that the consequent part of rule is $THEN\ \mathrm{y} = \mathrm{F_s}(x_1,...,x_m)$, we remove randomly $x_p \in \{x_1,...,x_m\}$, and train the new neural network NN and infer to obtain mean variance $\varepsilon_{m-1}^{s,p}$ making use of check sample data according to step 5. If $\varepsilon_m^s > \varepsilon_{m-1}^{s,p}$, then it is possible that to remove input variable x_p affects hardly the consequent part of fuzzy rules, and x_p is removed from input vector.

(7) According to step 5 and step 6, we treat likely the rest input variable, until remaining input variable in input vector cannot be removed, and then we obtain optimal $\mathrm{NN_s}$.

(8) We establish the fuzzy reasoning system connecting all of fuzzy neural network models in term of the underside relation:

$$\mathrm{y}^* = \frac{\sum_{s=1}^{r} w^s y^s}{\sum_{s=1}^{r} w^s} \tag{4}$$

where $w = (w_1,...,w_r)^T$ and $y_s = (y_1^s,...,y_n^s)^T$ is output vector of $\mathrm{NN_w}$ and $\mathrm{NN_s}$, s＝1, …, r respectively, and y^* is total result of fuzzy reasoning system, and thus we constitute fuzzy reasoning neural network mode described as fig.1.

To sum up, the system can be widely applied to the reasoning of uncertainty process. In fact, the process of fuzzy reasoning is essentially system recognition for the process; on the other hand, it is practically model remembrance and recognition process in the view of network.

4 The Principle of Intrusion Detection Technique

Intrusion detection finished by SecAgent. SecAgent operate in the relevant server system among network, going on some network security inspection according to its own specialty. If find the online network security unusual, report to the blackboard system immediately at once. The blackboard system collects all the information that each agent reports, carry on reasoning and judging through the inference machine, confirm the event type , make corresponding decision and action , maintain the security of the network[4].

4.1 Scanning Agent

Scanning attack sets up connection with TCP/IP port of target host computer , and asks some services (such as TELNET , FTP, etc.), records the replying of the target

host computer , collects the relevant information of the target host computer, thus find the security weakness of the target host computer . Its characteristic is during shorter time, the attacker, who are from the same source address, attempt to connect a lot of target ports, which usually include the ports that are not in the monitoring state. when attacker give connecting request, these ports will send a TCP message with SYN token, scan Agent and count on these, sum up the number of those request messages which want connection to a host's different ports, refresh the statistic result at intervals of a certain time, if the result exceed the settlement constant allowed for a fixed period of time, the inference machine will think that the network is being scanned and attacking. Describe with the algorithmic language as follows:

Store target and source address that ask for connection in address table, each address corresponds to a Hash table and a count value. When data package reach, match with existing record in address table. If it succeeded to match, make a confirmation whether the location of the ports of Hash table have been replaced according to Hash table; if not, then the COUNT of these connected is increment and set this location; alert when COUNT is up to a certain limit ,refresh program refresh all the HASH table and COUNT with regularity according to REFRESH-RATE, COUNT deducts a constant while refreshing, until COUNT is less than or equal to zero, this item will be deleted from the table. Thus Scanning Agent only sensitive to large request of density, which pass from one source address to a certain target address, do not influence the normal access.

4.2 DoS Agent

Denial of service (DoS) attack are different from most other attack, they don't regard getting visiting authority of information on the network or the network itself as their purpose, but make the server unable to provide service for normal use. Most denial of service attack has the weakness of the whole structure of the system instead of the little defect of systems soft ware and application software or security loophole. It usually can be accomplished by the limited resource, such as exhausting network, operating system , application program, etc. DoS Agent finds the denial of service attack on the following corresponding rule that has been set up on unusual phenomenon.

(1) According to analysis, the attacker always analyzes the name of the target host computer before DDoS attacking. If an Agent detect that the DNS has received many PTR inquiring request about the inverse analysis of target IP of host computer name, then think that the network will be attacked by DDos.
(2) Agent calculates the extreme value corresponding with source address, when obviously beyond this extreme value, then indicates that the DoS attack exist.
(3) If Agent detects communication of unusual connection of TCP and UDP data package.
(4) The content of data segment only includes characters and digital data of character data package (for example, there are no blanks, punctuation mark and control characters).

(5) The content of data segment only includes binary scale and high-bit character of data package. It may transmit binary file at this moment, but if these data don't belonging to normal effective communication, Agent think it transmits the data package of controlling message communication that don't encode by BASE64 code but encrypt.

4.3 System Back Door Agents

This Agent checks the consistency of the protected host computer file in the system to find system back door. The system structures a database which includes a lot of file characteristics, such as the authority, node count, user, group, size of file, time, date and joining count etc. File characteristic also includes check-up sum of each file based on different algorithms (include shal, md5,rmd160,tiger).system back door Agents carry on following file characteristic inspection by comparing with this database each time, so as to find the changed file. The system back door, such as ROOTKIT which the invader installed in the systems can find quickly, finish the intrusion detection function based on host computer.

5 System Test and Result Analysis

For researcher and developer about IDS, they usually test performance of every kind of IDS, can help themselves understand status that technique have developed and these shortage, thereby would study those key and difficult technique problem weightily; Moreover, for IDS user, they can choose the product that may suit for them via by testing performance of IDS.

5.1 Training Sample Data

We collect the training sample data manually, in order to get over some difficult how to supervise learning and obtain these training sample data.

After applying to the actual system, BP neural network must be initialized. Commonly speaking, neural network need combined with learning phase and execution phase to treat the information included, and then it can implement a sound treatment process. NN's learning process for information is to obtain the feature of the information or to adapt to the information. And yet NN's execution process for information is to search its feature or to classify the information. By its learning phase, NN can be trained and be especially sensitive to some kind of information model, and thus can detect out the intrusion action, but it has two problems as follows: (1) it must be learned under the condition of supervision, i.e. it has the expected output knowledge for every input vector; (2) it is very difficult to obtain the training sample data corresponding to expected output information. Moreover, if abnormal input information includes known attack feature, NN will learn and be able to identify these feature, and yet it can not identify attack feature of other type.

In our test environment, we collect all of training sample data manually in the training process of NN, and then which can be solved two above problems. For every input vector of NN, it can composed of random data, and it will be tested through special assert supervised by NN. Furthermore, output expected by all random vector

that is supported by some above assert are set as “NORMAL”; other output are marked as “ABNORMAL”.

Therefore, we produce enough abundant random vectors, which collect 40000 to 60000 vectors in the test, and then our fuzzy neural network system can cover with abnormal and normal input model in n dimensions space, and thus NN can learn to distinguish normal model and abnormal model.

5.2 Attack Simulation

The attack simulation is the kernel of the testing environment. It’s also the key of testing IDS. The attack simulation must firstly collect enough attack methods. There’re many security website, BBS and mail list that publish lots of security knowledge. Some countries and companies have already built complete system vulnerabilities and attack database. All these are the source of attack simulation.

To implement the attack simulation, three stages must be performed: verification, analysis and automation. Verification is to check if the method is effective through some experiments. In the analysis stage, we analyze the character of attacks, add some information and parameter values into the FNN-IDS to simulate network and host attack methods, and do some hiding process for some attack methods. For complex attacks, we manually simulate.

5.3 Test Result and Analysis

In our test environment, we mainly investigate two targets of performance in FNN-IDS: the ability of intrusion detection and the rate of distort. The result is shown as follows in Table 1:

Table 1. Test results of FNN-IDS

Rate of distort %	Ability of intrusion detection%
17.6	83.5
12.1	74.3
6.5	67.2
1.4	35.7

Compared with current best intrusion detection system, which can obtain 60-70 percent of the rate of detecting intrusion, the system provides with obvious superiority. Moreover, it can be detected some unknown intrusion action.

The result indicates that the new method of intrusion detection, combined with fuzzy technology and neural network technology, can be response commendably to many abnormity states, and detect preferably many known attacks. It is possible that our system can bring alert information for unknown attack. What’s more, our intrusion detection system provides with definite scalability.

Indeed, actual environment is more complex than our test environment. Actual intrusion detection system need added attack type, and then adopt the design idea of layered and fuzzy multi-neural cell, and make use of multi-detector, and thus it can be provided with better sensitive measure and robustness.

6 Conclusion

As we all know, fuzzy membership function is the basis of fuzzy neural network, whose exact degree directly affects on whether our system can realize successfully or not, and that it's a complicated work how to obtain every factor in membership function, and thus we need increase continually the exactness of every membership function. At the same time, how to ensure better adaptability of IDS product that we design and how to make the best of detecting all kind of intrusion actions, needs our further study, and is the focal and difficult point of our next research.

Acknowledgement

Supported by Hunan Provincial Natural Science Foundation of China (03JJY3103).

References

1. Fang, B.: Constructing Frame of Computer Emergency Response Protecting the Cyber Security. Journal of China Institute of Commun. 23 (2002) 4-8
2. Jiang, J., Ma, H., Ren, D.: Net Security Intrusion Detection: A Research Summary. Journal of Software 11 (2004) 1460-1466
3. Li, Z., Li, J.: Application of Fuzzy Neural Networks to Intrusion Detection. Mini-Micro Systems 23 (2002) 1235-1238
4. Fei, Y., Li, R., Zhu, M., *et al.*: An Intrusion Detection System Based on FRete Net Algorithm. Proceedings of Fifth World Congress on Intelligent Control and Automation, Vol.5 (2004) 4362-4366,
5. Bace R G. Intrusion Detection. People's Posts & Telecom Press, Beijing (2001)
6. Li, Q., Zhang, D., Sun, Z., *et al*.: Intrusion Detection Method Based on Fuzzy Reasoning Driven by Neural Network. Computer Engineering 29 (2003) 133-135
7. Ouyang, M., Zhang, Y., Kpotchie, K., *et al.*: A Representing Matrix for Intrusion Signatures Based on Fuzzy Comprehensive Evaluation. Computer Engineering 29 (2003) 6-8

Intrusion Detection Using PCASOM Neural Networks

Guisong Liu and Zhang Yi

Computational Intelligence Laboratory,
School of Computer Science and Engineering,
University of Electronic Science and Technology of China,
Chengdu 610054, P.R. China
{lgs, zhangyi}@uestc.edu.cn

Abstract. This paper proposes a method to detect network intrusions by using the PCASOM (principal components analysis and self-organizing map) neural networks. A modified unsupervised learning algorithm which is more suitable for intrusion detection is presented. Experiments are carried out to illustrate the performance of the proposed method by using DARPA 1998 evaluation data sets. It shows that the proposed method can cluster the network connections into proper clusters with high detection rate and relatively low false alarm rate.

1 Introduction

Building of intrusion detection systems (IDSs) has been widely studied since the early 1980's. The challenges faced by designers increase as the targeted systems become more diverse and complex [1]. Intrusion detection has been proven to be a very valuable and powerful approach for network management as well as network security. Generally, there are two fundamental approaches used in intrusion detection technology: misuse detection and anomaly detection [2]. The misuse detection uses a signature-based database of well known intrusions and use a pattern matching scheme to detect intrusions. The anomaly detection, on the other hand, tries to quantify the normal operation of the host, or the network as a whole, with various parameters and looks for anomalous values for those parameters in real-time [3].

Artificial neural networks (ANNs) provide the potential to identify and classify network activity based on limited, incomplete, and nonlinear data sources [1]. The goal in using ANNs for intrusion detection is to be able to generalize from incomplete data and to be able to classify online data as being normal or intrusive [4]. Principal components analysis (PCA) [5, 6] and Self Organizing Feature Map (SOFM) [7] are wildly used for intrusion detection. Ezequiel López-Rubio et al. [8] proposed a new self-organizing neural network model that performs principal components analysis named PCASOM. In this paper, we use the PCASOM neural networks to to build an intrusion detection model.

This paper is organized as follows. Some basic knowledge of PCASOM algorithm is described in Section 2. The proposed algorithm for intrusion detection is

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 240–245, 2006.

given in Section 3. In Section 4, some experiments are carried out to validate the performance of the proposed method. Finally, conclusion is given in Section 5.

2 The PCASOM

It is known that in SOM the neurons are trained to model the input space and the information are stored in the weight connections of the neurons. PCASOM uses convariance matrix to store the information [8]. For the purpose of locating the winning neuron given by the data sample, PCASOM uses principal components analysis (PCA) to give the distance measure by computing the projection errors among all the neurons.

The covariance matrix of the input sequence vector $\{x\}$ is defined by $R = E[(x - E(x))(x - E(x))']$ [9]. Suppose we have M input samples at time t, the covariance matrix and expectation can be given as

$$R(t) = \frac{1}{M-1}\sum_{i=1}^{M}(x_i - E(x))(x_i - E(x))', \tag{1}$$

$$e(t) = E(x) = \frac{1}{M}\sum_{i=1}^{M} x_i. \tag{2}$$

For online computing, the covariance matrix cannot be calculated in advance. An iterative method for computing $R(t)$ and $e(t)$ is proposed in [8]. Suppose one new input connection x_{i+1} is given at time instant $t+1$, then,

$$R'(t+1) = (x_{i+1} - E(x))(x_{i+1} - E(x))', \tag{3}$$

$$e'(t+1) = x_{i+1}. \tag{4}$$

It follows from (1),(2),(3) and (4) that,

$$R(t+1) = \frac{1}{M}\sum_{i=1}^{M+1}(x_i - E(x))(x_i - E(x))' = \frac{1}{M}((M-1)R(t) + R'(t+1)), \tag{5}$$

$$e(t+1) = \frac{1}{M+1}\sum_{i=1}^{M}(x_i + x_{i+1}) = \frac{1}{M+1}(Me(t) + e'(t+1)). \tag{6}$$

Define learning rate $\eta_r = \frac{1}{M}$ and $\eta_e = \frac{1}{M+1}$, the iterative equations can be given by:

$$R(t+1) = R(t) + \eta_r(R'(t+1) - R(t)), \tag{7}$$

$$e(t+1) = e(t) + \eta_e(e'(t+1) - e(t)). \tag{8}$$

Clearly, $M(t+1) = M(t) + 1$, so the learning rate η_r and η_e are approaching to zero as time goes on.

We initialize the unit number as m in the model. For the unit j at time instant t, based on the theory of principal components analysis and subspace

decomposition [9], the K main components directions of the input vector x_i (network connection data) along which x_i has the largest deviation can be obtained (This means every unit has K neurons.The weights of the neurons represent the K principal components directions.). Those orthogonal vectors $B = \{b_i | i = 1, \ldots, K\}$ consist the feature space of the input network connections. The input vector error on the unit j can be easily defined by projecting it on the basis vector B_j as follows [8]:

$$\hat{x}_i^j(t) = \left\| x_i(t) - e^j(t) - \sum_{h=1}^{K} \left(b_h^{iT}(x_i(t) - e^j(t)) b_h^i \right) \right\|. \tag{9}$$

The winning unit has the minimum value $\hat{x}_i^j(t)$. For simplicity we assume the degree of neighborhood of the winning unit as one. The winning unit C can be obtained through competition process as follows:

$$C = \arg\min_j \left\{ \hat{x}_i^j(t) \right\}, j = 1, \cdots, m. \tag{10}$$

For every unit j, two small constants μ_r and μ_e are defined. If at time t, the new input x_{i+1} satisfies

$$||R'(t+1) - R(t)|| \leq \mu_r, \tag{11}$$

$$||e'(t+1) - e(t)|| \leq \mu_e, \tag{12}$$

we can ignore the influence of the new input to the winning unit and need not update the weights of the neurons in the unit. The training process of that unit will be stopped.

3 Algorithm Description

The proposed algorithm can be described as follows:

1. Preparing training and testing data. Decide the unit number m according to particular applications. Set the initial principal components number K.
2. For every unit j, initialize $R^j(0)$ and $e^j(0)$. $R^j(0)$ is a random symmetric nonnegative matrix by setting the main diagonal element near to one and all the others near to zero. Initial the vector $e^j(0)$ by using small random values near to zero [8].
3. Training process: at time t, input the training data. For every unit j, according to (3) and (4), compute $R'(t+1)$ and $e'(t+1)$.
4. Competition and updating: according to (9)and(10), find the winning unit j, then update $R^j(t+1)$ and $e^j(t+1)$ according to (7) and (8).
5. For the winning unit j, if (11) and (12) are satisfied, stop training process. Otherwise go to step 3. Until all the units satisfy (11) and (12), stop the whole training process.

4 Experiments

We use KDD Cup 1999 Data in our experiments [10]. It is a subversion of DARPA project [11]. The available database is made up of a large number of network connections related to normal and malicious traffic. Each connection includes forty-one feature values. We discard protocol-type, service and flag three pure symbolic features. Then we obtain the dataset with thirty-eight feature values.

We select $5,760$ ($1,740$ normal and $4,020$ intrusions) labeled connections randomly to test our IDS model. Besides normal, the other five popular type of attacks are included (such as smurf,neptune,back,guess-password and satan). The labeled value (data type) in the connections is discarded in our training process but just for experimental result evaluation. Gopi K. Kuchimanchi et al [5] use traditional PCA to reduce the dimensionality from forty-one to nineteen. We select the principal components number as twenty and the unit number as ten. The experimental results are shown in Table 1.

Table 1. The clustering results of PCASOM (20 PCs, 10 units)

clusters	1	2	3	4	5	6	7	8	9	10	detection
type	–	smurf	–	–	gpwd	–	neptune	normal	–	back	rate(%)
normal	3	26	0	0	1	1	32	1676	1	0	96.32
neptune	0	0	0	0	0	0	992	0	0	28	97.26
smurf	0	1000	0	0	0	0	0	0	0	0	100
gpwd	3	9	0	13	693	1	0	181	0	0	77
back	0	0	0	0	0	0	0	1	0	799	99.87
satan	0	1	4	0	0	0	295	0	0	0	0
accuracy(%)	–	96.53	–	–	99.86	–	75.21	90.20	–	96.61	–

Obviously, the selection of learning rate in the training process is very important for every unsupervised algorithm. In our experiments, the learning rate is defined as follows:

$$\eta_r^j(t) = \eta_e^j(t) = \frac{1}{1 + WinTimes(t,j)/p}.$$

For the winning unit j at time t, the winning times is recorded in $WinTimes(t,j)$. The parameter p is an integer type constant. The learning rate can be adjusted through varying p. In the above experiment, we selected p as 8 to obtain the best performance.

Table 1 shows that all the training data are clearly clustered into five clusters which have been labeled with type name using our label machine. The last row is named accuracy. For example,cluster two (smurf, 96.53%) means our model has detected $1,036$ connections as smurf, but only $1,000$ connections are real smurf (accuracy $= 1,000/1,036 * 100\%$). We can calculate total intrusion detection rate (IDR, as the view of two types, normal or intrusions), average detection rate (ADR) and false alarm rate (FAR), where

$$IDR = 1 - (1 + 181)/4020 * 100\% = 95.47\%,$$
$$ADR = (1676 + 992 + 1000 + 693 + 799)/5760 * 100\% = 89.58\%,$$
$$FAR = (1 - 1676/1740) * 100\% = 3.68\%.$$

The second step of the experiment is to test the performance of this model in different unit setup. We select the unit number as 6, 8, 10, 12, 15 respectively. Table 2 shows the results under different scenarios.

Table 2. The performance comparison using different unit number setup (20 PCs)

unit number	6	8	10	12	15
main clusters	5	5	5	7	7
$IDR(\%)$	95.57	95.52	95.47	94.60	95.52
$ADR(\%)$	84.97	82.69	89.58	87.80	84.30
$FAR(\%)$	3.68	23.68	3.68	2.70	17.70

In PCASOM model, the different unit setup will impact the performance as other clustering algorithms do. But Table 2 shows that the ADR and IDR vary little and maintain a high performance value under different scenarios. The main cause of FAR rising is that this model classifies the normal connections into two or three clusters. To solve the problem in actual intrusion detection system, we can use a anomaly-based classifier to separate the intrusions from normal first, then this model can be used to categorize those filtered attack data to avoid relative high false alarm rate.

Generally, supervised learning methods significantly outperform the unsupervised ones if the testing data contain no unknown attacks. But the performance of unsupervised learning is not impacted by unknown attacks. The following experiments demonstrate this standpoint. Some data of new type intrusion are added(700 new intrusive connections like *teardrop*) into the testing data, which never appears in the previous training. The model can group them as a new cluster (but not all other types of intrusion like this). The detection rate is 94.286% (660 of 700). The result shows that our model is adapted to novel intrusion detection with high accuracy.

In our experiments with different unit setup, the satan type intrusion cannot be separated from neptune. Maybe these tow type connections are too similar to separate by using this model. Hence, the distance measure methods deserve a more detail analysis. Another good way is using hybrid neural network [12]. These are all our future works.

5 Conclusions

The PCASOM is used in this paper to study the network-based intrusion detection. This model can cluster the input data as SOM with competition process based on principal components analysis. Unlike supervised learning algorithm,

this model has the ability to learn from unlabeled training data. It's helpful for data preparation, because it is difficult to label the network connections as normal or not in the real network environment.

The simulations shown that this model is adapted to both anomaly detection and misuse detection. Different unit number setup will impact the performance of ADR and IDR little. It can achieve high detection rate and relative low false alarm rate.

References

1. Cannady, J.: Artificial Neural Networks for Misuse Detection. Proceedings, National Information Systems Security Conference (NISSC'98), Arlington VA (1998) 443-456
2. Anderson, D., Frivold, T., Valdes, A.: Next-generation Intrusion Detection Expert System(NIDES): A Summary. SRI International Technical Report, SRI-CSL-95-07
3. Ramadas, M., Ostermann, S., Tjaden, B.: Detecting Anomalous Network Traffic with Self-organizing Maps. Lecture Notes in Computer Science, Vol. 2820. Springer-Verlag GmbH (2003) 36-54
4. Ghosh, A., Schwartzbard, A.: A Study in Using Neural Networks for Anomaly and Misuse Detection. In Proceedings of the Eighth USENIX Security Symposium (1999) 141-151
5. Kuchimanchi, G.K., Phoha, V.V., Balagami, K.S., Gaddam, S.R.: Dimension Reduction Using Feature Extraction Methods for Real-time Misuse Detection Systems. Proceedings of the 2004 IEEE Workshop on Information Assurance and Security (2004) 195-202
6. Labib, K., Vemuri, V. R.: Detecting and Visualizing Denial-of-Service and Network Probe Attacks Using Principal Component Analysis. Third Conference on Security and Network Architectures, La Londe, France (2004)
7. Lei, J.Z., Ghorbani, A.: Network Intrusion Detection Using an Improved Competitive Learning Neural Network. Second Annual Conference on Communication Networks and Services Research (2004) 190-197
8. Rubio, E.L., Prez, J. M., Antonio, J., Ruiz, G.: A Principal Components Analysis Self-organizing Map. Neural Networks 17(2) (2004) 261-270
9. Haykin, S.: Neural Networks: A Comprehensive Foundation. 2nd edn. Tsinghua University Press, Beijing (2001)
10. KDD Cup 1999 Data: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
11. DARPA Intrusion Detection Evaluation Project:http://www.ll.mit.edu/LST/ideval/
12. Pan, Z.S., Chen, S.C., Hu, G.B., Zhang, D.Q.: Hybrid Neural Network and C4.5 for Misuse Detection. Proceedings of the Second International Conference on Machine Learning and Cybernetics (2003) 2463-2467

A Mutated Intrusion Detection System Using Principal Component Analysis and Time Delay Neural Network

Byoung-Doo Kang, Jae-Won Lee, Jong-Ho Kim, O-Hwa Kwon,
Chi-Young Seong, Se-Myung Park, and Sang-Kyoon Kim

Department of Computer Science, Inje University, Kimhae, 621-749, Korea
{dewey, jwviolet, lucky, eujin13, cyseong,
smpark, skkim}@cs.inje.ac.kr

Abstract. The Intrusion Detection System (IDS) is generally used the misuse detection model based on rules because this model has low false alarm rates. However, the rule based IDSs are not efficient for mutated attacks, because they need additional rules for the variations of the attacks. In this paper, we propose an intrusion detection system using the Principal Component Analysis (PCA) and the Time Delay Neural Network (TDNN). Packets on the network can be considered as gray images of which pixels represent bytes of the packets. From these continuous packet images, we extract principal components. And these components are used as an input of a TDNN classifier that discriminates between normal and abnormal packet flows. The system deals well with various mutated attacks, as well as well-known attacks.

1 Introduction

The number of crimes committed, worm virus and denial of service attacks on a large scale, on the network is increasing under the rapidly development of the information technology [1].

The main purpose of IDS is protecting networking systems from the potential threat. Two kinds of the network based IDSs are the misuse detection and the anomaly detection. The misuse detection model detects intrusion when the acts of user are in perfect harmony the fixed model using the well-known attacks. The anomaly detection model analyzes patterns of users and regards them as intrusion if they are out the justified model [2]. Among these detection methods, misuse detection is used more often because of false alarm rates [3]. Misuse detection uses rule-based system and is efficient for well-known attacks. To deal with new types of attacks, an expert system must be supported. Since the system recognizes the signature which is absolutely matching to the rules as intrusion, it is not much flexible. Moreover, according to the limit of sniffing speed, it is impossible to collect and analyze the entire packets of the network [4]. Therefore, a method detecting mutated attacks and also with a part of packet flow is needed. In this paper, we propose IDS using PCA and TDNN.

Packet flow on the network is transformed to the packet image pattern using PCA which is one of the multivariate statistical analysis ways. PCA is concerned with explaining the variance-covariance structure through a few linear combinations of the

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 246 – 254, 2006.

original variables. Its general objectives are data reduction, and interpretation. These components from PCA are the linear combination of the packet information, and used for making the packet image pattern. We construct TDNN classifier to learn the image pattern about normal and abnormal flow. The proposed system can detect the ways of more various attacks because the system depends on the packet image patterns not on the particular values of the packet header.

2 The Structure of the IDS

The structure of the IDS using PCA and TDNN is as follows:

1. The packet collector accumulates the packets using libpcap library, and returns the packet structures which have time, length and raw data information.
2. From the packets, image patterns are extracted using PCA.
3. Packet image patterns are normalized and used for learning of TDNN classifier.
4. When the learning is accomplished, the classifier monitors network packets in real time, and discriminates the packet flow whether a normal packet or an abnormal packet. Because the classifier manages the packet as a continuous packet image pattern, it can detect intrusion with a portion of the attack image sequence.

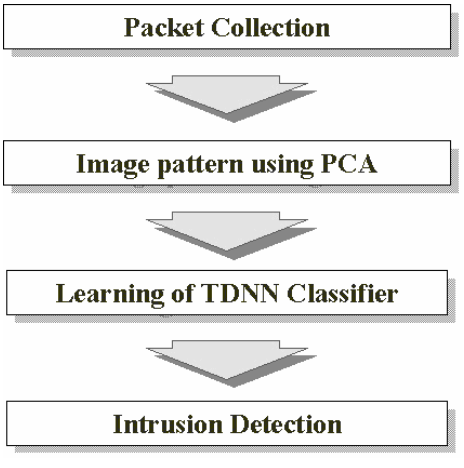


Fig. 1. The Structure of the IDS using PCA and TDNN

2.1 The Collection of Packet

We use libpcap library to accumulate packets. This library was developed at Berkeley Laboratory, University of California, USA. It is a public library to manage efficiently packet collection on the user level of the system. It provides the upper level interface of promiscuous mode to accumulate packets. Network packets can be accessed with the interface provided in the libpcap library. This library works on the most UNIX systems, and network monitoring tools like tcpdump use this packet collecting library [5, 6].

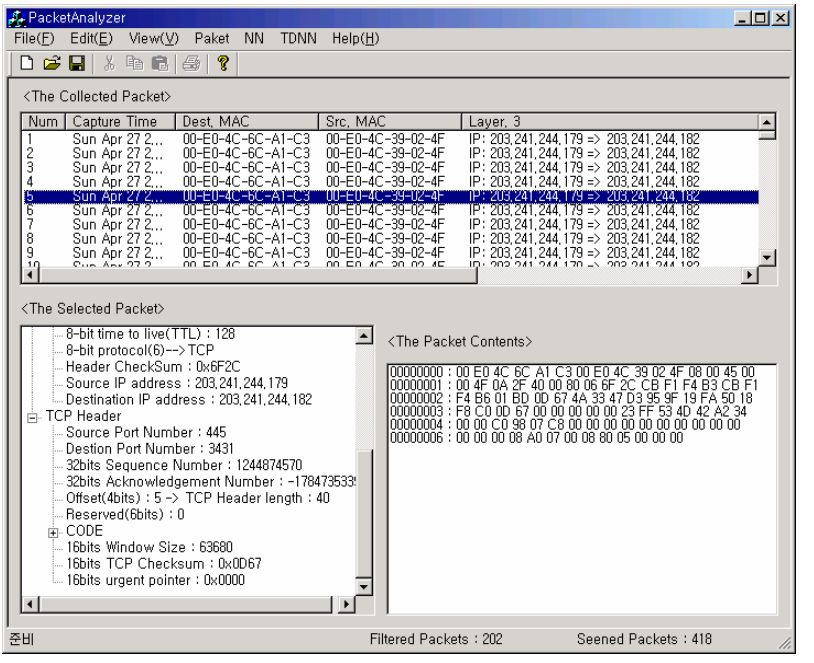


Fig. 2. The Collection of Packet

We collect the real time packets through a network environment composed of one attack system and five normal systems. The upper part of the Fig. 2 is the accumulated packet information. It denotes the collecting times and the information of the packet header. The bottom left side of the Fig. 2 shows the detailed information of the selected packet, and the right side denotes the packet contents with the hexadecimal notation.

2.2 Principal Component Analysis

PCA is a probability analyzing method which analyzes the relationships among multivariable, seeks the principal components denoted as a linear combination, and explains the entire changes with several components. The purpose is to make the effective explanations through dimension reduction using linear equations. Although p components are required to reproduce the total system variability, often much of this variability can be accounted for by a small number, k, of the principal components. If so, there is almost as much information in the k components as there is in the original p variables. The k principal components can then replace the initial p variables, and the original data set, consisting of n measurements on p variables, is reduced to one consisting of n measurements on k principal components. [7].

Let the training set of packet features be $\mathbf{a}_1,\mathbf{a}_2,\mathbf{a}_3,...,\mathbf{a}_M$. The average feature of the set is defined by $\mathbf{\Psi}=\frac{1}{M}\sum_{n=1}^{M}\mathbf{a}_n$, Each feature differs from the average by the vector $\mathbf{\Phi}_i=\mathbf{a}_i-\mathbf{\Psi}$. This set of vectors is then subject to principal component analysis, which

seeks a set of M orthonormal vectors, $\mathbf{u}_n$ which best describes the distribution of the data. The *k*th vector, $\mathbf{u}_k$, is chosen such that

$$\boldsymbol{\lambda}_k = \frac{1}{M}\sum_{n=1}^{M}\left(\mathbf{u}_k^T \boldsymbol{\Phi}_n\right)^2 \tag{1}$$

is a maximum, subject to

$$\mathbf{u}_l^T \mathbf{u}_k = \delta_{lk} = \begin{cases} 1, & if\ \ l=k \\ 0, & otherwise \end{cases} \tag{2}$$

The vectors $\mathbf{u}_k$ and scalars λ_k are the eigenvectors and eigenvalues, respectively, of covariance matrix

$$\begin{aligned} C &= \frac{1}{M}\sum_{n-1}^{M} \Phi_n \Phi_n^T \\ &= \mathbf{A}\mathbf{A}^T \end{aligned} \tag{3}$$

where the matrix $\mathbf{A} = \{\boldsymbol{\Phi}_1, \boldsymbol{\Phi}_2, ..., \boldsymbol{\Phi}_M\}$.
Consider the eigenvectors $\mathbf{v}_i$ of $\mathbf{A}^T\mathbf{A}$ such that

$$\mathbf{A}^T\mathbf{A}\mathbf{v}_i = \boldsymbol{\mu}_i \mathbf{v}_i \tag{4}$$

Premultiplying both sides by **A**, we have

$$\mathbf{A}\mathbf{A}^T\mathbf{A}\mathbf{v}_i = \boldsymbol{\mu}_i \mathbf{A}\mathbf{v}_i \tag{5}$$

From which we see that $\mathbf{A}\mathbf{v}_i$ are the eigenvectors of $C = \mathbf{A}\mathbf{A}^T$.

Following this analysis, we construct the M by M matrix $L = \mathbf{A}\mathbf{A}^T$, where $L_{mn} = \boldsymbol{\Phi}_m^T \boldsymbol{\Phi}_n$, and find the M eigenvectors, $\mathbf{v}_i$, of *L*. These vectors determine linear combinations of the M training set packet features to components $\mathbf{u}_l$.

$$\mathbf{u}_l = \sum_{k=1}^{M} \mathbf{v}_{lk} \boldsymbol{\Phi}_k, \tag{6}$$

In this paper, we used eight principal components that are more than the rate of the cumulative explanation, $p = \sum_{n=1}^{k}\lambda_n \big/ \sum_{n=1}^{M}\lambda_n$, 90%. And then, Packets are represented and combined on the space of principal component. This combination feature is used to input data of TDNN classifier.

2.3 The Packet Image Patterns

One packet image pattern represents sixty packets and the pattern consists of eight major principal components from the individual packet. Fig. 3 shows a spectrogram of a packet image pattern. The vertical line denotes one of the packet principal components. The number of horizontal lines is sixty.

Fig. 3. An Example of Packet Image Pattern

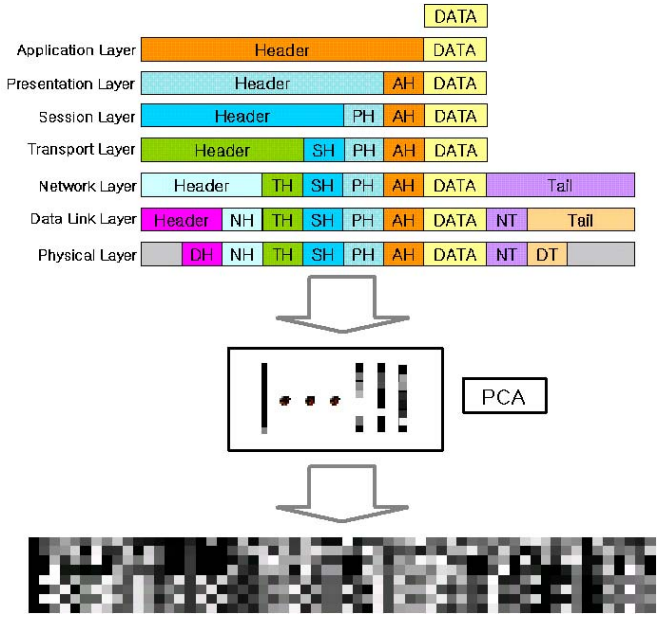


Fig. 4. The Changes of Learning Pattern

Fig. 4 shows the procedure of making from the collected packets into packet image pattern through PCA. The continuous packet image patterns are used as input of TDNN.

3 TDNN Classifier

Fig. 5 shows the entire structure of TDNN classifier. The kinds of TDNN learning algorithms are temporal back- propagation, back-propagation and more various ways. In this paper, we used the method of Waibel [8] who first created TDNN.

The TDNN is always fully connected between two layers in the direction of the spatial axis, and can use the partial connection in the direction of the time axis. Fig. 6 shows the side shape seen in the direction of the time axis of TDNN. The connection between the hidden layer and the input layer is partial. Each of the nodes' kernel size is five and the step size is three.

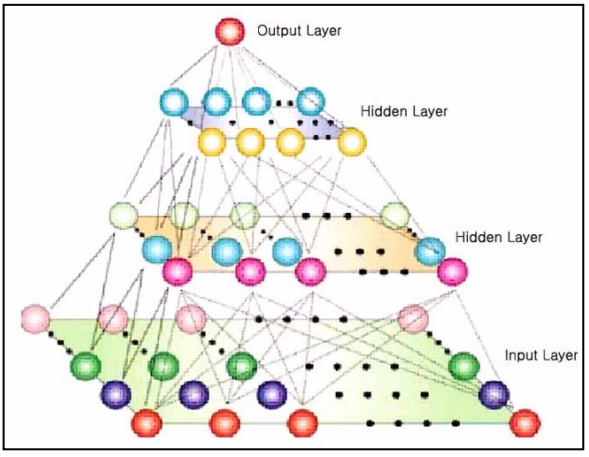


Fig. 5. Schematic Diagram of a TDNN

Table 1. The configuration of TDNN classifier

	Spatial Size	Kernel Size	Step Size	The number of node
Input Layer	8	0	0	60×8
Hidden Layer 1	4	4	2	29×4
Hidden Layer 2	2	5	2	13×2
Output Layer	1	13	0	1×1

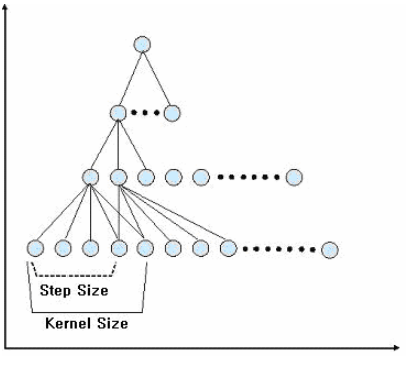


Fig. 6. Time Axis Section of a TDNN

The structure of the input layer has the most influence on deciding the structure of each level in the TDNN system. Also the decision of each level's temporal size and spatial size to accommodate the characteristics of the application data has influence

on the decision of the node number of the next layer. Table 1 shows the specific configuration for each level of TDNN classifier.

4 Experiment and Result

The system implemented with Visual C++ on the Pentium IV PC and Windows environment. We discriminated the data used in the learning and not used from each of the extracted packet, and experimented them to evaluate the proposed system. Six thousands attack packets and six thousands normal packets about each of SYN Flooding, Land, TearDrop and New TearDrop attacks were collected. They were mixed randomly and transformed to one hundred abnormal packet image patterns and one hundred pure normal packet image patterns for each attacks.

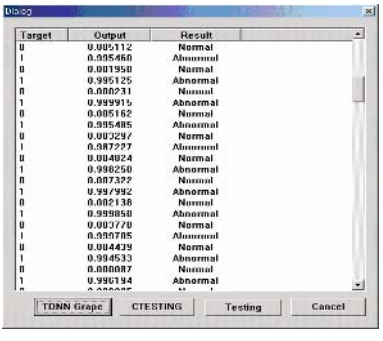
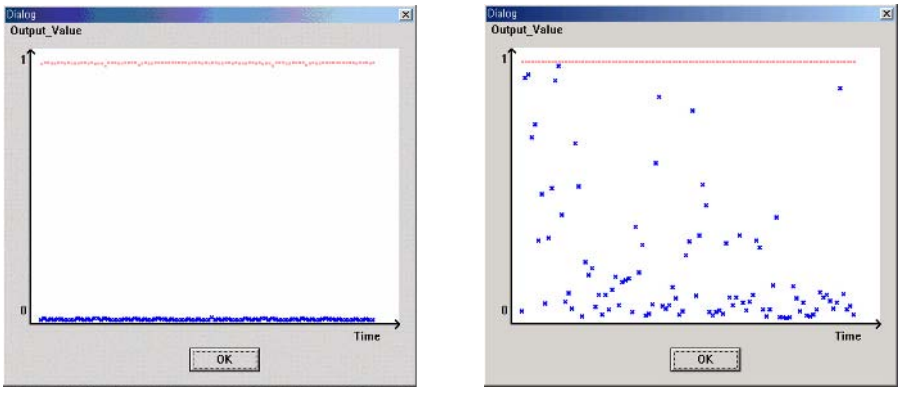


Fig. 7. The New TearDrop Result

Fig. 7 shows the result of testing TDNN classifier learned TearDrop attack with the mutated attack, New TearDrop. The target value is the expected data value, therefore



(a) (b)

Fig. 8. (a) The New TearDrop (through PCA), (b) The New TearDrop (not through PCA)

if it is zero, it is a normal data, but if it is one, it is an abnormal data. The items of the 'Result' represent the results of the TDNN classifier; 'Normal' means normal packet flow, and 'Abnormal' means attack.

Fig. 8 (a) shows the experiment results from the Fig. 7 in graph. The 'x' marks on the bottom represent the normal packets, and the '■' on the top represent the abnormal packets. Fig. 8 (b) shows the result of the packets which were not through PCA. It has more successful detecting rates when we experimented with eight principal components through PCA. More efficient results were obtained when the dimensions of the packets were reduced because the unnecessary elements were eliminated.

Table 2. The experimental results. Mutated Attacks(Learning TearDrop, Testing for New TearDrop).

	Rule-based IDSs		TDNN classifier	
	False positive packets(images)	False negative packets(images)	False positive packets(images)	False negative packets(images)
SYN Flooding	0	0	0	0
Land	0	0	0	0
TearDrop	0	0	0	0
New TearDrop	0	0	0	0
Mutated Attacks	0	6000(100)	0	1200(20)

Table 2 shows the result for each of the attack. The attacks such as SYN Flooding, Land, TearDrop and New TearDrop were detected correctly without false positive and false negative. Moreover, the TDNN classifier learning the TearDrop attack works well with the mutated attack of TearDrop, New TearDrop, except for twenty packet image pattern cases.

5 Conclusions

The rule-based IDSs are not able to detect intrusion unless it is satisfied with the conditions in the signature database justified with the detect module. For example, the rule-based system which has the signature database only for the TearDrop attacks, not for the New TearDrop attacks is not able to detect the New TearDrop attacks. Because TearDrop attack's signature database of the rule-based system is able to compare the length information of the only IP fragmented packets. However, New TearDrop attacks use both of IP and UDP, so that the rule-based system which does not examine UDP header information is not able to detect the New TearDrop attacks. The proposed IDS using PCA and TDNN can detect the New TearDrop attacks because the packet image patterns of them are similar to that of the TearDrop attacks.

References

1. Sekar, R., Guang, Y., Verma, S., and Shanbhag, T.: A High-Performance Network Intrusion Detection System. Proceedings of the 6th ACM Conference on Computer and Communications Security, ACM Press, New York, NY (1999) 8-17
2. Paxson, V.: Bro: A System for Detecting Network Intruders in Real-time. Computer Networks 31(23-24) (1999) 2435-2463
3. Giacinto, G., Roli, F., and Didaci, L.: Fusion of Multiple Classifiers for Intrusion Detection in Computer Networks. Pattern Recognition Letters 24(12) (2003) 1795-1803
4. http://www.snort.org
5. McCanne, S., and Jacobson, V.: The BSD Packet Filter: A New Architecture for User-level Packet Capture. In Proceedings of the 1993 Winter USENIX Conference (1993) 259-269
6. Quinn, B., and Shute, D.: Windows Sockets Network Programming. Addison-Wesley Professional (1996)
7. Johnson, R.A., and Wichern, D.W.: Applied Multivariate Statistical Analysis. Prentice Hall (2002) 356-395
8. Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., and Lang, K.J.: Phoneme Recognition Using Time-Delay Neural Networks. IEEE Transactions on Acoustics, Speech, and Signal Processing ASSP 37(3) (1989) 328-339

A Novel Intrusion Detection Model Based on Multi-layer Self-Organizing Maps and Principal Component Analysis

Jie Bai[1], Yu Wu[1], Guoyin Wang[1], Simon X. Yang[2], and Wenbin Qiu[1]

[1] Institute of Computer Science and Technology
Chongqing University of Posts and Telecommunications Chongqing, 400065, P.R. China
bibaboemail@gmail.com
[2] School of Engineering, University of Guelph,
Guelph, ON, Canada, N1G 2W1

Abstract. In this paper, the Self Organizing Maps (SOM) learning and classification algorithms are firstly modified. Then via the introduction of match-degree, reduction-rate and quantification error of reducing sample, a novel approach to intrusion detection based on Multi-layered modified SOM neural network and Principal Component Analysis (PCA) is proposed. In this model, PCA is applied to feature selection, and Multi-layered SOM is designed to subdivide the imprecise clustering in single-layered SOM layer by layer. Experimental results demonstrate that this model can provide a precise and efficient way for implementing the classifier in intrusion detection.

1 Introduction

Self Organizing Maps (SOM) is a competitive learning networks developed by Kohonen[5]. It has been applied to intrusion detection[2,3,6]. Forepart experiments demonstrate that satisfactory results can't be obtained if traditional SOM is directly applied to intrusion detection. One of main reasons is that SOM has an imprecise question in classification as is shown in Fig.1. It is evident from Fig.1 that the majority of neurons only have a cluster, but a few neurons still have various types. For example, a neuron marked underneath Fig.1 accounts for 12 normal samples, 11 portsweep attacks and 5 back attacks. Therefore traditional SOM can't be directly applied to intrusion detection.

On the other hand, the existing intrusion detection has two methods on feature selection of KDD99, which are selecting all the features[4] or extracting a fraction of features[3,6]. The former assures the accuracy of intrusion detection but increases processing overheads. Although the latter decreases processing overheads, the accuracy is not so good as the former. A balance therefore exists between the use of resources and the accuracy and timeliness of intrusion detection information. To achieve this, Multi-layered SOM and PCA are applied to the problem of intrusion detection on computer networks.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 255 – 260, 2006.

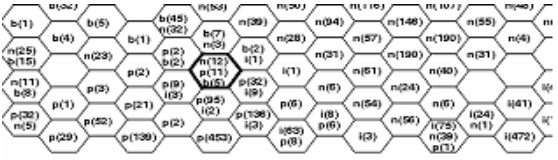


Fig. 1. Hit histogram of the output map. n denotes normal sample, b is back attack; p is portsweep attack, i is ipsweep attack, the number in the bracket is the number of input samples.

2 MPCA-MSOM Model and Algorithm

2.1 Intrusion Detection Model Based on MPCA-MSOM

Considering questions mentioned in the first section, a novel approach to intrusion detection based on MPCA_MSOM is proposed as shown in the Fig.2.

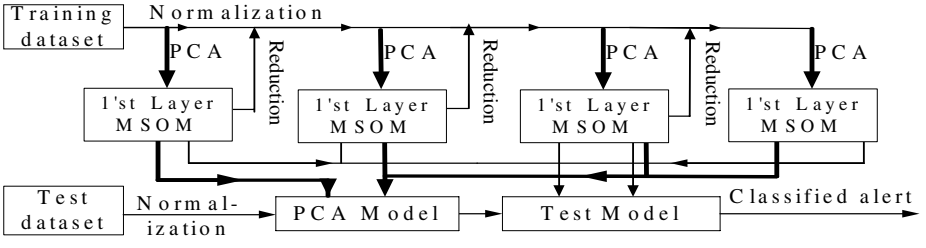


Fig. 2. Intrusion detection model based on MPCA_MSOM

In the MSOM module, a new neighboring function is introduced and used in the modified algorithms. PCA module is in charge of reducing dimensions of inputting samples and decreasing computational complexity efficiently in the model. The introduction of the hierarchy idea in the model can eliminate the high-precision classification in current layer and subdivide the low-precision classification in later layers.

2.2 Related Definitions of Multi-layer MSOM

Traditional SOM labeling method only labels winning neuron. But this method is incomprehensive because an input sample affects not only winning neuron but its neighboring neuron in different degree. So we think the method should have different effects on neighboring neurons. One neighboring function is introduced here.

Def. 1. Neighbor function is defined as:

$$h'(c,i,t) = \begin{cases} \exp(-d^2(c,i)/2r^2(t)), & r(t) \geq d(c,i); \\ 0, & r(t) < d(c,i). \end{cases} \tag{1}$$

Def. 2. Given $I_{all}=\{I_1,I_2,\ldots I_m\}$ is a set of all neurons, m denotes the number of neurons, $L=\{l_1,l_2,\ldots l_n\}$ is a set of all types in neuron I, H(I)is hit count of the i^{th} neuron which corresponds to all training samples, H_{all} is hit count of all neurons which corresponds to all input samples, $H_{li}(I)$ is hit counts of type l_I in neuron I, match_degree of type l_I in neuron I is defined as:

$$M(I,l_I) = H_{lI}(I)^2 / (H_{all} * H(I)) \tag{2}$$

Where $H(I) = \sum_{i=I,t\in T} h'(t,c,i)$, $H_{all} = \sum_{I\in I_{all}} H(I)$ and $H_{lI}(I) = \sum_{t\in T,i=I,L=l_I} h'(t,c,i)$.

Def. 3. Given $M_i(I,l_I)$ is match_degree , reduction_rate in neuron I is defined as:

$$R\_factor(I) = Max(M(I,l_I)) / \sum_{i=I,l_I\in L} M(I,l_I) \tag{3}$$

Def. 4. Quantification error of reducing sample is defined as:

$$Q\_error = \sum_{x_j\in C_i} \| w_i - x_j \| \tag{4}$$

Where i=1,2…,n and x_j are elements of the set of input vectors C_i satisfying $R\_factor_i > R_{value}$. Based on Def. 4, termination hierarchy criterion is defined as:

$$Q\_error_{i+1} < k \times Q\_error_i \tag{5}$$

2.3 MPAC-MSOM Algorithm

According to Def.1, MSOM algorithm needs to adjust hit count L_i in the neighborhood of the winning neuron by applying the following formula when SOM algorithm adjusts weights vector：

$$L_i(t+1) = L_i(t) + h'(c,i,t) \tag{6}$$

Algorithm 1. MPCA-MSOM training Algorithm

Input: Cumulative component value P_{value}, k among formula 5, reduction threshold R_{value} and the training dataset T_{set}.

Output: Weight matrix W_i, feature vector V_i and match_degree $M_i(I,l_I)$

Step 1. Read training samples T_{set0} , set up the number of layer, layer=1.

Step 2. Normalize training samples T_{seti} .

Step 3. Apply PCA to T_{seti} and save feature vector V_i according to P_{value} .

Step 4. Apply MSOM algorithm, and save weight matrix M_i in the i^{th} layer.

Step 5. Compute $M_i(I,l_I)$, reduction_rate $R\_factor_i(I)$ of each neuron in layer i.

Step 6. for each neuron I in the i^{th} layer, if $R\_factor_i(I) > R_{value}$, then $T_{seti} = T_{seti} - T_{redundancy}$ Where $T_{redundancy}$ is satisfied with $R\_factor_i(I) > R_{value}$.

Step 7. Computer quantification error of reducing sample Q_error_i in the i^{th} layer.

Step 8. If $Q\_error_{i+1} < k * Q\_error_i$, then layer=layer+1, repeat steps 2 – 8.

Algorithm 2. MPCA-MSOM test Algorithm

Input: Test dataset T_{test}, W_i, V_i and $M_i(I,l_I)$, where 1<=I<=layer.
Output: Classification alert.
Step 1. Set up initial value: layer i=1,Max=-1, Winner=0.
Step 2. Compute Principal Component of T_{test} according to V_i.
Step 3. Apply the Euclidean norm to find winner neuron c of T_{test} in the i^{th} layer.
Step 4. If Max<$M_i(I,l_I)$, then L_{num}=i, Max=$M_i(I,l_I)$, Winner=c;
Step 5. If i<layer, then i=i+1, Repeat steps 2 - 5.
Step 6. T_{test} belongs to clustering of Winner neuron in layer L_{num}.

3 Results

3.1 Test Scheme Description

In this paper, all tests adopt KDD-99 dataset, and experiments select 10000 normal samples and 1247 ipsweep attacks, 1040 portsweep attacks, 2203 back attacks as training dataset from 10% KDD, and select 1000 normal data, 306 ipsweep attacks, 354 portsweep attacks, 1098 back attacks from Corrected as test dataset 1. Five kinds of new attacks are added as test dataset 2 on the basis of test dataset 1, which include 158 httptuned attacks, 87 pod attacks, 9 xlock attacks, 17 sendmail attacks and 84 nmap attacks. All tests in this paper are based on the SOM toolbox and SOM-PAK.

Four test schemes are designed for testing comprehensively system performance. In scheme 1, the number of every layer neuron is satisfied with $M_{unit1} \cong \left\lceil 5\sqrt{D_{num}} / 4 \right\rceil$, where D_{num} denotes the length of input samples. The one of scheme 2, 3 and 4 is satisfied with $M_{unit2} \cong 0.5 * M_{unit1}$, $M_{unit3} \cong 0.25 * M_{unit1}$ and $M_{unit4} \cong 0.25 * M_{unit1}$. Cumulative component value P_{value} is initialized 0.6 in each scheme. Reduction threshold of schemes 1-3 is 1,and one of schemes 4 is 0.998.

3.2 Test Result and Performance Analysis

Table 1 and 2 detail detection rate (DR) and false positive rate (FPR) of MPCA-MSOM for four schemes on test dataset 1 and 2. The model not only obtains the better performance, but also each attack can be accurately classified.

Table 1. Test results of test dataset 1

	Scheme 1		Scheme 2		Scheme 3		Scheme 4	
Name	**DR**	**FPR**	**DR**	**FPR**	**DR**	**FPR**	**DR**	**FPR**
Normal	0.977	/	0.978	/	0.954	/	0.968	/
Back	0.985	0.004	0.979	0.004	0.913	0.004	0.969	0.004
Ipsweep	0.990	0.009	0.980	0.008	0.882	0.027	0.980	0.003
Portsweep	0.997	0.010	1.000	0.010	0.912	0.015	1.000	0.025
Total	0.988	0.023	0.984	0.022	0.909	0.046	0.978	0.032

Table 2. Test results of test dataset 2

	Scheme 1		Scheme 2		Scheme 3		Scheme 4	
Name	**DR**	**FPR**	**DR**	**FPR**	**DR**	**FPR**	**DR**	**FPR**
Normal	0.979	/	0.982	/	0.954	/	0.978	/
Back	0.994	0.003	0.985	0.004	0.954	0.004	0.971	0.003
Ipsweep	0.980	0.010	0.980	0.003	0.990	0.027	0.980	0.001
Portsweep	0.997	0.008	0.997	0.011	0.873	0.015	1.000	0.018
Httptunnel	0.873	/	0.880	/	0.892	/	0.956	/
Pod	0.966	/	0.954	/	0.540	/	0.919	/
Xlock	0.888	/	0.667	/	1.000	/	0.667	/
Sendmail	0.647	/	0.059	/	0.059	/	0.588	/
Nmap	1.000	/	1.000	/	0.012	/	1.000	/
Total	0.979	0.021	0.969	0.018	0.880	0.046	0.971	0.022

The termination hierarchy criterion is only applied to schemes 1, 2, 4 whereas the number of hierarchy is initialized to 4 in scheme 3. Tables 1-2 show that scheme 3 didn't obtain satisfying results when compared to the other schemes. From Fig.3, it can be observed that the trend of quantification error of reducing sample in scheme 3 is not satisfied with this criterion. The phenomenon demonstrates that the introduction of this criterion can make the performance of MPCA-MSOM more stable.

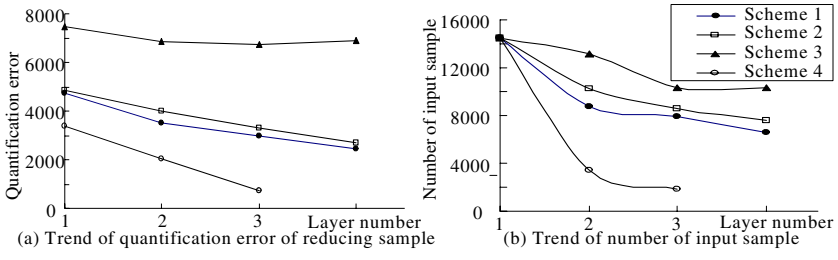


Fig. 3. Comparison of performances of four schemes

Fig.3 being analyzed, the question of scheme 3 can be ascribed to reduction threshold set up irrelevantly. To validate this conclusion, the number of neurons in scheme 4 is set up as same as one in scheme 3, and reduction threshold is modified to 0.998. Fig.3 shows that the drop of reduction rate makes input samples reduce effectively and accelerates the convergence of MPCA-MSOM model. Tables 1 and 2 show that scheme 4 obtain approving results for test dataset 1 and 2.

In Fig.4, scheme 3 hasn't perfect results, and training and test time of it both increase compared with scheme 2. Scheme 1 has the best performance whereas its training time is the longest. Scheme 4 makes the training time fall dramatically by adjusting reduction threshold and the performance of scheme 4 is close to scheme 1.

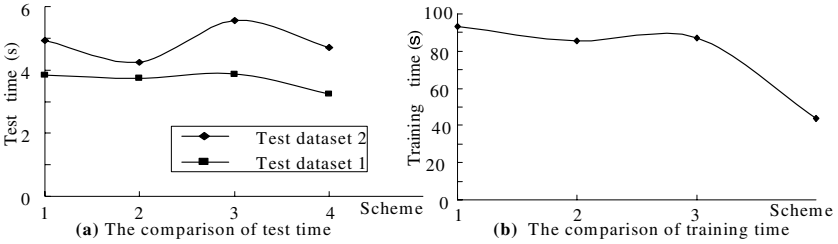


Fig. 4. Comparison of time of four schemes

Table 3 provides a summary of some recent results based on the same dataset. It can be easily found that MPCA-MSOM can not only improve detection rate and false positive rate, but also carry out relatively accurate classification of attacks.

Table 3. Performance comparison to the other methods

	K-NN	RoughSet [1]	SOM [2]	MPCA-MSOM
DR	88.7%	93.5%	96.1%	97.0%
FPR	9.1%	2.8%	7.8%	2.2%

4 Conclusions

The idea of subdivision layer by layer solves imprecise questions in classification of traditional SOM. The introduction of PCA can reduce the computational complexity. Experimental results demonstrate that MSOM-MPCA can provide a precise and efficient way for implementing the classifier in intrusion detection.

Acknowledgements. This work is partially supported by National Natural Science Foundation of P.R. China (No.60373111), Program for New Century Excellent Talents in University, Key Science & Technology Research Foundation of Chongqing of China and Science & Technology Research Foundation of Chongqing of China.

References

1. Li, Z.J., Wu, Y., Wang, G.Y.: A new framework for intrusion detection based on rough set theory. Proceedings of SPIE Vol.5433. (2004) 122-130
2. Lin, S.C., Shi, D.H.: A Study of Intrusion Detection System Based on Anomaly Detection in Windows Environment, Master thesis, Taiwan of China (2004)
3. Lichodzijewski P.: Dynamic Intrusion Detection Using Self Organizing Maps. 14th Annual Canadian Information Technology Security Symposium (2002)
4. Eskin, E.A. (ed.): A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data. Data Mining for Security Applications, Kluwer (2002)
5. Kohonen, T.: The Self-Organizing Map. Proceedings of the IEEE 78(9) (1990)
6. Kayacik, H.G., Zincir Heywood, A.N., Heywood, M.I.: On Dataset Biases in a Learning System with Minimum a priori Information for Intrusion Detection. Proceedings of the IEEE CNSR, Fredericton, Canada (2004) 181-189

A Modified RBF Neural Network for Network Anomaly Detection

Xiaotao Wei, Houkuan Huang, and Shengfeng Tian

School of software, School of Computer and Information Technology,
Beijing Jiaotong University, Beijing, 100044, China
{weixt, hkhuang, sftian}@center.njtu.edu.cn

Abstract. A modified RBF (radial basis function)-based neural network is proposed for network anomaly detection. Special attention is given to the determination of the parameters of the hidden layer. We propose a novel grid-based approach to compress and cluster the training data. The number, center and radii of the RBFs are determined according to the clustering result. At the detecting stage, we expand each input node with a sigmoid function to meet the type of input data. Experimental result on KDD 99 intrusion detection datasets shows that our RBF based IDS has high detection rate while maintaining a low false positive rate. It also shows the remarkable ability of our IDS to detect new type of attacks.

1 Introduction

An intrusion detection system (IDS) is an automated system for the detection of computer system intrusions. There are two main classifications of IDSs. The first one divides the techniques of intrusion detection into anomaly and misuse detection. The anomaly detection approach establishes the profiles of normal activities of users, systems, system resources, network traffic and/or services and detects intrusions by identifying significant deviations from the normal behavior patterns. The misuse detection approach defines suspicious misuse signatures based on known system vulnerabilities and a security policy. According to the difference in monitoring objects, IDSs are divided into network-based IDSs and host-based IDSs. In this paper, we focus on network anomaly detection (NAD).

RBFNNs (Radial Basis Function Neural Networks) have been widely used because of their simpler architecture, quick learning and local learning characteristics. Closely related to our investigation, Yang et al. [1] applied RBF to network intrusion detection. They proved that RBFNN has better property of approximation than BP neural network. Jiang et al. [2] proposed a hierarchical IDS based on RBF for real-time detection of network traffic. Both of [1] and [2] test their IDS with a customized sub dataset of KDD 99 intrusion detection datasets. In this paper, a modified RBF neural network is employed to identify the abnormal records in network traffic. We propose a novel grid-based method to cluster the training data. And the number, center and radii of the hidden nodes of the RBFNN are determined according to the clustering

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 261 – 266, 2006.

result. We train our IDS on the KDD 99 intrusion detection training 10% dataset and test it with the test dataset. Experiments show that this method is simple and efficient.

The rest of this paper is organized as follows. Section 2 describes our modified RBFNN for NAD. In section 3, we describe the KDD 99 intrusion detection datasets and the experimental result. Finally, Some conclusions are given in Section 4.

2 RBF Neural Network for Network Anomaly Detection

2.1 Brief Introduction of RBF Neural Network

An RBF classifier has an architecture very similar to that of a traditional three-layer BP (Back Propagation) network. Connections between the input and hidden layer have unit weights and, as a result, do not have to be trained. Nodes in the hidden layer, called BF nodes, produce a localized response to the input. The basis functions used are Gaussians, where the activation level y_i of the hidden unit $\boldsymbol{i}$ is given by:

$$y_i = \phi_i(\|X - u_i\|) = \exp(-\sum_{k=1}^{D} \frac{x_k - u_{ik}}{2h\delta_{ik}^2}) \tag{1}$$

Where $\boldsymbol{h}$ is a proportionality constant for the variance, x_k is the kth component of the input vector $X = [x_1, x_2, ..., x_D]$, and u_{ik} and δ_{ik}^2 are the kth components of the mean and variance vectors, respectively, of basis function node $\boldsymbol{i}$. Each hidden unit can be viewed as a localized receptive field.

A two-stage learning framework is traditionally employed to train RBF networks. It combines an unsupervised learning strategy for the RBF hidden layer with a supervised learning mechanism for the RBF output layer. In the first stage, the hidden nodes' parameters can be determined by the prior understanding of training data. Thus the hidden nodes of an RBFNN can be understood intuitively to reduce the uncertainty during the design procedure. But the disadvantage of RBFNN is that the classification error strongly depends on the selection of the number, centers and widths of basis functions. So it is a key task to predetermine these parameters.

2.2 Pretreatment of the Training Data

There are two reasons for preprocessing and analyzing the training data: i) We want to learn the character of the training data in order to decide the number of hidden nodes. ii) The training network traffic data are too huge for quick learning. A good method to compress the training data is the grid-based method which divide the object space into finite number of cells called hyper-rectangles or units [3]. The main advantage of this approach is its fast processing time, which is typically dependent mainly on the number of cells in each dimension in the domain space.

For numeric attributes, the dimensions are commonly divided into equal length partitions. But it does not work on network traffic data. For example, in KDD 99 intrusion detection training 10% dataset, the values of 'duration' range from 0 to 58329. The average duration of normal records is 217.82 and the attack records is

6.34. When dividing this dimension into 10 equal length intervals, it seems that most records fall into the first interval. And this attribute lost its ability for classification.

We propose a novel method, called S-function projection method, to divide the range of each dimension into 10 non-overlapping intervals that are not of equal length, and designate each interval by the value of the low boundary of the interval. The attribute values are projected into 10 unequal intervals according to a sigmoid function defined as (2).

$$f(x)=\left\lfloor \frac{10}{1+e^{-\alpha(x-m)/c}} \right\rfloor \tag{2}$$

where α is a constant. Let m_n and d_n denote the average value and standard deviation of normal records on this dimension, m_a and d_a denote the average value and standard deviation of attack records on this dimension. m and c are defined as: $m=\frac{1}{2}(m_n+m_a)$, $c=\frac{1}{2}(d_n+d_a+|m_n-m_a|)$.

The main idea of our method is that near the boundary of the two classes we split the dimension into narrow intervals and at the other area we split the dimension into wide intervals.

In network connection records, there are also symbolic attributes, such as 'protocol'. For these attributes we map the values to integers. The range of the dimension is equal to the number of discrete values in the attribute. Therefore, the number of intervals for these dimensions is also equal to the range of the dimension.

Let α=2.5. By the algorithm described above, we project the KDD 99 training 10% data (494021 records) into 3380 units with only 13 units containing both normal records and attack records, witch totally involving 2812 records (1304 normal records and 1508 attack records). We label these units with the majority label in the units. If we define the dense-unit as the unit that containing at least 3 records, we got 1245 dense-units among the 3380 units. So our training data are compressed into 3380 records. It greatly reduces the computational time of learning.

2.3 The Structure and Parameters of the RBFNN

There are 3 common ways to determine the number of RBFs. It can be determined by:

1. the prior knowledge of training data,
2. the result of an unsupervised clustering of the training data, or
3. assigning an RBF to each training data.

For network intrusion detection, [2] proposed a RBFNN that has 4 hidden nodes because their customized training data contain the samples that belong to the four classes namely: NORMAL, PROBE, DoS and R2L. But the distribution of the 4 classes in the input space may not form into spherical shape. In this case, it is difficult to determine the center and radii of each RBF.

As for 2, the distribution of RBFs in the training data space as computed by the unsupervised learning technique does not reflect the local complexity of the classification problem at hand. For example, unsupervised methods may form mixed clusters of training data that are closely spaced but belong to different classes.

So we adopt a method similar to 3. Because our training data is now compressed into some thousand units, and the dense units are only 1245. We can assign an RBF to each dense-unit, and use the unit's center as the RBF center and its size s scaled by a parameter λ as the RBF radius, $r = \lambda s$. The scalar λ has the same value for all nodes. For anomaly detection, the RBFNN need only one output node. The input layer has 41 nodes because each connection record in KDD 99 training dataset consists of 41 different attributes. Figure 1 shows the structure of our RBFNN at the training stage.

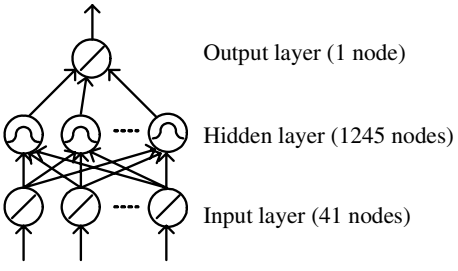


Fig. 1. The structure of RBFNN at the training stage

2.4 Run-Time RBFNN for Network Anomaly Detection

At the detecting stage, testing data should be transformed to a unit-form data to input to the RBFNN. So we expand each input node with a sigmoid node. The output of each sigmoid node is calculated by Formula (2). The parameters are the same as used in training stage. Figure 2 shows the modified RBFNN at the detecting stage.

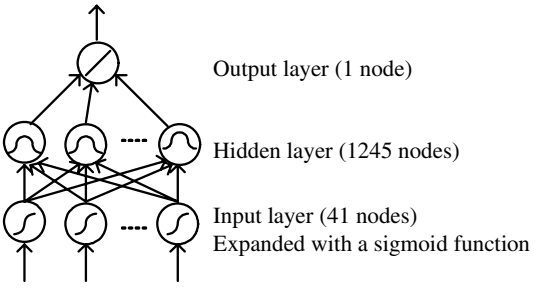


Fig. 2. The structure of RBFNN at the detecting stage

3 Experiments and Results

3.1 Description of KDD 99 Intrusion Detection Datasets

The KDD 99 training dataset[4] contained about 5,000,000 connection records, and the training 10% dataset consisted of 494,021 records among which there were 97,278 normal connections (i.e. 19.69%). Each connection record consists of 41 different attributes that describe the different features of the corresponding connection, and the

value of the connection is labeled either as an attack with one specific attack type, or as normal. There are 22 different attack types present in the 10% datasets. Each attack type falls exactly into one of the following four categories: Probing, DOS, U2R and R2L.

The task was to predict the value of each connection (normal or one of the above attack categories) for each of the connection record of the test dataset containing 311,029 connections. Among the 37 types of attacks in the test dataset, there are 17 new types of attacks that are unseen in the training dataset.

We use the training 10% dataset as training data and test the IDS with the 'Corrected' test dataset.

3.2 The Result of the Experiment

Firstly, the training data are projected to units by the grid-based method mentioned in section 2.2. Then we input these units as training data to our RBF based IDS. Test data are directly input into the expanded RBFNN for classification. We try different values for the parameter λ with 0.1, 0.4, 0.6 and 1.5. The experiment results are listed as Table 1.

Table 1. The result of the experiments

λ	0.1	0.4	0.6	1.5
Detection Rate	95.685	95.651	94.419	93.824
False Positive Rate	6.129	5.066	4.065	2.687

Table 2. Detection rate for novel attacks (λ=0.4)

Attack name	Total instance	Detected instance	DR
apache2.	794	792	99.748%
httptunnel.	158	158	100.000%
mailbomb.	5000	5000	100.000%
mscan.	1053	1053	100.000%
named.	17	16	94.118%
processtable.	759	759	100.000%
ps.	16	15	93.750%
saint.	736	139	18.886%
sendmail.	17	11	64.706%
snmpgetattack.	7741	0	0.000%
snmpguess.	2406	2	0.083%
sqlattack.	2	2	100.000%
udpstorm.	2	1	50.000%
worm.	2	2	100.000%
xlock.	9	9	100.000%
xsnoop.	4	4	100.000%
xterm.	13	13	100.000%
		Average:	77.723%

There are 37 types of attacks in the test dataset. Among them, 17 new types of attacks are unseen in the training dataset. Our IDS discovered 16 out of the 17. The performance on new attacks is detailed in Table 2.

4 Conclusion

An RBF based IDS is proposed for network anomaly detection. A novel grid-based algorithm is employed to project the training data to units. The dense units are used to determine the parameters of RBFs. Experimental results proved that this method is simple and efficient.

Acknowledgements

This paper is supported by the National Natural Science Foundation of China (No. 60442002) and the Science and Technology Foundation of Beijing Jiaotong University (No. 2004SM010).

References

1. Yang, Z.M., Wei, X.M., Bi, L.Y., Shi, D.P., Li, H.: An Intrusion Detection System Based in RBF Neural Network. In: Proceedings of the 9th International Conference on Computer Supported Cooperative Work in Design. Coventry, UK (2005) 873-875
2. Jiang, J., Zhang, C.L., Kamel M.: RBF-based Real-time Hierarchical Intrusion Detection Systems. In: Proceedings of the International Joint Conference on Neural Networks. Oregon, USA (2003) 1512-1516
3. Wang, W., Yang, J., Muntz, R.R.: Sting: A Statistical Information Grid Approach to Spatial Data Mining. In: Proceedings of the 23rd International Conference on Very Large Data Bases. Athens, Greece (1997) 186-195
4. KDD Cup 1999 Data. Machine Learning for Intrusion Detection Project. The University of Columbia (2000) http://www.cs.columbia.edu/ids/ml/

Anti-worm Immunization of Web System Based on Normal Model and BP Neural Network

Tao Gong and Zixing Cai

College of Information Science and Engineering, Central South University, Changsha, Hunan 410083, China
taogong@sigmaxi.org

Abstract. Pattern recognition and learning of unknown worms have become a bottleneck of network security since a lot of variants of old worms and new worms occurred. To overcome this bottleneck, many traditional approaches were tested but failed. In this paper, a normal model of a web system was proposed to detect all selfs and all non-selfs, especially all unknown worms. The normal model was built on the 2-dimension attributes of space and time of the system. Moreover, a BP neural network was used to design an adaptive learning mechanism of the immunized web system. The non-self learning was utilized to recognize most unknown worms through the trained BP network, which was trained with the feature data in the worm database. Besides, the innate non-self selection was designed to recognize all known worms. Experiments validated effectiveness of this approach on the BP network and the normal model.

1 Introduction

More and more worms are dangerous to many networks, especially such as the Internet, though the early worm was designed for harnessing the (possible heterogenous) nodes of a multicomputer (networked computers) [1]. Staniford et al. described a worm that could infect the entire Internet in about 30 s [2]. A worm of this scale and speed could bring the entire network to a halt, or worse [3]. To prevent the spread and damage of the worms, some network techniques were used to detect and stop the propagation of the worms [4], [5], [6]. But most of these approaches failed to detect and recognize unknown worms, because the variants of old worms and new worms occurred continuously. The detection and recognition of unknown worms become a bottleneck of this problem. To overcome the bottleneck, an adaptive immunization approach was proposed for a web system on a BP neural network (BPNN) and the normal model.

Part of the reason of the BP neural network was the effectiveness of learning unknown objects. The BP network was fit to learn optimal parameter offline and some adaptive factors were added into the BP algorithm to improve its optimization. On the other hand, part of the reason of the normal model was the full detection of all selfs of the web system and its non-selfs. Inspired from the natural immune system, self/non-self discrimination of the web system was built on its normal model, which was represented with the space-time features of its all normal components as similar to the

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 267 – 272, 2006.

special molecule structures of all normal cells of the body. All the 2-dimension features were represented and stored in the self database. Thus, all normal components were found in the self database.

2 Normal Model of Web System

Suppose a web system *S* is comprised of 3 web directories and 100 HTML page files, then the file set of the system is represented as such.

$$\{p_{ij} \Big| \sum_{i=1}^{3} n_i = 100, p_{ij} \in S, i \in \{1,2,3\}, j = 1,2,\cdots,n_i\} \tag{1}$$

Here, p_{ij} denotes the *j*th file in the *i*th directory of the system *S*, and n_i denotes the sum of all files in the *i*th directory of the system *S*.

Thus, the normal states of the web system are represented with the normal states of all components in the web system. Inspired from the normal states of the human immune system, the normal states of the web system are represented on the space and time dimensions of its normal components, to identify its self components. The space dimension d_i of every component p_i is presented as its absolute pathname. The time dimension t_i of every component p_i is presented as its last revision time.

$$< D,T >= \{(d_i,t_i) \Big| d_i \in p_i, t_i \in p_i, i = 1,2,\cdots,n\} \tag{2}$$

Here, *n* denotes the sum of all the normal components of the web system *S*.

Part of the reason of the 2-dimension modeling is the time-space positioning thought of Einstein [7]. In today's cyberspace, every component can be also uniquely identified with both its space dimension and its time dimension. If and only if every normal component of the normal web system is identified uniquely, the normal state of the system can be identified uniquely.

Theorem 1. Suppose a web system *S* is comprised of *n* components, each of which has its unique absolute pathname and unique last revision time. Let the absolute pathname of a component p_i be d_{p_i} and the last revision time of the component p_i be t_{p_i}. Also suppose the cyberspace has the same order of space and time as nature and human society. Thus, the absolute pathnames and last revision time of all normal components in the normal system uniquely identify the normal states of the system.

Proof. In the web system there are *n* components of web pages, which uniquely identify the system. In other words, if the components are all normal, then the web system is normal; but if any component is abnormal, then the web system is abnormal. Therefore, the normal state of every component in the web system should be identified with its space feature and time feature before the whole web system is identified.

(1) Given an absolute pathname d_{p_i}, its corresponding component p_i is unique. Again given the last revision time t_{p_i} of the component p_i, its corresponding state is

also unique. Moreover, the last revision time t_{p_i} is acquired when the web system is normal, i.e. the component p_i is also normal at that time. Therefore, the absolute pathname d_{p_i} and the last revision time t_{p_i} identify the normal state of the unique component p_i in both the space and time dimensions.

(2) Given a component p_i of the web system *S*, the absolute pathname d_{p_i} of the component is unique, because this fact is required by the rules of file management on any operation system. Besides, according to the hypothesis of this theorem, the last revision time t_{p_i} of the component is also unique, because the last revision time of everything is unique in the real world.

Therefore, the absolute pathname and last revision time of every normal component in the normal web system uniquely identify the normal state of the component. Based on the unique identification between the web system and its components, the normal states of the web system are uniquely identified through the absolute pathname and last revision time of its all normal components. Thus, the proof of Theorem 1 is accomplished.

Thus, all the normal states of the web system are represented as the above set of 2-element group, and the data of the model can be visualized as some molecules in a kind of 3-dimension space. On the Java platform, the absolute pathname and the last revision time of the component can be read through two methods of the class *File*. Modeling the normal states of the web system in 2 dimensions is shown in Fig. 1.

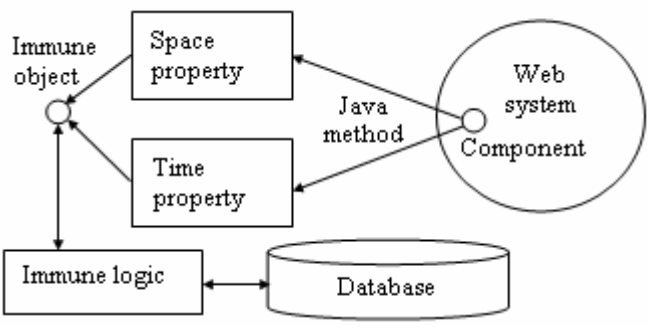


Fig. 1. Modeling normal states of the web system

Every component of the normal web system has its space property and time property, i.e. its absolute pathname and last revision time. The two properties can be read through Java method in the Java class package File, and the two properties of every component are also encapsulated as an immune object. The immune object is utilized in the immune logic when the web system is abnormal. Besides, the immune object is stored and accessed in the self database.

The reason of modeling the normal states of the web system is the good test-bed of the immune model [8]. Thus, it is every fit for building its normal model and artificial immune system with pure Java, which is easy for developing web applications.

3 BP Neural Network and Immunization of Web System

In the natural immune system, adaptive learning of immune cells against unknown viruses is a kind of very complex process, which is even known little by doctors and immunologists. Alike, the adaptive learning of the artificial immune system against unknown worms is also very complex and is still a hard problem. To explore the secret, a BP neural network is built and used for the adaptive learning, and then the web system is immunized on its normal model and the BP neural network.

The BP neural network consists of three tiers, i.e. input tier, hidden tier and output tier. In the input tier, 6 features of the known worms are represented as $\{x_i | i = 1,2,\cdots 6\}$, and these features include type, coding language, proliferation manner, engine, feature string and damage. In the hidden tier, the names of 3 known worms are represented as $\{x_i | i = 7,8,9\}$. In the output tier, 3 elimination schemas of worms are represented as $\{x_i | i = 10,11,12\}$. Thus, the BP network is constructed as shown in Fig. 2.

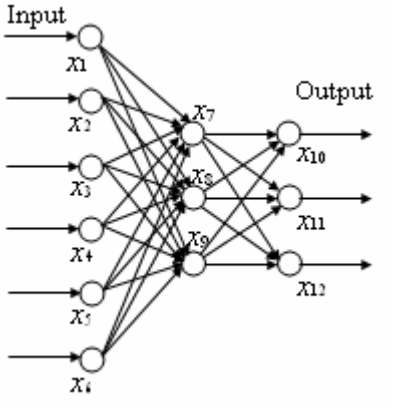


Fig. 2. Structure of BP neural network

The input data are acquired from many samples of all known worms in the worm database, which is used to represent and store the features of all known worms. The three known worms are the love worm, the happy-time worm and the code-red worm. The three worms have different elimination schemas, which are also represented and stored in the worm database. The BP neural network satisfies the following formulas.

$$P = \sum_{y} (\sum_{z} (d_{yz} - O_{yz})^2), \qquad \Delta W_{i \to j} = r O_j (1 - O_j) \beta_j$$
$$\beta_j = \sum_{k} W_{j \to k} O_k (1 - O_k) \beta_k, \qquad \varepsilon_z = d_z - O_z \tag{3}$$

Here, P represents the performance of the BP neural network; y represents the training input; z represents the output node; d_{yz} represents the anticipant output of

the node z through the input y, and d_j represents the anticipant output of the *j*th node; O_{yz} represents the actual output of the node z through the input y, and O_j represents the actual output of the *j*th node; $W_{i\to j}$ represents the weight value between the nodes of the *i*th tier and those of the *j*th tier, and $\Delta W_{i\to j}$ represents its change; r represents the learning rate; β_j represents the value of the *j*th node, ε_z represents the error of the output node.

The BP neural network is trained for 2117 times with the input data of all known worms, and the learning result is shown in Fig. 3.

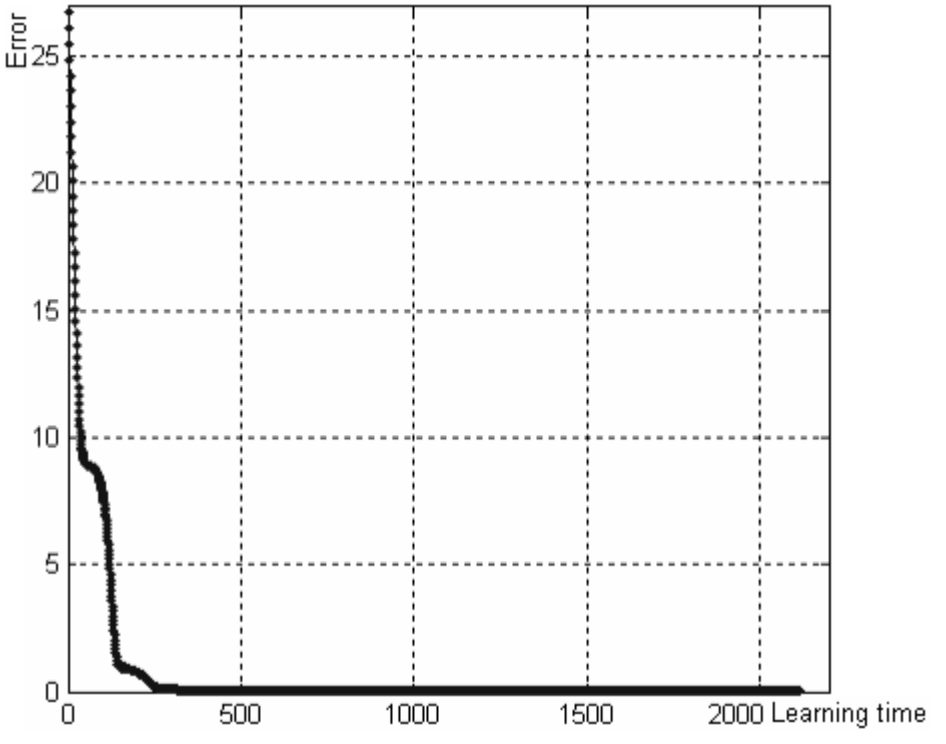


Fig. 3. Adaptive learning result of BP neural network

The adaptive learning of the BP neural network is improved through controlling the momentum of the learning rate, and used to recognize learn and memorize unknown worms from the unknown non-selves. Besides, all known worms can be recognized through querying the features of the non-self in the worm database, which is the main function of the innate non-self selection.

After the BP network is trained for 2117 times of learning the samples of all known worms, the web system is tested to be infected by some love worms, happytime worms and the variants of the love worms. Through 260 experiments of immunizing the web system, the detection rate of these worms is 100%, the recognition rate

of love worms and happy-time worms is 100%, and the recognition rate of the unknown worms, i.e. the variants of the love worms, is 98%. When some totally unknown worms are tested, the recognition rate may be much lower than 98%, but the recognition rate can be increased through using much more known worms as the samples for learning.

4 Conclusions

Though detection and recognition of unknown worms are very difficult for traditional approaches in the web system, the normal model is proved to identify the normal states of the system and validated its effective detection through repeated experiments. Also, the BP neural network is designed to learn unknown worms and validated its effective recognition through repeated experiments. Therefore, the normal model and the neural learning are both important and useful for immunizing the web system.

Acknowledgements

Thanks for support of National Natural Science Foundation of China (60404021 & 60234030) and the Excellent Doctoral Degree Project of Central South University (040125). Moreover, we sincerely thank reviewers for their good advice.

References

1. Paker, Y., Kindberg, T.: The Worm Program Model: an Application Centred Point of View for Distributed Architecture Design. In: Proceedings of the 3rd Workshop on ACM SIGOPS European Workshop, ACM Press, New York (1988) 1-4
2. Staniford, S., Paxson, V., Weaver, N.: How to Own the Internet in Your Spare Time. In: Boneh, D. (ed.): Proceedings of the 11th USENIX Security Symposium, USENIX, Berkeley (2002) 149-167
3. Balthrop, J., Forrest, S., Newman, M.E.J., Williamson, M. M.: Technological Networks and the Spread of Computer Viruses. Science 304(5670) (2004) 527-529
4. Levy, E.: Worm Propagation and Generic Attacks. IEEE Security and Privacy 3(2) (2005) 63-65
5. Gray, R.S., Berk, V.H.: Rapid Detection of Worms Using ICMP-T3 Analysis. In: Carapezza, E.M. (ed.): Proceedings of SPIE, Vol. 5403. SPIE Press, Bellingham (2004) 89-101
6. Zou, C.C., Gong, W., Towsley, D.: Code Red Worm Propagation Modeling and Analysis. In: Atluri, V. (ed.): Proceedings of the 9th ACM Conference on Computer and Communications Security, ACM Press, New York (2002) 138-147
7. Einstein, A.: Relativity: the Special and General Theory. Three Rivers Press, New York (1920)
8. Gong, T.: An Immune Agent for Web-based AI Course. International Journal on E-Learning 5(4) (2006) to appear

Data Hiding in Neural Network Prediction Errors

Guangjie Liu[1], Jinwei Wang[1], Shiguo Lian[2], Yuewei Dai[1], and Zhiquan Wang[1]

[1] Department of Automation, Nanjing University of Science & Technology, 210094 Nanjing, China
guangj_liu@yahoo.com.cn
[2] Beijing R&D Center, France Telecom, 100000 Beijing, China

Abstract. In this paper, a new and efficient image hiding scheme is proposed. Different from the existing methods, the secret data is embedded into the prediction errors produced by the neural network nonlinear predictor, and the non-uniform quantization method is used to embed secret data. The proposed method can achieve higher embedding payload while keeping smaller distortion, and experimental results are given to show the advantage of this scheme.

1 Introduction

Steganography is used to send secret message under the cover of a host signal. It is generally accepted that the steganography must possess two important properties: imperceptibility and high hiding capacity. In literatures, many techniques for data hiding have been proposed[1,2,3], but they can not balance the relationship of capacity and imperceptibility well.

2 Neural Network Predictor

In the image compression researches , the predictive coding is often used to eliminate the correlation redundancy and make lossless of lossy compression. However, it is imprecise to use the linear model and some simple nonlinear predictors such as MEF, GAP and ALCM [4] to capture the statistics of the neighboring pixels. Neural networks offer effective tool for nonlinear prediction. In [5], Dianat's experiments have shown that the neural network predictor can significantly improve the efficiency of data compression in the MSE sense.

The design of a nonlinear neural network predictor is to find the optimal parameter set W for a nonlinear function of the previous p input, as shown by

$$\hat{x}(n) = f_{nn}(x(n-1), ..., x(n-p), W) \tag{1}$$

such that the mean squared value of the prediction errors $E[(\hat{x} - x)^2]$ is minimized. In this paper, the multilayer perception (MLP)network is used to design

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 273–278, 2006.

our predictor. Each unit computes the weighted sum of the inputs, pluses the bias weight and passes this sum through activation function to calculated the output value as

$$y_j = f_{nn}(\sum_j w_{ji} x_i + \theta_j) \tag{2}$$

where x_i is the ith input value for the neuron and w_{ji} is the corresponding weight. The activation function maps the potentially infinite range of the weight sum to a limited, finite range. A common activation function is Sigmod defined by the logistic function.

The architecture of the neural network for prediction in this paper is shown in Fig. 1, with one hidden layer containing six perceptrons. The activation function of inputlayer and output layer is linear function, and the hidden layer activation function is Sigmod. To compute the prediction out at each pixel position, the values of four previous pixel samples are presented as input to the neural network. These four pixels are the left, left-up, up and right-up of the current pixel respectively, as Fig. 1(a)shows. The weights of the network can be computed by training using the BP algorithm [6], which is a supervised learning algorithm.

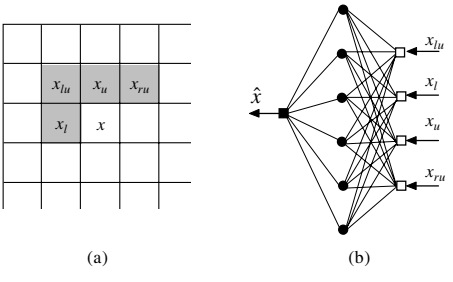


Fig. 1. Multilayer perceptron network for prediction coding (a) Notation used for specifying neighboring pixels of the current pixel x; (b) A three-layer perceptron network for predicting coding

Before the data hiding, training must be performed to find proper network parameters to minimize the prediction error energy. And, the parameters of the natural network predictor used in embedding and extracting must be kept identical. It will cause too much transmission burden on the side channel. A possible and feasible choice is to use a fixed prediction network for the same type of images. Here, we divide images into eight types according to the mean square adjacent pixels differences D, which can reflect the smoothness of images. The eight types is defined with D on [0,200),[200,400),[400,600),[600,800),[800,1000),[1000,1500), [1500,2000) and [2000,10000) respectively. Based on 400 gray images with size 512×512, for images of each type, the corresponding typical prediction network is trained,and eight typical predictive neural networks are obtained. By using these typical predictive neural network, the larger payload demands of side channel is avoided, just the index of typical neural network needs to be transmitted.

3 Proposed Hiding Method

In our scheme, the four edges of the cover image are not used to carry the secret data of the neighbor structure as Fig.1(a) shows. While these pixels can be used as the side channel to carry the index of typical prediction network discussed in Section 2. Before embedding ,for a given cover image, the corresponding prediction network is chosen according to D. Then the network index v is embedded into the pixels of four edges pixels by the simple LSB method.

The data to be hidden in the image is considered as a long bit stream. The hiding process is depicted as Fig. 2. For the current input pixel value $x(n)$, the prediction error e is obtained through subtracting $P(\hat{x}(n))$, the quantized predictive value $\hat{x}(n)$ computed by the prediction network v from the current cover image pixel value $x(n)$, then m bits of the data stream are selected and embedded into e by the Eq.(3).

$$\begin{aligned} \hat{x}(n) &= f_v(x'_l(n), x'_l u(n), x'_u(n), x'_r u(n)) \\ e &= x(n) - P(\hat{x}(n)) \\ e' &= G_K(e, s_m) \\ x'(n) &= e' + P(\hat{x}(n)) \end{aligned} \tag{3}$$

Here, $P(\cdot)$ represents quantizing to the nearest integer from the output of neural network predictor, $G_K(\cdot,\cdot)$ is the embedding function under the control of the stego key K, s_m stands for m bits of data stream, and $x'(n)$ is the stego pixel value. The four pixel value $x'_l(n)$, $x'_l u(n)$, $x'_u(n)$, $x'_r u(n)$ are the neighboring pixels in the stego image that correspond to the position of the current cover image pixel $x(n)$.

Inspired by the work of Wu in [4], a quantization-based information hiding method is proposed. First, the prediction error e is quantized to a quantization point q_i according to the equation as follows.

$$Q(e) = \arg \min_{q_i, q_i - |e| \le 0} |q_i - |e|| \tag{4}$$

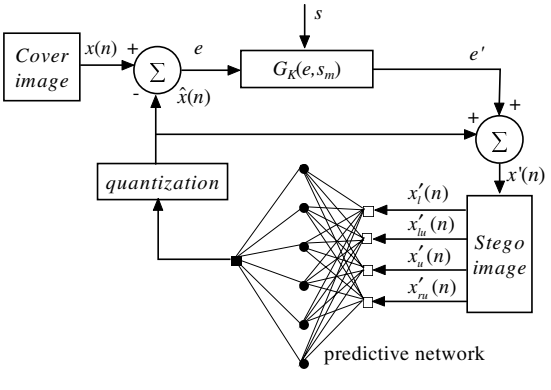


Fig. 2. Data Hiding Based on Neural Network Predictor

Then the amount of bits to be embedded into e is chosen equal to $\log_2 \Delta q_i$. The quantization point q_i and the quantization cell size Δq_i must obey Eq. (5). It means a partition of the prediction error absolute value range $[0, 255]$.

$$\begin{aligned} \Delta q_i = 2^{k_i}, \sum_{i=1}^{N} 2^{k_i} = 256 \\ q_1 = 0, q_i = \sum_{j=1}^{i-1} 2^{k_j}, i > 1 \end{aligned} \tag{5}$$

Here, $k_i (i = 1, \ldots, N)$ are all integers. To afford more data payload and make less image quality distortion, the partition should obey the sensitivity of HVS. It means that for larger error, the corresponding cell size is also larger, which implies that the larger the error value is, the more amounts of bits can be embedded. In this paper, we choose the integers $k_i (i = 1, \ldots, N)$ equal to $[2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 6]$. there may be also other feasible partition manners, when considering the demanded combination of the image quality and capacity.

After the quantization, $m = log_2 \Delta q_i = k_i$ bits and the stego-key controlled dither value is added onto the quantized value $Q(e)$, the sum is multiplied by the sign of the prediction error e to generate the modified prediction error e', the whole process can be described by Eq.(6).

$$e' = G_k(e, s) = sign(e) \cdot (Q(e) + mod(s_{log_2 \Delta q_i}(n) + \lfloor \beta \Delta q_i \rfloor, \Delta q_i)) \tag{6}$$

where, $\beta \in [0, 1]$ is generated from the stego-key K, and Δq_i is the cell size of the quantization point q_i, and $\lfloor \cdot \rfloor$ represents the operation of truncation to the integer part. Here, the dither value $\lfloor \beta \Delta q_i \rfloor$ is used to enhance the security of our method. Furthermore, it is more important that the computation of Eq. (6) is able to guarantee $Q(e') = Q(e)$, which makes the blind data extracting possible.

The mistakes will come forth during extracting, when the sum of the quantized prediction pixel value $\hat{(}x)(n)$ and the modified prediction error e' exceeds the range $[0, 255]$. To avoid data overflow, the condition (7) should be examined to decide whether the data is able to be hidden in the current error.

$$x' = P(\hat{x}(n)) + sign(e) \cdot (Q(e) + \Delta q_i) \in [0, 255] \tag{7}$$

If the condition is not satisfied, the current pixel will be skipped, and the embedding process will move to the next pixel.

Before data extracting, the decoder must decodes the prediction network index v firstly. Then the predictor network v is used to predict the current pixel $x(n)$ to get $\hat{x}(n)$. Accordingly, the prediction error e is obtained. Like the examination in the embedding process, Eq. (7) is examined to decide whether the pixel contains secret data. If not, goes next, else uses the same parameter β generated by the identical stego-key to extract the secret data by the equation (8).

$$s_m(n) = mod(e' - Q(e') - \beta \Delta q_i + \Delta q_i, \Delta q_i) \tag{8}$$

4 Experiments and Discussions

To evaluate the performance of the proposed scheme, the experiment is made, which is based on four typical test images, "Lena", "Baboon", "Jet" and "Peppers", each with size 512×512. The secret data used in the experiments are random bits, which is generated by a pseudo-random number generator(PRNG). The measurement of the image quality in the experiment is PSNR. Table 1 gives the capacity and image quality of the proposed scheme.

Table 1. Capacities and PSNRs of four stego images

Cover image	Capacity(bits)	PSNR(dB)
Lena	554,564	43.17
Baboon	671,423	38.64
Jet	558,565	43.00
Peppers	574,736	42.10

From Table 1, we can see that the images with more edges and textures can carry more information than the flat ones. And the PSNRs of those four images are all kept above 38dB.

In Fig. 3(a)and (c), the two of the stego images ,Lena and Baboon, are shown. And in Fig. 3(b)and (d), we show the corresponding enhanced images between the stego images and the cover images. It is clearly exhibited that most information is embedded into the edges and textures of the cover images. These

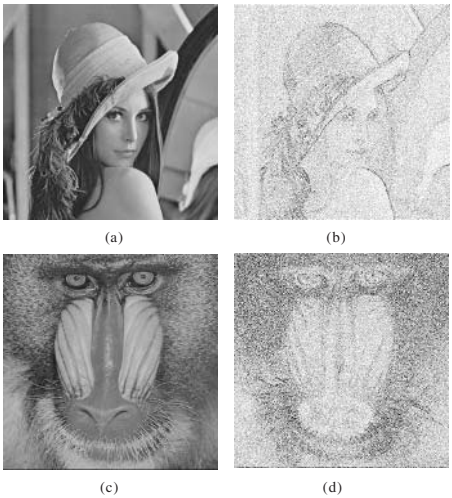


Fig. 3. Two of stego images and the corresponding enhanced difference images after embedding. (a) The stego image of Lena. (b) The enhanced difference image of Lena. (c) The stego image of Baboon. (d) The enhanced difference image of Baboon.

distortion are less noticeable because the changes in edge and texture parts of image are generally less obvious to human eyes.

5 Conclusions

In this paper, we use the prediction errors produced by the neural network predictor to hide the secret data. Moreover, the non-uniform quantization embedding method is used to achieve the good combination between image quality and hiding capacity. The experimental results reveal the practicability and superiority of the presented scheme.

Acknowledgement

This work was supported by he National Natural Science Foundation of China through the grant number 60374066.

References

1. Lie, W.N., Chang L.C.: Data Hiding in Images with Adaptive Numbers of Least Significant Bits Based on the Human Visual System. Proc. Int. Conf. Image Processing. IEEE Computer Society, Kobe Japan (1999) 286-290
2. Wu, D.C., Tsai, W.H.: A Steganographic Method for Images by Pixel-Value Differencing. Pattern Recognition Letters 24(9-10) (2003) 1613-1626
3. Yu, Y.H., Chang, C.C.: Hu, Y.C., Hiding Secret Data in Images via Predictive Coding. Pattern Recognition 38(5) (2005) 691-705
4. Memon, N., Wu, X.L.: Recent Developments in Context-based Predictive Techniques for Lossless Image Compression. Computer Journal 40(2-3) (1997) 127-136
5. Dianat, S.A., Nasrabadi, N.M., Venkataraman, S.: A Non-linear Predictor for Differential Pulse-code Encoder(DPCM) Using Artificial Neural Network. Proc. Int. Conf. Acoustics, Speech, and Signal Processing. IEEE Computer Society, Toronto Canada (1991) 2793-2796
6. Hagan, M.T., Demuth, H.B., Beale, M.H.: Neural Network Design. PWS Publishing Company, Boston (1996)

The Minimum Detectable Capacity of Digital Image Information Hiding

Fan Zhang[1], Ruixin Liu[2], and Xinhong Zhang[3]

[1] College of Computer & Information Engineering, Henan University, Kaifeng 475001, P.R. China
zhangfan@vip.sohu.com
[2] Yellow River Conservancy Technical Institute, Kaifeng 475001, P.R. China
kflrx@1163.com
[3] Department of Computer Center, Henan University, Kaifeng 475001, P.R. China
hnkfzxh@163.com

Abstract. Information hiding capacity of digital image is the maximum information that can be hidden in an image. But the lower limit of information hiding, the minimum detectable information capacity is also an interesting problem. This paper proposes a method to analyze the minimum detectable capacity of information hiding in digital images based on the theories of attractors and attraction basin of neural network. The results of research show that the attractors of neural network decide the lower limit of information hiding.

1 Introduction

Information hiding capacity of digital image is an evaluation of how much information can be hidden within images. Information hiding can be considered as a communication process. The image is the communication channel in which the information is hidden. Information hiding capacity corresponds to the communication capacity of the "information hiding channel" [1],[2]. Consider the original image as an independent additive white Gaussian noise (AWGN) channel, the information hiding capacity can be calculated according to the well-known Shannon capacity formula.

Recently, some works on the information hiding and watermarking capacity have been presented. Servetto considers each pixel as an independence channel and calculates the capacity based on the theory of Parallel Gaussian Channels (PGC) [3]. Barni's research focuses on the information hiding capacity of digital image in the DCT and the DFT domain [4]. Moulin's work introduces a game-theoretical approach for the evaluation of information hiding capacity problem under attacking [5],[6]. Lin presents a capacity analysis of zero-error information hiding in the JPEG compressed domain using adjacency-reducing mapping technique [7]. Zhang presents an adaptive watermarking capacity analysis in the spatial domain and the wavelet transform domain [8],[9].

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 279–284, 2006.

Almost all the previous works on information hiding capacity used the information theoretic model, and those researches focus on the maximum information that can be hidden in an image, which is the upper limit of hidden information. But the lower limit of information hiding, the minimum detectable information capacity is also an interesting problem. It is difficult if we analyze the minimum detectable information capacity according to the information theoretic model, so few works are proposed in this field. This paper proposes new information hiding limits analysis method that is based on the theories of attractors and attraction basin of neural network.

The rest of this paper is organized as follows. An information-hiding algorithm is proposed in Section 2. In Section 3, we discuss the lower limit of information hiding, namely the minimum detectable information capacity. Finally, the conclusions of this paper are drawn in Section 4.

2 An Information-Hiding Algorithm Based on Neural Network

In the information-hiding algorithm, a Hopfield neural network is used to store the original image during the information embedding process. And the same neural network is used to retrieve the image during the information extraction process.

Hidden information: The information that is hidden in images is a binary sequence, which is generated by a pseudo-random sequence generator. The length of an information sequence is less than or equals to the number of pixels.

Neural network learning: A discrete Hopfield network is trained with the original image as the input pattern. The number of neurons equals to the number of pixels. The neural network learns and stores the original image. The hidden information sequence can also be stored at the same time if the original hidden information is necessary in the information detection.

Information hiding embedding: The information is embedded into an image according to the bits of the hidden information sequence. Each bit of the hidden information sequence affects the amplitude of one pixel. If the bit is 0, there is no change to the amplitude of corresponding pixel. If the bit is 1, the amplitude of the pixel is modified. The magnitude of modification for each pixel may vary and is determined by a perceptual model in order to the information invisible. In this algorithm, the magnitude of modification is simply added to the amplitude of corresponding pixels.

Information hiding extraction: Firstly, the neural network retrieves the original image from the stego images (stego image is a terminology which denotes the information hidden image) or the noised images. Then, the hidden information data are extracted by subtracting the retrieved image from the stego image. Finally, the hidden information sequence is reconstructed according to a threshold.

A block diagrams is shown in Figure 1 to illustrate the information hiding embedding, and a block diagrams is shown in Figure 2 to illustrate the information hiding extraction.

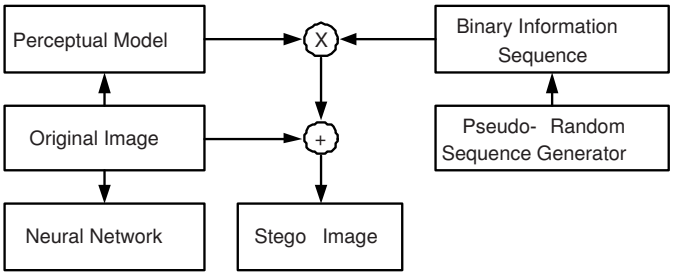


Fig. 1. Illustration of the information embedding

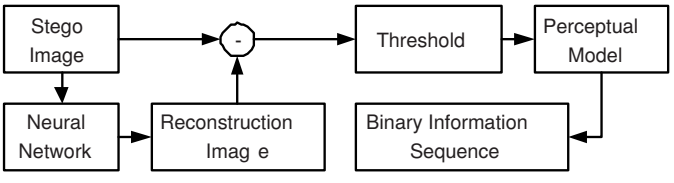


Fig. 2. Illustration of the information extraction

The stego Baboon image and the reconstruction image are shown in Figure 3. The peak signal to noise ratio (PNSR) is 31.87 dB.

3 The Minimum Detectable Information Capacity

The Hopfield network is a recurrent neural network that stores information in a dynamically stable configuration. An energy function is used to evaluate the stability property, and the energy function always decreases to a state of the lowest energy. We can concentrate the Hopfield model as an associative memory that is a storage device that is able to obtain stored information from some input data [10],[11]. Attractors in an energy surface represent the stored patterns. The basin of attraction is the set of states in the system within which almost all states flow to one attractor. For a trained neural network, the attraction basin gives a measure of neural network error-correcting capability. Once a pattern is stored, the Hopfield network can reconstructs the original pattern from the degraded or incomplete pattern.

The concept of capacity in the neural network is different to the capacity concept of information theory. In the Hopfield neural network, a pattern is called

(a) (b)

Fig. 3. The stego Baboon image (PNSR is 31.87 dB)(a) and the reconstruction image (b)

stored if it is a fixed point of the retrieval dynamics and the storage capacity is understood as the asymptotic of the number of patterns such that all of them are fixed points. The storage capacity in this concept is defined as the number of patterns that can be permitted such that they (one or all of them) are fixed points of the retrieval dynamics [12],[13]. The other approach to storage capacity takes into account small errors we are willing to accept in the restoration of the patterns (with the idea to increase the storage capacity) [14],[15]. So we are satisfied if the retrieval dynamics converges to a configuration, which is not too far away from the original patterns.

Assume that the set of P stored patterns is given by $\xi = (\xi^1, \xi^2, \cdots \xi^P)$, the nodes in the network are labeled 1, 2, ..., N. If an input pattern ζ is same as one of the stored patterns ξ^υ, then the stability condition of Hopfield model is:

$$sgn(\zeta_i^\upsilon) = \xi_i^\upsilon, \forall i, \tag{1}$$

where

$$\begin{aligned} \zeta_i^\upsilon &= \sum_{j=1}^{N} w_{ij}\xi_j^\upsilon = \frac{1}{N}\sum_{j=1}^{N}\sum_{\mu=1}^{P}\xi_i^\mu\xi_j^\mu\xi_j^\upsilon \\ &= \xi_i^\upsilon + \frac{1}{N}\sum_{j=1}^{N}\sum_{\mu=1,\mu\neq\upsilon}^{P}\xi_i^\mu\xi_j^\mu\xi_j^\upsilon. \end{aligned} \tag{2}$$

The second term is called the crosstalk term. If it is zero, the network has stability. But even if it is not zero, the network can still has stability if its magnitude is smaller than 1, in which case it cannot changes the sign of ζ_i^υ. It turns out that if this is the case, and the initial state of the network is near one of the stored patters (in Hamming distance), the network moves towards ξ_i^υ. ξ_i^υ

is an attractor. The more patterns that are stored, the lower the chances that the crosstalk term is sufficiently small.

Let

$$C_i^v = -\frac{1}{N}\sum_{j=1}^{N}\sum_{\mu=1,\mu\neq v}^{P}\xi_i^\mu\xi_j^\mu\xi_j^v. \tag{3}$$

If $C_i^v < 0$, the crosstalk term has the same sign as the desired ξ_i^v. Thus, ζ_i^v always has the same sign as ξ_i^v. We can define the error probability that some bit in the pattern will not be stable (because in this case the crosstalk term changes the sign and the bit is flipped) as follows.

$$P_e = P(C_j^\mu > 1). \tag{4}$$

The quantity C_i^v has a binomial distribution since they are the sum of random numbers (-1, +1). For P random patterns and N units, the distribution of values for the quantity C_i^v close to Gaussian distribution with variance $\sigma^2 = P/N$. Thus,

$$P_e = \frac{1}{\sqrt{2\pi}}\int_1^\infty \exp\left(-\frac{x^2}{2\sigma^2}\right)dx$$

$$= \frac{1}{2}\left[1 - erf(\sqrt{\frac{N}{2(P-1)}})\right]. \tag{5}$$

Where $erf(x)$ is the Complementary Error Function that is defined as,

$$erf(x) = \frac{2}{\sqrt{\pi}}\int_0^x \exp(-u^2)du. \tag{6}$$

If the error is lesser than 0.01, then the maximum number of patterns that can be stored is $0.185N$. Now we consider this problem in the counter way, if we want to the neural network store some patterns, the minimum number of nodes $N_{\min} = P/0.185$. In proposed information-hiding algorithm, the minimum number of nodes are 6, namely at least 6 neurons are needed. each neurons corresponds with a point that the information data is hidden, so at least 6 pixels will be modified. In proposed information-hiding algorithm, each of the embedded points corresponds with one bit information messages, so, when $P_e <$ 0.01, the minimum detectable information messages is 6 bits.

4 Conclusions

This paper proposes new information hiding limits analysis method that is based on the theories of attractors and attraction basin of neural network. The proposed method abandons the information-theoretic model, the results of research show that the attractors of neural network decide the lower limit of information hiding.

References

1. Cox, I., Millller, M., McKellips, A.: Watermarking as Communications with Side Information. Proceedings of the IEEE, Special Issue on Identification and Protection of Multimedia Information 87(7) (1999) 1127–1141
2. Cox, I., Kilian, J., Leighton, F.: Secure Spread Spectrum Watermarking for Multimedia. IEEE Transactions on Image Processing 6(12) (1997) 1673–1687
3. Servetto, S., Podilchuk, C., Ramchandran, K.: Capacity Issues in Digital Image Watermarking. In: IEEE International Conference on Image Processing, Chicago, Illinois, USA (1998) 445–449
4. Barni, M., Bartolini, F., Rosa, D.: Capacity of Full Frame DCT Image Watermarks. IEEE Transactions on Image Processing 9(8) (2000) 1450–1455
5. Moulin, P., Mihcak, M.: A Framework for Evaluating the Data-Hiding Capacity of Image Sources. IEEE Transactions on Image Processing 11(6) (2002) 1029–1042
6. Moulin, P.: The Role of Information Theory in Watermarking and Its Application to Image Watermarking. Signal Processing 81(6) (2001) 1121–1139
7. Lin, C., Chang, S.: Zero-error Information Hiding Capacity of Digital Images. In: IEEE International Conference on Image Processing. Thessaloniki, Greece (2001) 1007–1010
8. Zhang, F., Zhang, H.: Digital Watermarking Capacity and Reliability. In: Proceedings of e-Commerce Technology. San Diego, California, USA (2004) 295–298
9. Zhang, F., Zhang, H.: Image Watermarking Capacity and Reliability Analysis in Wavelet Domain. Journal of Imaging Science and Technology 49(5) (2005) 481–485
10. McEliece, R., Posner, C., Rodemich, R., Santosh, R.: The Capacity of the Hopfield Associative Memory. IEEE Transactions on Information Theory 33(4) (1987) 461–482
11. Davey, N., Hunt, S.: The Capacity and Attractor Basins of Associative Memory Models. In: 5th International Conference on Artificial and Natural Networks. Lecture Notes in Computer Science, 1606. Springer-Verlag, Berlin Heidelberg New York (1999) 330–339
12. Hopfield, J.: Neural Networks and Physical Systems with Emergent Collective Computational Abilities. Proc. Natl. Acad. Sci. USA 79(4) (1982) 2554–2558
13. Hertz, J., Krogh, A., Palmer, R.: Introduction to the Theory of Neural Computation. Addison-Wesley, USA (1991)
14. Amit, D., Gutfreund, G., Sompolinsky, H.: Statistical Mechanics of Neural Networks Near Saturation. Annals of Physics 173(1) (1987) 30–67
15. Newman, C.: Memory Capacity in Neural Network Models: Rigorous Lower Bounds. Neural Networks 3(2) (1988) 223–239

Robust Digital Image Watermarking Algorithm Using BPN Neural Networks

Cheng-Ri Piao, Wei-zhong Fan, Dong-Min Woo, and Seung-Soo Han

Department of Information Engineering & NPT Center, Myongji University,
Yongin, Kyunggi, 449-728, South Korea
shan@mju.ac.kr

Abstract. This paper proposes a new watermarking scheme in which a logo watermark is embedded into the spatial domain of the original image using Back-Propagation neural networks (BPN). BPN will learn the characteristic of the image, and then watermark is embedded and extracted by the trained BPN. The image is divided into 8×8 blocks and the average pixel value of each block is used as the desired output value of the BPN. The quantized DC coefficient of discrete cosine transform (DCT) domain of each block is used as input value of the BPN to be trained. After the BPN is trained using those input/output values, watermark is embedded into the spatial domain using the trained BPN. The trained BPN also used in watermark extracting process. Experimental results show that the proposed method has good imperceptibility and high robustness to common image processing.

1 Introduction

With the rapid development of computer and communication networks, the digital multimedia reproduction and distribution are becoming extremely easier and faster. However, these advances also afford unprecedented opportunities to pirate copyrighted digital multimedia products. As a result, the watermarking technique, which embeds a watermark into digital multimedia contents for detecting and tracing copyright violations, has recently become a very active area of multimedia security [1]. The watermarking techniques can be classified into two classes depending on the domain of watermark embedding, i.e. a spatial domain and a frequency domain. Among the spatial domain watermark embedding methods, Schyndel *et al*. proposed a watermark embedding technique by changing the least significant bit of some pixels in an image [2]. Bender *et al*. described a watermarking approach by modifying a statistical property of an image called 'patchwork' [3]. On the other hand, there are many algorithms for watermark embedding in frequency domain. Cox *et al*. described a method where the watermark is embedded into the large discrete cosine transform (DCT) coefficients using ideas borrowed from spread spectrum in communications [4]. Xia *et al*. proposed a frequency domain method of embedding the watermark at all the subbands except LL subband, using discrete wavelet transform (DWT)[5].

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 285 – 292, 2006.

Recently, quantization index modulation (QM) [6-8] technique is widely used in watermarking area, and this technique is very robust to various attacks such as JPEG compression, noise insertion, image resize and so on. And Mei, *et. al*. [9] proposed a watermarking technique using neural network. It is a very robust method, but a non-blind method that requires original image to extract the watermark.

In this paper, a new watermark embedding/extracting algorithm using error back-propagation neural network (BPN) is introduced. In this algorithm, an image is divided into 8×8 blocks, then calculate the average pixel value for each block, make it the desired output for BPN, and start the BPN learning procedure. After that, embed and extract the watermark by the trained BPN.

The experimental results show that the watermarked image has at least 43 dB in peak signal-to-noise ratio (PSNR). Also, the results are compared among the presented scheme, the method that uses quantization index modulation (QM) technique on the DC coefficient of DCT domain, and the neural network method applied on DCT domain [9].

2 Related Theories

2.1 The Error Back-Propagation Neural Network (BPN)

The BPN is a kind of supervised learning neural network. It is one of the most frequently used learning techniques in neural networks. The principle behind the BPN involves using the steepest gradient descent method to reach a small approximation. A general model of the BPN has an architecture like that depicted in Fig. 1. There are three layers including input layer, hidden layer, and output layer. Two nodes of each adjacent layer are directly connected to one another, which is called a link. Each link has a weighted value, which represents the relational degree between two nodes. A training process described by the following equations updates these weighted values:

$$\begin{aligned} net_j(t) &= \sum_i \alpha_{i,j} o_i(t) - \theta_j \\ o_j(t+1) &= f_{act}(net_j(t)) \end{aligned} \tag{1}$$

where $net_j(t)$is the activation value of the j^{th} node in iteration t, $o_j(t+1)$is output of the j^{th} node in iteration $t+1$, $f_{act}(x)$ is called the activation function of a node, which usually is a sigmoid function in hidden layers and a pureline function in output layer. Generally, all initial weight values $\alpha_{i,j}$ are assigned using random values. In each iteration process, all $\alpha_{i,j}$ are modified using the delta rule according to the learning samples. The trained neural network can memorize the characteristics of the learning samples, and predict a new output due to the adaptive capability of it. BPN will be used to learn the characteristics of image for improving the performance of the proposed watermarking scheme in the section 3.

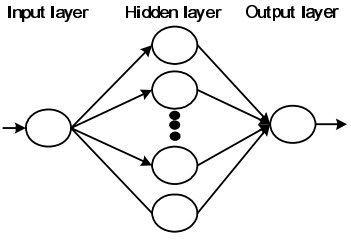


Fig. 1. Architecture of a BPN

2.2 The BPN Neural Network Training Procedures

The BPN neural network training procedures are as Fig. 2 below.

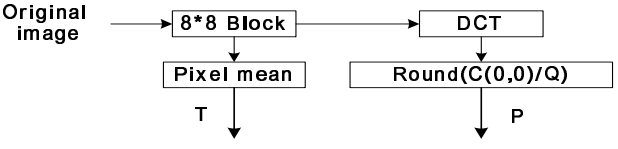


Fig. 2. BPN training procedures

In the figure, *C(0,0)* is the DC coefficient of the discrete cosine transform (DCT) domain of the original image, and *Q* is the quantization value. The output of the equation *Round(C(0,0)/Q)*, which is represented as *P*, is used as an input value for the BPN, and *T*, which is the average pixel value of each block, is used as a desired output value for the BPN neural network. The structure of BPN network presented in the paper is 1-120-1, which is obtained as an optimal neural network structure from many experiments. The hidden layer uses sigmoid function, and the output layer uses pureline function. The training method is Levenberg-Marquardt rule. And the training error is set to 0.01 and the number of maximum learning iteration is set to 5000.

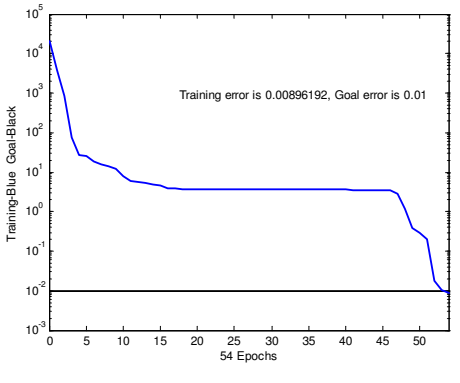


Fig. 3. The training process of BNP

The training is finished when either training error is smaller than 0.01 or the iteration is reached to the maximum iteration number. Fig. 3 shows the learning error in each step when the BPN is trained using a 256×256 size Lena image. The BPN trained by this way will be used to embed and extract the watermark.

3 Watermark Embedding and Extracting

Generally, a watermark algorithm includes 3 steps: watermark generation, embedding, and extraction. In this paper, a logo image, which can be easily distinguished by human eyes, is embedded as a watermark. The proposed watermark technique is a kind of blind watermarking algorithm, which embed/extract the watermark into/from the spatial domain.

3.1 Watermark Embedding

Fig. 4 below is the block diagram of the watermark embedding procedure. The trained BPN in the figure is discussed in section 2.2. Q is the quantization value.

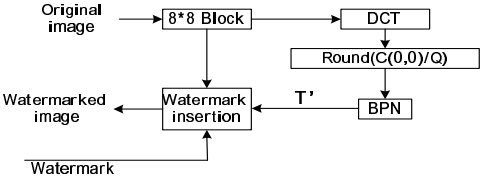


Fig. 4. Watermark embedding procedure

The watermark embedding procedures are as follows:

Step1: Divide the original image to be 8×8 blocks and make DCT transformation.
Step2: Quantize the DC coefficient $C(0,0)$ of the DCT by Q, and use that value as the input value of BPN then get the output T'.
Step3: Embed the watermark according to the equation 2 and 3 below which use the output value T' and the pixel values of the corresponding block.

$$\Delta P_{blcok\,1} = \frac{1}{4} * 8 * Q + 64 * T' - \sum_{i,j=0}^{7} p_{i,j}$$
$$\Delta P_{blcok\,0} = -\frac{1}{4} * 8 * Q + 64 * T' - \sum_{i,j=0}^{7} p_{i,j} \tag{2}$$

$$p'_{i,j} = \begin{cases} p_{i,j} + round\ (\Delta P_{blcok\ 1} / \sum_{i,j}^{7} p_{i,j}) & if \quad w_{m,n} = 1 \\ p_{i,j} + round\ (\Delta P_{blcok\ 0} / \sum_{i,j}^{7} p_{i,j}) & if \quad w_{m,n} = 0 \end{cases} \tag{3}$$

Where $p_{i,j}$ is the pixel value of the original image, $p'_{i,j}$ is the pixel value in which watermark is embedded. w is the watermark and Q is the quantization value. ΔP_{blok1} is the change of sum of all pixels in a block when watermark is 1, and ΔP_{blok0} is the change of sum of all pixels in a block when watermark is 0. $\sum_{i,j=0}^{7} p_{i,j}$ represents the sum of pixel value in a block, and $64\times T'$ is also the sum of pixel value of a block, but calculated by BPN. A change $\frac{1}{4}*8*Q$ should be given to each block because $Round(C(0,0)/Q)$ must have no change after embedding a watermark, so that there will be no change in $Round(C(0,0)/Q)$ when extracting the watermark. With many experiments, we concluded that the embedded watermark is most robust to attacks when the change is $\frac{1}{4}*8*Q$.

3.2 Watermark Extracting

The watermark extracting procedures are the converse procedures of watermark embedding, as shown in Fig.5. The BPN here is trained neural network, which is discussed in section 2.2, and Q is the quantization value.

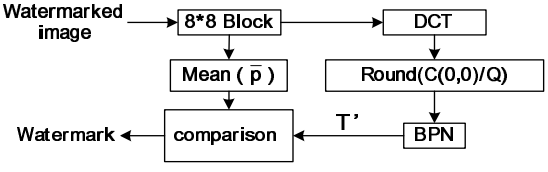


Fig. 5. Watermark extraction procedure

The watermark extraction procedures are as follows:

Step1: Divide the watermarked image into 8×8 blocks and calculate the average pixel value $\overline{p}$ for each block, and make a DCT transformation.

Step2: Quantize the DC coefficient $C(0,0)$ of the DCT by Q, and use it as input value of the BPN to get the output T'.

Step3: Extract the watermark using the equation 4 below, which is in use of output T' and $\overline{p}$. w' of equation 4 is extracted watermark.

$$w'_{m,n} = \begin{cases} 1 & if \quad \overline{p} > T' \\ 0 & else \end{cases} \tag{4}$$

Step 4. The correlation between the original watermark and the extracted watermark is calculated to detect the existence of the watermark. The similarity between the original watermark w and the extracted watermark w' is quantitatively measured by the bit correlation ratio (BCR), defined as follows:

$$BCR = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}\overline{(w_{i,j} \otimes w'_{i,j})}}{\sum_{i=1}^{m}\sum_{j=1}^{n}\overline{(w_{i,j} \otimes w_{i,j})}} \times 100\% \tag{5}$$

Where $w_{i,j}$ is the original watermark bit, $w'_{i,j}$ is the extracted watermark bit, and $\otimes$ is the exclusive OR.

4 Experimental Results

In this paper, 8 bit 256×256 sized Lena, cameraman, baboon, boat images are used as cover (original) images, and a 32×32 sized logo image, which can be easily distinguished by human eyes, is used as watermark. Fig.6 shows the images used in this experiment and the watermark in use.

Fig.7 shows the relationship between the embedding quantization step-size Q and the peak signal-to-noise ratio (PSNR). We can see that the PSNR of the watermarked image is decreasing with increasing Q value, but in any case the PSNR is more than 43dB. This shows that the watermarked image has a good PSNR.

We tested the robustness of the proposed method with several typical images attacked by JPEG compression. The watermarks extracted from JPEG compressed versions of the watermarked image with various compression quality factors, and

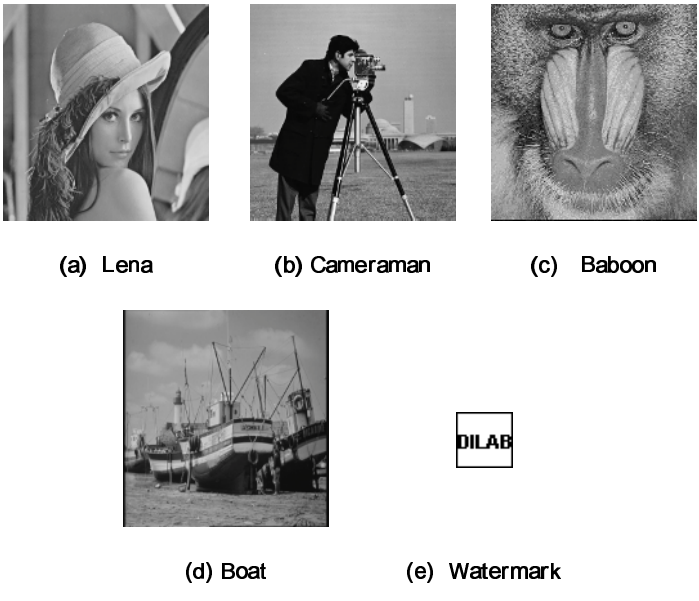


Fig. 6. Experiment images and watermark

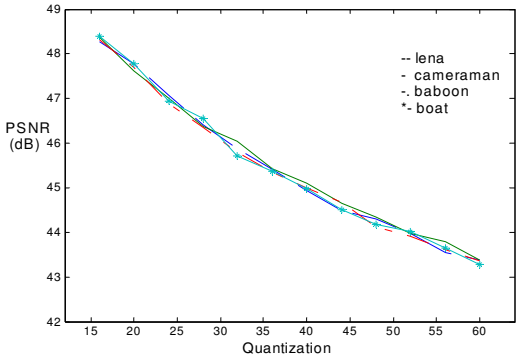


Fig. 7. The relationship between Q and PSNR

Table 1. BCR values of the extracted watermarks (%)

Images / PSNR / Quality Factor	Lena	Baboon	Cameraman	Boat
	44.48 [dB]	44.65 [dB]	44.64 [dB]	44.51 [dB]
100	100	100	100	100
90	100	100	100	100
80	99.99	100	99.70	100
70	100	99.80	99.21	99.99
60	99.95	99.60	98.53	100
50	99.68	98.53	97.36	99.80
40	96.68	97.06	95.87	96.09

Table 2. Comparison(BCR) results among QMs' method, nerual netwrok method described in [9] , and presented method

	QM method	Method [9]	The proposed method
PSNR [dB]	44.16	44.12	44.53
Embedded	100	100	100
JPEG (Quality factor = 40)	89.45	88.04	96.42
Noise (Gaussian)	87.50	85.86	89.04
Resize (225×225)	82.71	84.92	86.23

their corresponding BCR values are listed in Table 1. According to Table 1, we can see that the extracted watermark is still recognizable when the compression quality factor reaches 40, which means the proposed method has good robustness to JPEG compression.

Finally, we compared the performance of the proposed method with that of the one proposed in QM method, and with the performance of the scheme using neural network on DCT domain. All the three methods have the same test conditions

including the same test image "Lena" (256×256), the same amount of embedded information (1024 bits i.e. a 32×32 binary pattern watermark). Comparison results are listed in Table 2. According to this table, the BCR of the extracted watermarks using the proposed method are always higher than the others. These results prove that the proposed method has superior performance.

5 Conclusions

A new watermarking method for image has been proposed. It has following characteristics: First, a logo watermark is embedded into the spatial domain of the image using Back-Propagation neural networks (BPN). The embedding scheme can result in good quality of the watermarked image. Second, a BPN model is used to learn the characteristics of image. Due to the learning and adaptive capabilities of the BPN, the embedding /extracting strategy can greatly improve the robustness to several attacks. Experimental results illustrate that the performance of the proposed technique is superior to that of the similar one in the literature.

Acknowledgement

This work was supported by the ERC program of MOST/KOSEF (Next-generation Power Technology Center).

References

1. Hartung F., Kutter, M.: Mulimedia Watermarking Techniques. Proceedings of the IEEE (1999) 1079-1094
2. Schyndel, R.G., Tirkel, A.Z., Osborne, C.F.: A Digital Watermarking. Int. Conf. on Image Processing, **2** (1994) 86-90
3. Bendor, W., Gruhl, D., Morimoto, N., Lu, A.: Techniques for Data Hiding. IBM Systems Journal **3&4** (1996) 313-336
4. Cox, I.J., Kilian, J., Leighton, T., Shamoon, T.: Secure Spread Spectrum Watermarking for Multimedia. IEEE Trans. on Image Processing **12** (1997) 1673-1687
5. Xia, X., Boncelet, C.G., Arce, G. R.: A Multiresolution Watermark for Digital Images. Proc. IEEE ICIP **3** (1997) 548-551
6. Chen, B., Wornell, G.W.: Quantization Index Modulation: A Class of Provably Good Methods for Digital Watermarking and Information Embedding. Information Theory, IEEE Transactions on, **4**, (2001)1423 - 1443
7. Jang, Y., Kim, I., Kang, H., Kim, K., Han, S.: Blind Watermarking Algorithm Using Complex Block Selection Method. Lecture Notes in computer science, Vol. **2195.** Springer (2001) 996-1001
8. Inoue, H., Miyazaki, A., Araki, T., Katsura, T.: A Digital Watermark Method Using the Wavelet Transform for Video Data. IEICE Trans. Fundamentals. **E83-A** (2000) 90-95
9. Mei, S., Li, R., Dang, H., Wang, Y.: Decision of Image Watermarking Strength Based on Artificial Neural Networks. Neural Information Processing, ICONIP '02. Proceedings of the 9th International Conference on, Vol. 5. (2002) 2430 - 2434

A Novel Watermarking Method with Image Signature

Xiao-Li Niu[1,2], Ju Liu[1,2], Jian-De Sun[1], and Jian-Ping Qiao[1]

[1] School of Information Science and Engineering, Shandong University, Jinan, Shandong 250100, China
[2] National Laboratory on Machine Perception, Peking University, Beijing 100871, China
xiaoliniu@mail.sdu.edu.cn,
{juliu, jd_sun, jpqiao}@sdu.edu.cn

Abstract. In this paper, we propose a novel invisible watermarking method with image signature, which can resist not only common image attacks but also copy attack. The image signature algorithm referred is based on Image Independent Block Feature (IIBF) obtained by Independent Component Analysis (ICA). Simulations demonstrate the robustness of this method. In addition, this invisible watermarking method can be applied to protect the visible-watermarked image. That is to say, we can claim the ownership of the image after visible watermark removal by the invisible watermark.

1 Introduction

With rapid spread of computer network and wide use of multimedia technology, many watermarking techniques are now under investigation for protecting owner's intellectual property rights. In general, digital watermarks can be classified into visible and invisible watermarks.

Invisible watermarks are expected to be invisible in perception and robust to malicious attacks as well as common image processing. In some situations, a watermark of a legal marked image may be maliciously embedded in another illegal non-marked image. This is called copy attack which can create the false positive problem and cause ownership ambiguity [1].

In this paper, a novel invisible watermarking method with image signature is proposed, which can resist copy attack. Meanwhile, a new image signature algorithm based on Independent Block Feature (IIBF) has been studied. Simulations demonstrate that the watermarking method is not only robust to common image processing but also to copy attack and the image signature algorithm is better for identifying different images than the current algorithm [2]. Furthermore, this method is considered to protect the visible- watermarked image. We can exactly claim the ownership of the image after visible watermark removal by the invisible watermark.

2 Independent Component Analysis

Independent Component Analysis (ICA) is probably the most powerful and widely-used method for performing Blind Source Separations (BSS) [3]. The simplest

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 293 – 298, 2006.

ICA model assumes the existence of n independent components $s_1, s_2, \cdots, s_n$ and the same number of linear and instantaneous mixtures of these sources $x_1, x_2, \cdots, x_n$.

In vector-matrix notation, the mixing model can be represented as

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s} \tag{1}$$

where $\boldsymbol{s} = [s_1, s_2, \cdots, s_n]^T$, $\boldsymbol{x} = [x_1, x_2, \cdots, x_n]^T$ and $\boldsymbol{A}$ is a $n \times n$ mixing matrix. The de-mixing process can be formulated as computing the separating matrix $\boldsymbol{B}$, which is the inverse of mixing matrix $\boldsymbol{A}$. Finally, the independent components are obtained by

$$\boldsymbol{s} = \boldsymbol{B}\boldsymbol{x} \tag{2}$$

In this method, to fit the ICA model, an image is divided into m blocks which are regarded as observation signals $x_1, x_2, \cdots, x_n$. The Image Independent Block Features (IIBFs) $s_1, s_2, \cdots, s_n$ can be obtained by (2) [4].

3 Image Signature

John Barr pointed out that image signature is helpful to cope with copy attack and the low frequency DCT values of an image could be used to calculate its signature [2]. In this paper, we propose a new image signature algorithm by ICA.

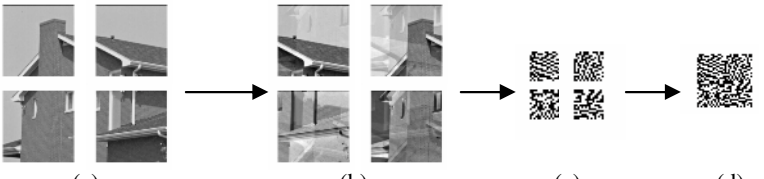


Fig. 1. (a)Image blocks; (b) Image Independent Block Features (IIBFs); (c) Image block signatures; (d) Image signature

The process of image signature calculation is shown in Fig.1. Firstly, the image is divided into some image blocks. And then, the IIBFs are derived by ICA. We perform DCT on the IIBFs, and pick up the lowest $m_1 \times m_2$ DCT coefficients following zigzag path in each block. The median of these DCT coefficients except the DC value is set as a threshold, by which the $m_1 \times m_2$ coefficients can be converted to a binary sequence as the block signature. Finally, the image signature can be formed by combining all block signatures.

We selected 432 images from the Eigenfaces [5] to test the effectiveness of our algorithm. Hamming distances between different image signatures are used to measure their differences. Image signatures of size 16×16 are calculated. So the expected average Hamming distance is 128 according to the knowledge of stochastic process and probability theory. In this experiment, the tested average Hamming distance using our

algorithm was 126.2491, and the value was 121.3864 while using the algorithm in [2]. The tested average Hamming distance of our algorithm is closer to the expected value than of John Barr's. It means that the image signature calculated by our algorithm is better for identifying different images. Fig.2 shows the experimental result.

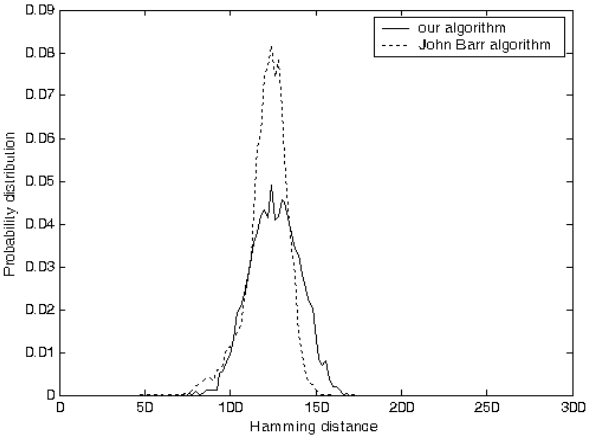


Fig. 2. Hamming distance distributions between different image signature algorithms

4 Watermarking Method

4.1 The Invisible Watermark Embedding

Firstly, we obtain IIBFs $\boldsymbol{f}_k\,(k=1,2,\cdots,m)$ of the host image $\boldsymbol{G}$ by ICA. Then, we do DCT transform on each IIBF, and get m DCT coefficient matrixes as follows:

$$\boldsymbol{D}_k = DCT(\boldsymbol{f}_k), \quad k=1,\cdots,m \tag{3}$$

Middle-frequency coefficient matrixes $\boldsymbol{M}_k=(m_{ij})_k, i=1,\cdots,n; j=1,\cdots,n; k=1,\cdots,m$ of $\boldsymbol{D}_k$ are restored as key matrixes and chosen to embed invisible watermark.

We calculate the image signature of the host image by our algorithm, and divide it into *m* segments. These segments $\boldsymbol{s}_k=(s_{ij})_k, i=1,\cdots,n; j=1,\cdots,n; k=1,\cdots,m$ are embedded in the corresponding middle-frequency coefficient matrixes as watermark by

$$m_{ij}^w=\begin{cases} m_{ij}+\alpha, & s_{ij}=1 \\ m_{ij}-\alpha, & s_{ij}=0 \end{cases} \quad (i=1,\cdots,n; j=1,\cdots,n) \tag{4}$$

where $\boldsymbol{M}_k^w=(m_{ij}^w)_k, i=1,\cdots,n; j=1,\cdots,n; k=1,\cdots,m$ are the middle-frequency coefficient matrixes of watermarked image and α is the embedding intension.

Then do the inverse processing as follows:

$$\boldsymbol{f}_k^w = DCT^{-1}(\boldsymbol{D}_k^w), \quad k = 1,\cdots,m \tag{5}$$

$$\boldsymbol{i}_k^w = \boldsymbol{B}^{-1} * \boldsymbol{f}_k^w, \quad k = 1,\cdots,m \tag{6}$$

where $\boldsymbol{D}_k^w = (d_{ij}^w)_k, i = 1,\cdots,N/m; j = 1,\cdots,N/m; k = 1,\cdots,m$ are DCT coefficient matrixes of watermarked image. Finally, the watermarked image $\boldsymbol{G}^w$ which is composed of $\boldsymbol{i}_k^w (k = 1,\cdots,m)$ is obtained.

4.2 The Invisible Watermark Detection

In the detection, we can get the IIBFs as the processing in the subsection 4.1. Now what we do with is $\boldsymbol{G}_w'$, which is not the same as $\boldsymbol{G}_w$ for some attack. First we get IIBFs $\boldsymbol{f}_k^{w'} (k = 1,2,\cdots,m)$ of $\boldsymbol{G}_w'$ by ICA. Then DCT transform is performed on each IIBF, and m DCT coefficient matrixes are obtained by

$$\boldsymbol{D}_k^{w'} = DCT(\boldsymbol{f}_k^{w'}), \quad k = 1,\cdots,m \tag{7}$$

Middle-frequency coefficient matrixes $\boldsymbol{M}_k^{w'} = (m_{ij}^{w'})_k, i = 1,\cdots,n; j = 1,\cdots,n; k = 1,\cdots,m$ are chosen from $\boldsymbol{D}_k^{w'}$, and key matrixes $\boldsymbol{M}_k^{w'} = (m_{ij}^{w'})_k, i = 1,\cdots,n; j = 1,\cdots,n; k = 1,\cdots,m$ are utilized to extract the image signature segments in each block as follows

$$s_{ij}' = \begin{cases} 1, & m_{ij}^{w'} - m_{ij} \geq 0 \\ 0, & m_{ij}^{w'} - m_{ij} < 0 \end{cases} \quad (i = 1,\cdots,n; j = 1,\cdots,n) \tag{8}$$

Finally, we can obtain the extracted image signature $\boldsymbol{S}'$ which is combined by m extracted image signature segments.

Normalized cross-correlation (NC) is used as the objective evaluation criterion:

$$\mathrm{NC} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} \boldsymbol{W}(i,j) \cdot \boldsymbol{W}'(i,j)}{\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N} \boldsymbol{W}^2(i,j) \sum_{i=1}^{M}\sum_{j=1}^{N} \boldsymbol{W}'^2(i,j)}} \tag{9}$$

In this paper, $\boldsymbol{W}$ denotes the extracted image signature and $\boldsymbol{W}'$ denotes the watermarked image signature calculated by our algorithm. If the NC value is higher than threshold value 0.7, we determine that legal watermark exists in the image.

5 Computer Simulation Experiments

To apply the invisible watermarking method to protect the visible-watermarked image, we do experiments as follows. We first embed visible watermark to the host image. In the experiment, the visible watermark embedding strategy is based on the method in

[6]. Then, we embed invisible watermark in the visible-watermarked image. Finally, we perform visible watermark removal attack to the visible- and invisible-watermarked image using scheme in [7]. The process is shown in fig.3.

(a) (b) (c) (d)

Fig. 3. (a) The standard 'House' image of size 128×128 ; (b) The visible-watermarked image; (c) The visible- and invisible-watermarked image; (d) The attacked image

Table.1 shows the comparison on NC using different invisible watermark methods: A is the proposed method, and B is referring [8]. Assume that the watermarked image suffers following attacks: (a) JEPG compression (b) 3×3 median filtering; (c)Gaussian noise with mean 0.1 and variance 0.0015; (d) the" bilinear" : extend the image twice as much as the original image , and then resize it the same as the original image; (e)copy attack: assume that someone has obtained the image signature from watermarked image 'House', and embed it in the illegal image 'Cameramen'; (f)visible watermark removal, while applying to protect visible-watermarked image. In this table, most of the tested NC values are above the threshold and our method shows better robustness against the same common image attacks. The data indicate that A can resist copy attack ($NC < 0.7$), while B can not. In addition, we can exactly claim the ownership of the image after visible watermark removal by the help of invisible watermark.

Table 1. Detection results under different attacks

		No attack	(a)	(b)	(c)	(d)	(e)	(f)
NC	A	1	0.9297	0.8525	0.8949	0.9836	0.5232	0.9805
	B	1	0.8493	0.7129	0.7449	0.9266	1	0.8781

We choose 6 standard images and 6 natural images as illegal images to test the robustness of our method against copy attack. The calculated NC values are shown in Table 2. They are all lower than the threshold value 0.7, which indicates that there is no legal invisible watermark in images.

Table 2. Robustness to copy attack

	Peppers	Elaine	Airplane	Bridge	Man	Lake
NC	0.4873	0.5054	0.4717	0.4483	0.4600	0.4834
	Nature1	Nature2	Nature3	Nature4	Nature5	Nature6
NC	0.5053	0.4757	0.4873	0.4639	0.4546	0.4951

6 Conclusions

In this paper, an ICA-based invisible watermark method with image signature is proposed. The simulations show the robustness of this method. Furthermore, our invisible method is applied to protect visible-watermarked image. We can exactly claim the ownership of the image after visible watermark removal by the help of invisible watermark.

Acknowledgements. This work is supported by the Program for New Century Excellent Talents in University (NCET), Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry ([2005]55), Open Foundation of National Laboratory of Networking and Switching, Beijing University of Post and Telecommunications (07), Open Foundation of National Laboratory on Machine Perception, Peking University (0403) and Specialized Research Fund for the Doctoral Program of Higher Education (20050422017).The corresponding author is Ju Liu.

References

1. Kutter, M., Voloshynovskiy, S., Herrigel, A.: The Watermark Copy Attack, Proc. SPIE, 3971 (2000) 371-379
2. Barr, J., Bradley, B., Hannigan, B. T.: Using Digital Watermarks with Image Signatures to Mitigate the Threat of the Copy Attack, ICASSP'03, 3 (2003) 69-72
3. Cardoso, J.-F.: Blind Signal Separation: Statistical Principles, proceedings of the IEEE, vol.9, no.10 (1998) 2009-2046
4. Liu, Ju., Sun, J., Du, Z.F. and Wan, Y.: Embodying Information into Images by an MMI-Based Independent Component Analysis Algorithm, ICSP'02, 2(2002) 1600-1603
5. Turk, M. and Pentland, A.: Eigenfaces for Recognition, Journal of Cognitive Neuroscience, vol.3, no.1 (1991) 71-86
6. Wong, P. W., Memon, N.: Secret and Public Key Image Watermarking Schemes for Image Authentication and Ownership Verification, IEEE Trans. on Image Processing, vol.10 (2001) 1593-1601
7. Huang, C. H., Wu, J.L.: Attacking Visible Watermarking Schemes, IEEE Trans. on Multimedia, vol.6, no.1 (2004) 16-30
8. Sun, J.D., Liu, J.: A Novel Watermark Scheme Based on Image Independent Feature, RISS'03, vol.2 (2003) 1333 - 1338

Robust Halftone Image Watermarking Scheme Based on Neural Networks

Xiang-yang Wang and Jun Wu

School of Computer and Information Technology, Liaoning Normal University, Dalian 116029, China
wxy37@263.net

Abstract. In this paper, a novel robust halftone image watermarking algorithm based on neural networks is presented. By using neural networks, the error diffusion filter can be adjusted adaptively, which can integrate the data embedding with the halftone processing in our scheme. Experimental results show that the proposed halftone image watermarking is invisible and robust against various signal processions such as JBIG compression, noise adding, cropping, daubing and print-scan. Especially, the algorithm can extract the watermark without the help from the original image.

1 Introduction

As a special kind of image, halftone images have been widely used in many printer and publishing applications. With the rapid development of the network and multimedia technique, the protection of intellectual property rights for halftone images has been the key problem that we must solve.

In recent years, a few halftone image watermarking schemes have been proposed. J. P. Allebach et al. [1] proposed an algorithm that combining the spread spectrum watermarking technique with the direct binary search technique, which hides watermark in binary images. Yet, the imperceptibility and robustness of this algorithm are poor. G. N. Zhang et al. [2] proposed an algorithm for halftone images by stochastic error diffusion. But this algorithm is not robust against print-scan, and needs help of the original image in the watermark detecting process. Ming Sun Fu et al. [3] proposed new halftone image watermarking by using error correction code (ECC). However, since this kind of algorithm introduces ECC, it is inevitable that the amount of embedded data is decreased. Besides, ECC, if the error exceeds a special range, would be invalid, in other words, it can only withstand some weak attacks. S. Z. Niu et al. [4] proposed an algorithm that embeds data by DCT mid-frequency coefficients comparison. Although this algorithm can withstand print-scan, it is not robust against other attack such as noise adding, cropping, i.e., also the amount of embedded data is very limited (only 56 bits). C. Y. Xu [5] proposed an algorithm based on conditional probability. The algorithm opens a new window in halftoning and watermarking field. Nevertheless, since this algorithm ignores the HVS, the visual quality of halftone images generated by this algorithm is poor.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 299 – 305, 2006.

In this paper, the neural networks theory is introduced in the halftone image watermarking scheme for the fist time, and a robust halftone image watermarking is proposed. It can adjust the error diffusion filter adaptively such that the watermark embedding and the halftone processing would complete simultaneously. Especially, the algorithm can extract the watermark without help from the original image.

2 Halftoning Techniques and Error Diffusion

Halftoning is a process to convert multi-level gray or color images to two-level images. The halftoning techniques are widely used in many printer and publishing applications. Halftone images, obtained by halftoning, are widely used in books, magazines, printer outputs, and fax documents.

There are two main kinds of halftoning techniques: ordered dithering [6] and error diffusion [7]. Ordered dithering is computationally simple. It compares the pixel intensities with some pseudo-random threshold patterns or screens in order to determine its two-tone output. Error diffusion is more complex than ordered dithering, but it can yield higher visual quality. It compares the sum of image pixel intensity and error from the past with a fixed threshold to determine the output. The halftoning error is fed forward to its adjacent neighbors using a filter. Two commonly used filters, the Jarvis and Steinberg filter, are shown in Fig. 1. Different filters generate halftone images with significant different characteristics. For simplicity, the Steinberg filter will be used in the rest of this paper.

	(0,0)	7/16
3/16	5/16	1/16

(a)

		(0,0)	7/48	5/48
3/48	5/48	7/48	5/48	3/48
1/48	3/48	5/48	3/48	1/48

(b)

Fig. 1. Two kinds of error diffusion filters: (a) is Steinberg filter, and (b) is Jarvis filter

Let I be the original multi-tone image, Y be the halftone image. The error diffusion operation works as follows:

$$a(i,j)=[e(i-1,j-1)+5e(i-1,j)+3e(i-1,j+1)+7e(i,j-1)]/16 \tag{1}$$

$$f(i,j)=g(i,j)+a(i,j) \tag{2}$$

$$y(i,j)=\begin{cases}0, f(i,j)<128\\ 1, f(i,j)\geq 128\end{cases} \tag{3}$$

$$e(i,j)=f(i,j)-y(i,j) \tag{4}$$

Where $g(i,j)$ is the current pixel value; $y(i,j)$ is the halftone pixel (black or white) and $e(i,j)$ is the halftone error of image pixel at location (i,j).

3 Neural Networks

3.1 Neuron Model and LMS Algorithm

Adaptive Linear Neuron (Adaline) is a kind of neurons that widely used in neural networks.

Let X be the input vector:

$$X = (x_0, x_1, \cdots, x_n)^T \tag{5}$$

W be the weight vector:

$$W = (w_0, w_1, \cdots, w_n)^T \tag{6}$$

The net be generated by:

$$net = X^T \cdot W = X \cdot W^T \tag{7}$$

The output y is determined by function $f(t)$:

$$y = \frac{1}{2}(f(net - T) + 1),\ f(t) = sqn(t) = \begin{cases} 1, t \geq 0 \\ -1, t < 0 \end{cases} \tag{8}$$

Where T is the threshold, the neuron can only being activated when $net - T \geq 0$.

The weight vector W can be trained by Least Mean Square (LMS) algorithm, which is described as follow:

$$W_{k+1} = W_k + \eta \cdot \frac{\varepsilon_k \cdot X}{\|X\|^2} \tag{9}$$

Where k is the adaptive period, W_k and W_{k+1} is the current and next period weight vector value, η is a positive number, which decides the training velocity. The linear error ε_k is defined as the difference between the ideal output d_k and the actual output y_k :

$$\varepsilon_k = d_k - y_k \tag{10}$$

According (10), $\Delta\varepsilon$ will be

$$\Delta\varepsilon = \Delta(d - W^T \cdot X) = -\Delta W^T \cdot X = -\frac{\eta}{\|X\|^2}\varepsilon \cdot X^T \cdot X = -\eta \cdot \varepsilon$$

From above formula, we can know that $\Delta\varepsilon$ and ε always have the opposite expression. That means, in the training, absolute value of ε is monotone decreasing, and y will be always close to d. Therefore, LMS algorithm can assure the convergence of Adaline.

3.2 Neuron Based Error Diffusion

In a sense, we can consider the Steinberg filter as the Adaline neuron. Let $X=[e(i-1,j-1) \quad e(i-1,j) \quad e(i-1,j+1) \quad e(i,j-1)]^T$ be the input vector, $W=\left[1 \quad 5 \quad 3 \quad 7\right]^T/16$ be the weight vector. The error diffusion operation then can be re-described as follows:

$$a(i,j)=X\cdot W^T=\begin{bmatrix} e(i-1,j-1) \\ e(i-1,j) \\ e(i-1,j+1) \\ e(i,j-1) \end{bmatrix}\cdot\begin{bmatrix} \frac{1}{16} & \frac{5}{16} & \frac{3}{16} & \frac{7}{16} \end{bmatrix} \tag{11}$$

$$\begin{aligned} y(i,j)&=\frac{1}{2}(f(net-T)+1)=\frac{1}{2}\{sqn\left[g(i,j)+a(i,j)-T\right]+1\} \\ &=\frac{1}{2}\{sqn\left[f(i,j)-T\right]+1\} \end{aligned} \tag{12}$$

Where $T=128$, $f(i,j)$ and $e(i,j)$ are continue to be generated by using (2) and (4). Instead of (1) and (3), we modify $a(i,j)$ and $y(i,j)$ to be computed as in (11) and (12).

4 Embedding Scheme

Let $I=\{g(i,j),1\le i\le M,1\le j\le N\}$ represent the host image, and $g(i,j)$ denotes the pixel gray value at (i,j). $S=\{s(i,j),1\le i\le P,1\le j\le Q\}$ is a binary image to be embedded within the host image, and $s(i,j)\in\{0,1\}$ is the pixel value at (i,j). The main steps of the embedding procedure developed can be described as follows.

4.1 Watermark Pre-processing

In order to dispel the pixel space relationship of the binary watermark image, and improve the security performance of the whole digital watermark system, the watermark scrambling method is used at first. In our watermark embedding scheme, the binary watermark image is scrambled from S to S_1 by using Arnold transform.

And then, we transform S_1 into the 1-dimentional watermark sequences M

$$M=\{m(k),1\le k\le P\times Q,m(k)\in\{0,1\}\}.$$

4.2 Embedding

First of all, a pseudo-random number generator with a known seed is used to generate a set of $P\times Q$ pseudo-random locations.

Then, the host image I is scanned by using (2), (4), (11) and (12). Let (m,n) be any pseudo-random location at which one data bit is to be embedded. For any location

(m,n), with a probability of 0.5, the actual output $y(m,n)$ is equal to the embedded data $m(k)$. In these cases, the data are embedded naturally and thus no change is needed. However, with a probability of 0.5, $y(m,n)$ is different from $m(k)$. To solve this problem, we use neural networks algorithm to train $y(m,n)$ in order to make it is equal to $m(k)$. Let $m(k)$ be the ideal output $d(k)$, viz. $d(k)=m(k)$. The actual output $y(m,n)$ would be trained as follows (pseudo-codes):

if $d(k)!= y(m,n)$ /*The ideal output is different from the actual output */
then { /*Compute the linear error using formula (9) */
$\varepsilon = d(k) - y(m,n)$
repeat {
$W_{k+1} = W_k + (\eta \cdot \varepsilon \cdot X) \Big/ \|X\|^2$ /*Train weight vector using (10) */
$a^*(m,n) = X^T \cdot W_{k+1}$
if $g(m,n) + a^*(m,n) \geq T$
then $y^*(m,n) = 1$
else $y^*(m,n) = 0$
$\varepsilon^* = d(k) - y^*(m,n)$ }
until $\varepsilon^*!= 0$ }
else no change

After being trained, the weight vector must be updated so as to the sum of coefficients are not equal to 1, which would debase the visual quality of the halftone image. So the sum of each W coefficient should be equal to 1 by using formula (13).

$$W^* = W \Big/ \sum_{i=1}^{4} w_i \tag{13}$$

Finally, the watermarked halftone image I^* can be achieved by above computation.

5 Detecting Scheme

A blind watermark detection method is proposed in this paper. Let I^* represent the image to be examined. Our watermark detecting scheme is comprised of the following components.

1) The same seed is used to generate a set of $P \times Q$ pseudo-random locations. The binary sequence

$$M^* = \{m^*(k), 1 \leq k \leq P \times Q, m^*(k) \in \{0,1\}\}$$

can be obtained by reading the halftone values at these locations.

2) The inverse process to step 4.1 is performed, and the digital watermark image S^* is achieved.

$$S^* = \{s^*(i, j), 1 \le i \le P, 1 \le j \le Q, s^* \in \{0,1\}\}.$$

6 Experimental Results

In order to illustrate the imperceptible and robust nature of our watermarking scheme, the proposed watermarking algorithm is applied to 3 test images (512×512×8bit Lena, Mandrill, and Barbara), and the Normalized Cross-correlation (NC) and the Peak Signal-to-Noise Ratio (PSNR) of digital image signal are given. The 32×32 binary image 'Jun' is used as digital watermark for all host images. The training constant η is set to 0.6.

Table 1. PSNR between watermarked halftone image and original image （dB）

Image	Proposed Scheme	Scheme [5]
Lena	6.7052	6.5270
Mandrill	6.6679	6.2480
Barbara	7.0474	6.8812

Table 1-2 summarize the proposed watermark detection results comparing with that of scheme [5] against various attacks.

Table 2. Robustness against various attack （NC）

Attack	Lena		Mandrill	
	Proposed Scheme	Scheme [5]	Proposed Scheme	Scheme [5]
JBIG2 Compression	0.9821	0.9800	0.9708	0.9619
Noise Adding	0.9403	0.9327	0.9370	0.9131
Cropping	0.7220	0.7169	0.7220	0.7083
Daubing	0.9584	0.8614	0.9248	0.8297

7 Conclusion

Base on neural networks, a robust halftone image watermarking scheme is proposed in this paper. It can adjust the error diffusion filter adaptively by using neural networks such that the watermark embedding and the halftone processing would complete simultaneously. Experimental results show that the proposed halftone image watermarking is invisible and robust against various signal processions such as JBIG compression, noise adding, cropping, daubing. In addition, blind detection, easy calculation and implementation are all contributed to the practicality of the proposed system.

References

1. Kacher, D., Allebach, J.P.: Joint Halftoning and Watermarking. In: Proc. Int. Conf. Image Processing, Vol. 2. (2000) 487–489
2. Zhang, G., Wang, S., and Wen, Q.: Watermarking Algorithm for Halftone Images by Stochastic Error Diffusion. Journal of Jilin University 34 (4) (2004) 639–643
3. Fu, M. S., Au, O. C.: Data Hiding Watermarking for Halftone Images. IEEE Transactions on Image Processing 11(4) (2002) 477–484
4. Niu, S., Niu, X., Yang, Y., Hu, W.: Data Hiding Algorithm for Halftone Images. Acta Electronica Sinica 32(7) (2004) 1180–1183
5. Xu, C.: Digital Halftoning Image Watermarking Based on Conditional Probability. Kaohsiung: National Kaohsiung First University of Science and Technology (2002)
6. Bayer, B.E.: An Optimum Method for Two-level Rendition of Continuous-tone Pictures. In: Proc. IEEE Int. Conf. Commun. (1973) 2611–2615
7. Floyd, R.W., Steinberg, L.: An Adaptive Algorithm for Spatial Grayscale. In: Proc. SID Int. Symp Tech Papers (1976) 75–77

A Blind Source Separation Based Multi-bit Digital Audio Watermarking Scheme

Xiaohong Ma, Xiaoyan Ding, Chong Wang, and Fuliang Yin

School of Electronic and Information Engineering,
Dalian University of Technology,
Dalian 116023, China
maxh@dlut.edu.cn

Abstract. A multi-bit digital audio watermarking scheme based on blind source separation (BSS) is proposed in this paper. The host audio signal is first permuted and divided into frames equally, and the watermark embedding and extraction procedure are executed frame by frame. In embedding procedure, the embedding watermark signal is generated firstly, and then mixed together with the permuted host audio signal through a BSS model. After that, the obtained mixtures are combined and inverse permuted to form the watermarked audio signal. In extraction procedure, watermarked audio signal is permuted and split into two signals, and then BSS technique is applied to obtain the separated embedding watermark signal, by observing the correlation values between it and the embedding watermark signal, the existence of watermark can be determined. With the secret keys, we can recover the watermark image further more. Experimental results show the effectiveness of the proposed method.

1 Introduction

Recently, as an effective way of copyright protection and integrity authentication, digital watermark scheme has attracted much attention [1]. For different applications, different kinds of watermark scheme are desired. To ensure the integrity of the information, fragile watermark scheme is employed [2], [3], [4], and for copyright protection, robust watermark scheme is a good choice [1], [5], [6], [7], [8]. The scheme we discussed belongs to the latter. According to the embedding capacity, watermarking scheme can be divided into one-bit watermark scheme and multi-bit watermark scheme. In [7], the authors proposed a one-bit audio watermarking scheme based on BSS. The scheme proposed here has made some improvements on [7], and it is a scheme of multi-bit watermarking.

In our work, the watermark embedding and extraction procedure are executed frame by frame. The watermark signal is generated firstly, and then the watermark embedding process is completed by a BSS mixing model. The watermark extraction is achieved by using BSS technique [9]. Experimental results show the robustness and effectiveness of the proposed method.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 306 – 311, 2006.

2 Watermark Embedding

The block diagram of watermark embedding is shown in Fig.1. There are three main steps, the embedding watermark signal generation, which is enclosed with dashed line, the watermark embedding and the post-processing enclosed with another dashed line.

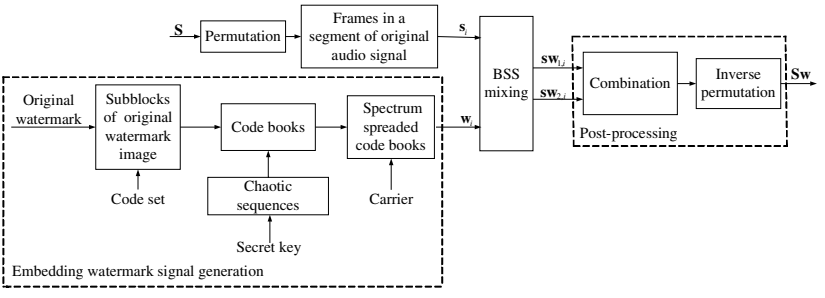


Fig. 1. The block diagram of watermark embedding

2.1 Embedding Watermark Signal Generation

The embedding watermark signal generation is composed of the following three parts.

1. Code book generation

The original watermark is a binary image with the size of 32×32. It is divided into 256 sub-blocks of 2×2 at first, which are denoted as $\mathbf{V}=[\mathbf{v}_0,\mathbf{v}_1,\cdots,\mathbf{v}_j,\cdots,\mathbf{v}_{255}]^T$, where $\mathbf{v}_j=[v_{j,0},v_{j,1},v_{j,2},v_{j,3}]^T$ is a vector with 4 elements whose values are 0 or 1. We can use the binary number $v_{j,0}v_{j,1}v_{j,2}v_{j,3}$ to represent $\mathbf{v}_j$. There are totally 16 kinds of combination. For safety consideration, we put a random permutation on them, and then a code set $\mathbf{C}=[\mathbf{c}_0,\mathbf{c}_1,\cdots,\mathbf{c}_i,\cdots,\mathbf{c}_{15}]^T$ which is kept as a secret key can be created. To represent the relationship between the value of each element in the code set and that of each sub-block, we define a code books matrix $\mathbf{B}=[\mathbf{b}_0,\mathbf{b}_1\cdots,\mathbf{b}_i,\cdots,\mathbf{b}_{15}]^T$, where $\mathbf{b}_i$ is a code book vector with 256 elements. For every element of $\mathbf{B}$, $b_{i,j}$ ($i=0,1,\cdots,15$, $j=0,1,\cdots,255$) can be defined as follows:

$$b_{i,j}=\begin{cases}1, & if\ \mathbf{c}_i=\mathbf{v}_j\\ -1, & else\end{cases} \tag{1}$$

2. Spreading codes generation

To improve the security of our method further more, we use chaotic sequences [10] to spread the obtained code books. As the chaotic sequence is sensitive to the initial condition and difficult to conclude the initial value from the chaotic sequence with finite length, we can use it as spreading code to improve the security of the watermark. Utilizing the delta-like correlation characteristic of chaotic sequence, we can achieve the purpose of embedding multi-information into the host signal.

3. Embedding watermark signal generation
For each code book, we generate different chaotic sequence as its spreading code. The spread spectrum code books are modulated further more by a carrier, whose frequency is denoted as f. Then 16 kinds of embedding watermark signals are obtained, which are denoted as $\mathbf{W}=[\mathbf{w}_0,\mathbf{w}_1,\cdots,\mathbf{w}_i,\cdots,\mathbf{w}_{15}]^T$, and the length of $\mathbf{w}_i$ is K. Here, the initial values are also kept as secret keys in the watermark extraction procedure.

2.2 Watermark Embedding

To improve the security and robustness of our scheme, we permute the host signal and select several segments of it to embed in the embedding watermark signal. Here, we take one segment of them to illustrate the embedding procedure.

Firstly, we divide the selected segment into many frames, the number of which must be equal to or greater than that of the spread code book, while each frame and the embedding watermark signal corresponding to each code book should share the same length. We denote the permuted host signal as $\mathbf{S}=[\mathbf{s}_0,\mathbf{s}_1,\cdots,\mathbf{s}_i,\cdots,\mathbf{s}_P]^T$, where $P\geq 16$, and $\mathbf{s}_i$ represents the i th frame.

The watermark embedding process is taken frame by frame. For each frame, $\mathbf{s}_i$ and $\mathbf{w}_i$ are mixed according to [9]. The mixing procedure is illustrated in Eq.(2)

$$\mathbf{x}=\begin{bmatrix}\mathbf{sw}_{1,i}\\ \mathbf{sw}_{2,i}\end{bmatrix}=\mathbf{A}\cdot\mathbf{S}=\begin{bmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix}\begin{bmatrix}\mathbf{s}_i\\ \mathbf{w}_i\end{bmatrix}=\begin{bmatrix}a_{11}\mathbf{s}_i+a_{12}\mathbf{w}_i\\ a_{21}\mathbf{s}_i+a_{22}\mathbf{w}_i\end{bmatrix} \tag{2}$$

where the mixing matrix $\mathbf{A}$ is a matrix of full rank. To ensure the robustness and transparency of our method, a_{11} should be much greater than a_{12}, and it is also true for a_{21} and a_{22}. $\mathbf{sw}_{1,i}$ and $\mathbf{sw}_{2,i}$ are two mixtures with length K.

The process is going on until all the frames in the segment are watermarked.

2.3 Post-processing

To decrease the number of secret keys, the two mixtures $\mathbf{sw}_{1,i}$ and $\mathbf{sw}_{2,i}$ should be combined together.

$$\begin{cases}\mathbf{msw}_i(2k-1)=\mathbf{sw}_{1,i}(k),\\ \mathbf{msw}_i(2k)=\mathbf{sw}_{2,i}(k),\end{cases}\qquad k=0,1,\cdots,K-1,\quad i=0,1,\cdots,15 \tag{3}$$

where $\mathbf{msw}_i$ is the combined signal in each frame whose length is $2K$.

To ensure the audible quality, after finishing the above work, the sampling rate of the watermarked audio signal is doubled and then inverse permuted to generate the final watermarked audio signal $\mathbf{SW}=[\mathbf{Sw}_0,\mathbf{Sw}_1,\cdots,\mathbf{Sw}_i,\cdots\mathbf{Sw}_{15}]^T$, where $\mathbf{Sw}_i$ denotes the watermarked audio signal in the i th frame with the length of $2K$.

3 Watermark Extraction

Just like the watermark embedding procedure, we take one segment of the watermarked audio signal to illustrate the extraction process, as described in Fig.2. We permute the watermarked audio signal and divide it into several segments at first, and as the watermarked audio signal may have been attacked during its transmission, we denote one segment of it as $\mathbf{SW}' = [\mathbf{Sw}'_0, \mathbf{Sw}'_1, \cdots, \mathbf{Sw}'_i, \cdots \mathbf{Sw}'_{15}]^T$, where $\mathbf{Sw}'_i$ represents the i th frame of the watermarked audio signal in this segment. For each segment, we perform the following works:

(i) Two observed signals are obtained according to Eq.(4)

$$\begin{cases} \mathbf{sw}'_{1,i}(k) = \mathbf{Sw}'_i(2k-1), \\ \mathbf{sw}'_{2,i}(k) = \mathbf{Sw}'_i(2k), \end{cases} \quad k = 0,1,\cdots,K-1 \quad i = 0,1,\cdots,15 \tag{4}$$

Applying FastICA method [9] on these two observed signals, the estimation of the permuted host signal and the embedding watermark signal can be obtained. As there is no telling which one of them is the embedding watermark that we needed, the fourth order statistics (kurtosis) is resorted to. We denote the extracted embedding watermark signal as $\mathbf{W}' = [\mathbf{w}_0', \mathbf{w}_1' \cdots, \mathbf{w}'_i, \cdots, \mathbf{w}'_{15}]^T$.

(ii) The existence of embedding watermark signal in this frame is detected by calculating the correlation coefficient between the extracted embedding watermark signal and the embedding watermark signal reproduced with the secret keys. The correlation coefficient is calculated according to the following formula:

$$\rho_{ww'}(i) = \left[\sum_{k=0}^{K-1} \mathbf{w}_i(k)\mathbf{w}'_i(k)\right] \Bigg/ \left[\sum_{k=0}^{K-1} \left|\mathbf{w}_i(k)\right|^2 \sum_{k=0}^{K-1} \left|\mathbf{w}'_i(k)\right|^2\right]^{1/2} \tag{5}$$
$$i = 0,1,\cdots,15, \quad k = 0,1,\cdots,K-1$$

(iii) According to the following rule, the embedded code books can be recovered:

$$b'_{i,j} = \begin{cases} 1, & \left|\rho_{ww'}(i)\right| \geq th \\ -1, & else \end{cases}, \quad i = 0,1,\cdots,15, \quad j = 0,1,\cdots,255 \tag{6}$$

where $b'_{i,j}$ is the j th element in the i th extracted code book, and th is the threshold used to judge whether the embedding watermark signal exists or not. If $b'_{i,j} = 1$, we assume that there exists watermark, otherwise, the watermark has been destroyed or it doesn't exist at all.

(iv) The same process is going on until all the frames in the segment are detected.

(v) Finally, the watermark image can be recovered by executing the reverse code book mapping, which can be described as:

$$\mathbf{v}'_j = \begin{cases} \mathbf{c}_i, & b'_{i,j} = 1 \\ 0, & b'_{i,j} = -1 \end{cases}, \quad i = 0,1,\cdots,15, \quad j = 0,1,\cdots,255 \tag{7}$$

where $\mathbf{v}'_j$ is the recovered sub-block corresponding to $b'_{i,j}$.

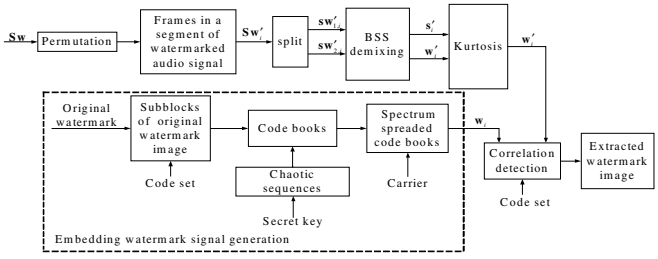


Fig. 2. The block diagram of watermark extraction

4 Experimental Results

The parameters used in our experiment are given as follows: $K = 1024$, $P = 16$, $f = 100\text{Hz}$, $a_{11} = 1$, $a_{12} = 0.05$, $a_{21} = 1$, and $a_{22} = 0.01$, $th = 0.5$. The waveform of original audio signal whose sampling rate is 44.1 kHz and length is 32768, and watermarked audio signal whose sampling rate is 88.2 kHz and length is 65536 are shown in Fig.3 and Fig.4 respectively. We can see that there is no distortion between them, and it is also true in listening test. The original watermark image is shown in Fig.5(a), and the extracted watermark without any attack is shown in Fig.5(b); Fig.5(c)-Fig.5(g) show the extracted watermarks under various attacks. We can see that all the extracted watermarks except Fig.5(g) are the same with the original one; though under the attack of adding white Gaussian noise with the SNR of 21 dB, the embedding watermark signal has been slightly degraded, the extracted watermark can still be recognized clearly.

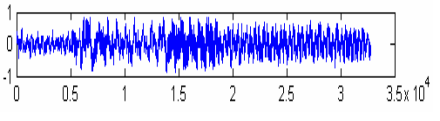


Fig. 3. Original audio signal

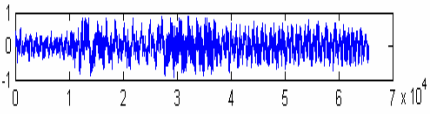


Fig. 4. Watermarked audio signal

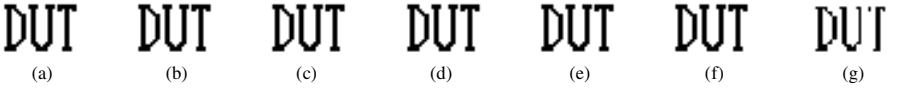


Fig. 5. The original watermark image and extracted watermark images under various conditions. (a) original watermark image. (b) extracted watermark image without any attack. (c) mp3 compression. (d) cropping. (e) lowpass filtering (cut off frequency is 16kHz). (f) resampling (from 44.1k to 88.2 kHz, then to 44.1 kHz). (g) adding white Gaussian noise (SNR is 21dB).

5 Conclusion

A robust multi-bit audio watermarking scheme based on BSS for blind watermark extraction is proposed in this paper. It has many advantages over traditional methods:

the watermark can be extracted blindly; what is more, we can not only detect the existence of the watermark, but also extract it if only it exists. The experimental results show that our method is robust enough against common signal processing operations, such as additive noise, cropping, lowpass filtering, resampling, and mp3 compression.

As an extension of our work, we will try to realize a BSS based multipurpose watermark system to achieve copyright protection and integrity authentication simultaneously.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant No. 60575011, No. 60372082 and the Liaoning Province Natural Science Foundation of China under Grant No. 20052181.

References

1. Liu, J., Zhang, X. G., Najar, M., Lagunas, M. A.: A Robust Digital Watermarking Scheme Based on ICA. International Conference on Neural Networks and Signal Processing, Vol. 2 (2003) 1481-1484
2. Watanabe, J., Hasegawa, M., Kato, S.: A Study on a Watermarking Method for Both Copyright Protection and Tamper Detection. International Conference on Image Processing, Vol. 4 (2004) 2155-2158
3. Liu, Y., Gao, W., Yao, H. X, Liu, S. H.: A Texture-based Tamper Detection Scheme by Fragile Watermark. International Symposium on Circuits and Systems, Vol. 2 (2004) 177-180
4. Quan, X. M., Zhang, H. B.: Perceptual Criterion Based Fragile Audio Watermarking Using Adaptive Wavelet Packets. International Conference on Pattern Recognition, Vol.2 (2004) 867-870
5. Sun, J., Liu, J.: A Novel Digital Watermark Scheme Based on Image Independent Feature. International Conference on Robotics, Intelligent Systems and Signal Processing, Vol. 2 (2003) 1333-1338
6. Li, X. Y., Zhang, M., Sun, S. L.: Adaptive Audio Watermarking Algorithm Based on SNR in Wavelet Domain. International Conference on Natural Language Processing and Knowledge Engineering (2003) 287-292
7. Ma, X. H., Wang, C., Cong, X. P., Yin, F. L.: A Digital Audio Watermarking Scheme Based on Blind Source Separation. International Symposium on Neural Networks. Springer-Verlag, Berlin Heidelberg Part II (2005) 550-555
8. Toch, B., Lowe, D., Saad, D.: Watermarking of Audio Signals Using Independent Component Analysis. International Conference on Web Delivering of Music, (2003) 71-74
9. Hyvärinen, A., Oja, E.: A Fast Fixed-point Algorithm for Independent Component Analysis. Neural Computation 9 (1997) 1483-1492
10. Zhang, Z. M., Wang, L.: A DCT Domain Image Watermark Scheme Based on Chaotic Encryption (in Chinese). Computer Engineering 17 (2003) 9-11

A 2DPCA-Based Video Watermarking Scheme for Resistance to Temporal Desynchronization

Jiande Sun[1,2], Ju Liu[1,2], and Hua Yan[1]

[1] School of Information Science and Engineering, Shandong University, Jinan 250100, China
[2] National Laboratory on Machine Perception, Beijing 100871, China
{juliu, jd_sun, yhzhjg}@sdu.edu.cn

Abstract. Temporal synchronization is a problem that cannot be ignored in most video watermarking application. In this paper, a video watermarking scheme is proposed to resist temporal desynchronization by using shot segmentation and two-dimensional principal component analysis (2DPCA). In this scheme, the to-be-watermarked video is first segmented into different shots. And each shot is analyzed by 2DPCA in *xt* tomography to get its principal feature frames (PFFs). And then the watermark is embedded into the PFFs. This scheme is robust to temporal desynchronization through a content dependent way. Simulations show the feasibility and reliability of this scheme. It is also robust to most of the common video processing/attacks, such as MPEG2 compression, filtering, frame resizing, noising, etc.

1 Introduction

With the development of network and multimedia techniques, video watermark has become more attractive in researching and has been applied to DVD, VOD, interactive television, etc [1,2]. Temporal synchronization is a non-negligible problem to most of video watermarking schemes [3]. In digital television and broadcast monitoring, the watermark should be detected from any arbitrary temporal point of the video. For the case of streaming video, some parts of video signal may be damaged or lost as it is transmitted over a network, or the video signal may have been interrupted for an indeterminate time due to some reasons, such as network congestion. In these applications, it is essential for watermark to be robust against temporal desynchronization.

There are two ways to improve the robustness against temporal desynchronization. One is frame dependent scheme, which resynchronizes video temporally according to the timing of frames. For example, Deguillaume embeds a template to temporally resynchronize, which a detector can examine to determine the orientation and scale of the watermark [3]. Lin achieves the temporal synchronization by introducing redundancy in the structure of the embedded watermark [4]. But in practice, the timing of video frames is often changed while the video content is kept. In such case, the frame dependent schemes cannot resynchronize any more. So the robustness should be improved through content dependent way. This kind of schemes usually extracts video content features and embeds the watermark into them. Swanson segments the video into different scenes, and

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 312 – 317, 2006.

performs temporal wavelet on each scene to get the wavelet feature frames (WFFs) [1]. But this kind of feature still depends on the timing of frames.

In this paper, a content dependent video watermarking scheme is presented. In this scheme, the video is firstly segmented into shots and two-dimensional principal component analysis (2DPCA) is used to extract the feature of each shot. Then the watermark is embedded into these features. Simulations show it is not only robust against temporal desynchronization, e.g., frame dropping, frame swapping, fps changing, etc, but robust to most of the common video processing/attacks, such as MPEG2 compression, filtering, frame resizing, noising, and so on.

2 Two-Dimensional Principal Component Analysis (2DPCA)

2DPCA is a novel method developed for feature extraction in the research of human face recognition [5]. It is based on 2D matrices, and the size of the covariance matrix using 2DPCA is much smaller than conventional PCA. So 2DPCA is easier to evaluate the covariance matrix accurately and less time is required to determine the corresponding eigenvectors. 2DPCA is helpful for the video watermarking to reach realtime.

Let $\boldsymbol{A}$ denotes an image with the size of $m \times n$, and its m-dimensional projected feature $\boldsymbol{F}$ can be gotten by:

$$\boldsymbol{F} = \boldsymbol{A}\boldsymbol{X} \tag{1}$$

where $\boldsymbol{X}$ is an n-dimensional unitary column vector. Before the features are extracted by using 2DPCA, training images are needed to get the image covariance matrix $\boldsymbol{G}_t$ by

$$\boldsymbol{G}_t = \frac{1}{M}\sum_{j=1}^{M}(\boldsymbol{A}_j - \overline{\boldsymbol{A}})^T(\boldsymbol{A}_j - \overline{\boldsymbol{A}}) \tag{2}$$

where $\boldsymbol{A}_j$ $(j = 1,2,...,M)$ is jth training image with the size of $m \times n$, and $\overline{\boldsymbol{A}}$ is the average image. A set of optimal projection axes, $\boldsymbol{X}_1,...,\boldsymbol{X}_d$, are usually selected to project $\boldsymbol{A}$, which are the eigenvectors of $\boldsymbol{G}_t$ corresponding to the first d largest eigenvalues. And the dimension is decreased to d. So the feature of $\boldsymbol{A}$ can be extracted by:

$$\boldsymbol{F}_k = \boldsymbol{A}\boldsymbol{X}_k,\ k = 1,2,...,d \tag{3}$$

where $\boldsymbol{X}_k$ is the kth optimal projection vector. And the $m \times d$ matrix $\boldsymbol{B} = [\boldsymbol{F}_1,...,\boldsymbol{F}_d]$ is called the feature of $\boldsymbol{A}$.

3 2DPCA-Based Video Feature

The video tomography method (VTM) is an efficient method for motion analysis of video [6].Video is a *xyt* 3D signal. Usually the spatial properties of motion in video can be gotten from *xy* tomography, and the temporal properties can be gotten from *xt* or *yt* tomography.

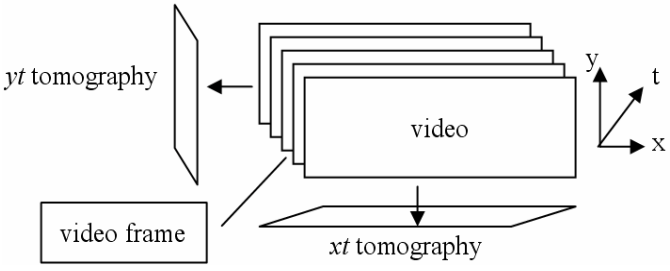


Fig. 1. Tomography of video is described. Based on the idea of VTM, in this paper, *y* is constant and *xt* tomography is analyzed by 2DPCA to get motion features of video.

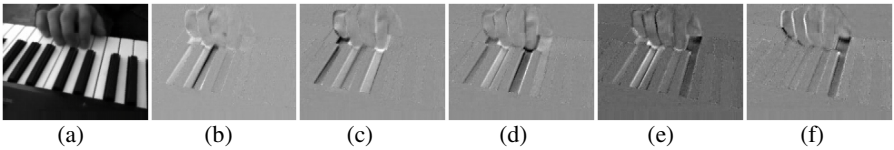


Fig. 2. The PFFs of the demo video are shown. In this video, a hand plays notes on a piano and there is only 5 keys played by each finger respectively. There is only one shot in this video. During feature extraction, the t dimension has been decreased to 6 to get principal vectors. (a) is the background feature of this video. And the other five PFFs are dynamic features, which describe the playing on the notes. These PFFs depend on the video content, so as long as the video content is not changed, these PFFs are invariable.

We perform 2DPCA on the video in *xt* tomography to get the temporal features. Then a feature video is obtained, whose frames in *xt* tomography are the 2DPCA features and frames in *xy* dimensions are called principal feature frames (PFFs) of the video in this paper. The feature frames have the same size with the frames in *xy* dimensions of original video. Besides, since during the feature extraction, the dimension is reduced by the optimal projection, the mis-segmentation can be ignored, if the total energy of mis-segmented frames is less than that of the shot.

4 Watermarking Scheme

4.1 Shot Segmentation

Shot segmentation is to divide the video temporally according to different scenes. It is well known that even if the video loses temporal synchronization, its content is still preserved. So, shot segmentation can help to improve the robustness of watermark against temporal desynchronization. Here, the global histogram comparison approach is selected as the demo [8]. The difference between two consecutive frames is:

$$Q(i,i+1)=\frac{\sum_{j=1}^{n}\left|H_i(j)-H_{i+1}(j)\right|}{P} \tag{4}$$

where $H_i(j)$, $j = 1, \cdots, n$ is the histogram value for the gray level j in the frame i, n is the total number of gray levels. P is the total number of the pixels in each frame. If the sum of histogram differences is greater than a threshold, a content changing is declared.

4.2 Embedding

In this paper, the obtained PFFs are looked on as images, where the watermark can be embedded by the Single Watermark Embedding (SWE) scheme [9].

In this scheme, $\boldsymbol{D} = [d_1, d_2, \ldots, d_N \,|\, d_i \in \Re^+ 1 \le i \le N]$ and $\boldsymbol{K} = [k_1, k_2, \ldots, k_M]$ are used as two keys, and the watermark $\boldsymbol{W} = [w_1, w_2, \cdots, w_N]$ is embedded into the host vector $\boldsymbol{Y}$ to form the watermarked one $\boldsymbol{Y}^{'}$. $\boldsymbol{Y}$, $\boldsymbol{Y}^{'}$ and $\boldsymbol{K}$ are split into N subvectors of equal length $L = \lfloor M / N \rfloor$, whose the *ith* subvector denoted as $\boldsymbol{Y}_i$, $\boldsymbol{Y}_i^{'}$ and $\boldsymbol{K}_i$, respectively. Its embedding algorithm is:

$$\boldsymbol{Y}_i^{'} = \boldsymbol{Y}_i + \alpha_i \boldsymbol{K}_i \tag{5}$$

Here, the value α_i is determined by correlation among $\langle \boldsymbol{Y}_i, \boldsymbol{K}_i \rangle$, $\boldsymbol{D}$ and $\boldsymbol{K}$. The details of watermark embedding and detection can be found in [9].

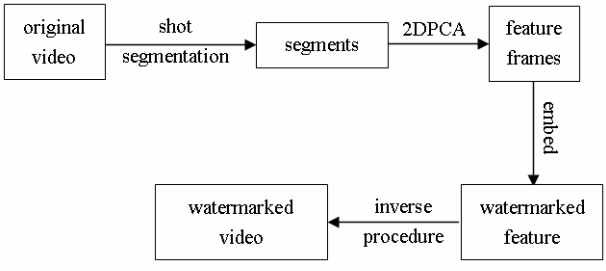


Fig. 3. The framework of the proposed watermarking scheme is described. In the proposed scheme, the original video is firstly divided into segments according to different scenes. And the PFFs are extracted from the obtained segments. And then the watermark is embedded into them.

4.3 Detection

During detection, the PFFs are extracted just as in embedding, and the watermark can be detected directly from these feature frames. $w_i^{'}$ denotes the extracted watermark. The score $S_w = \frac{1}{N} \sum_{i=1}^{N} (2 \cdot w_i - 1) \cdot (2 \cdot w_i^{'} - 1)$ is selected to evaluate the detection result. The maximum detection score of each segment is selected as the detection score of this segment. If the score is over a threshold, the segment is considered to be watermarked. As long as there is one watermarked segment, the whole video is watermarked.

5 Simulation

The simulation video is a 272×352×73 advertisement video. Its frame rate is 25fps. It has three shots. The watermark is a binary image of the four Chinese characters of Shandong University. Its size is 32×32. The threshold for detection score is 0.7.

Fig. 4. Frames and PFFs of each segment respectively. (a)-(c) are the original frames in the three segments. (d)-(f) are corresponding background PFFs, and (g)-(i) are the dynamic ones.

Fig. 5. Original watermark and detected ones after various attacks are shown. They are (a) original watermark, (b)from original video, (c) after 3×3 median filtering, (d) after frame shrink with factor 2, (e) after frame rotation with 0.5 degree, (f) after salt noising with density 0.02, (g) after Gaussian noising with mean 0 and variance 0.001, (h) after intra-video collusion in each video segment, (i) after frame swapping, (j) after frame dropping, (k) after fps changing from 25fps to 24fps, and (l) after fps changing from 25fps to 30fps.

Table 1. The comparison between PFF and WFF. In the comparison scheme, the watermark is embedded into the WFFs adopted in [1] by the SWE scheme. And all the above attacks are performed. Compared with WFF, PFF has better stability under various attacks. It is this property of PFF that makes the watermark more robust.

	(b)	(c)	(d)	(e)	(f)	(g)
PFF	0.0625	0.9258	0.9981	0.7754	0.7617	0.7578
WFF	-0.0625	0.9863	0.6230	-0.6152	-0.0625	-0.0625
	(h)	(i)	(j)	(k)	(l)	
PFF	1	1	1	0.9883	0.9902	
WFF	0.9051	0.7676	0.9941	0.9590	0.9707	

6 Conclusion and Discussion

In this paper, 2DPCA is used to get a novel temporal video feature, PFF, due to its superiority in calculation efficiency. As long as the content of a video shot is not changed, its

PFFs are invariable. The 2DPCA-based video watermarking scheme is robust to temporal desynchronization attacks, such as frame dropping, frame swapping, fps changing, etc. And it is also robust to MPEG2 compression, median filtering, Gaussian noising, resizing, frame averaging, intra-collusion, and so on. In addition, the selection of the approach of shot segmentation and the algorithm of embedding watermark into feature frames are independent of the proposed scheme. Different shot segmentation approaches and embedding algorithms can be chosen according to different videos and different applications. The realtime of this scheme depends on the number of shots in video and the duration of each shot. Less shot and longer duration of each shot result better realtime, and how to improve the realtime is the next research focus.

Acknowledgement

This work is supported by the Program for New Century Excellent Talents in University (NCET), Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry ([2005]55), Open Foundation of National Laboratory of Networking and Switching, Beijing University of Post and Telecommunications (07), Open Foundation of National Laboratory on Machine Perception, Peking University (0403) and Specialized Research Fund for the Doctoral Program of Higher Education (20050422017).The corresponding author is Ju Liu.

References

1. Swanson, M.D., Zhu, B., Tewfik, A. H.: Multiresolution Scene-Based Video Watermarking Using Perceptual Models, IEEE Journal on Selected Areas in Commun. 16(4) (1998) 540-550
2. Doerr, G., Dugelay, J.L.: A Guide Tour of Video Watermarking, Signal Processing: Image Communication 18(4) (2003) 263-282
3. Deguillaume F, Csurka G, Ruanaidh J, Pun T.: Robust 3D DFT video watermarking. In: Wong PW, Delp EJ (eds.): IS&T/SPIE's 11th Annual Symp., Electronic imageing'99: Security and Watermarking of multimedia Contents, Vol. 3657 (1999) 113□124.
4. Lin, E.T., Delp, E.J.: Temporal Synchronization in Video Watermarking, Proceedings of the SPIE Security and Watermarking of Multimedia Contents IV, Vol. 4675 (2002) 478-490
5. Yang, J., Zhang, D., Frangi, A.F., Yang, J.Y.: Two-dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition, IEEE Trans. on Pattern Analysis and Machine Intelligence 26(1) (2004) 131-137
6. Akutsu, A., Tonomura, Y.: Video Tomography: An Efficient Method for Camerawork Extraction and Motion Analysis, Proceedings of the Second ACM international conference on Multimedia (1994) 349-356
7. Liu, Y., Xu, S., Wu, Z.: A Hierarchical Retrieval Method for MPEG Video, Journal of Software (in Chinese) 14(3) (2003) 675-681
8. Koprinska, I., Carrato, S.: Temporal Video Segmentation: A Survey, Signal Processing: Image Communication 16(5) (2001) 477-500.
9. Wong, H., Au, C., Yeung, Y.: A Novel Blind Multiple Watermarking Technique for Images, IEEE Trans. on Circuit and Systems for Video Technology 13(8) (2003) 813-830

A Fast Decryption Algorithm for BSS-Based Image Encryption

Qiu-Hua Lin[1], Fu-Liang Yin[1], and Hua-Lou Liang[2]

[1] School of Electronic and Information Engineering,
Dalian University of Technology, Dalian 116023, China
qhlin@dlut.edu.cn
[2] School of Health Information Sciences,
The University of Texas at Houston, Houston, TX 77030, USA

Abstract. The image encryption based on blind source separation (BSS) takes advantage of the underdetermined BSS problem to encrypt multiple confidential images. Its security can be further improved if the number of images to be simultaneously encrypted increases. However, the BSS decryption speed will correspondingly decrease since the computational load of the BSS algorithms usually has nonlinear relation with the number of the source signals. To solve the problem, this paper presents a fast decryption algorithm based on adaptive noise cancellation by using the knowledge of the key images, which are used in the BSS-based method and available at the receiving side. As a result, the number of the source signals for the fast BSS decryption is decreased in half, and the decryption time is considerably reduced. Both computer simulations and performance analyses demonstrate the efficiency of the proposed method.

1 Introduction

With the widespread use of the computer networks in our daily life, more and more image information has been transmitted over the Internet. Image security has become an important topic in the field of information security, especially in the fields of military, economy, and diplomacy. Therefore, many image encryption techniques have been proposed so far [1], [2].

Blind source separation (BSS) aims to recover a set of source signals from their observed mixtures without knowing the mixing coefficients. It has received considerable attention in recent years and has been successfully applied in many fields such as wireless communications and biomedical engineering [3], [4]. Its application for image encryption, however, has been scarce [5], [6], [7]. Motivated by the fact that the security of many cryptosystems relies on the apparent intractability of the computational problems such as the integer factorization problem [8], we take advantage of the underdetermined BSS problem to propose a BSS-based image encrypt method [7], in which multiple confidential images are encrypted by masking them with the same size and the same number of independent key images. Since it approximately has the perfect key characteristics of the one-time pad cipher [8], which is unconditionally secure, the BSS-based method has very high level of security.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 318 – 325, 2006.

Because the multiple images are covered mutually while the key images mask them, the security of the BSS-based method can be further improved if the number of images to be simultaneously encrypted increases. However, the large number of images will be heavy burden for the BSS decryption since the computational load of the BSS algorithms usually has nonlinear relation (e.g., square, cubic, etc) with the number of the source signals. As a result, the speed of the BSS decryption will considerably decrease. Obviously, a direct solution to this problem between the security and the decryption speed is to reduce the number of the source signals for the BSS decryption. In this paper, we propose such a fast decryption method by using the knowledge of the key images at the receiving side. Specifically, adaptive noise cancellation technique is employed to half reduce the number of the source signals for the BSS decryption, thus the decryption time of the BSS algorithms is considerably shortened. The efficiency of the proposed method is shown through both computer simulation results and performance analyses.

2 BSS-Based Image Encryption

The block diagram of the BSS-based image encryption is shown in Fig. 1. $s_1(t), ..., s_P(t)$, $t = 1, \ldots, T$, are P original images to be encrypted simultaneously where T is the size (data length) of each original image; $s_{n1}(t), ..., s_{nP}(t)$ are P key images; I_0 is the secret seed for generating the key images; $x_1(t), ..., x_P(t)$ are P encrypted images for transmission; $\hat{s}_1(t), ..., \hat{s}_P(t)$ are P decrypted images by BSS [7].

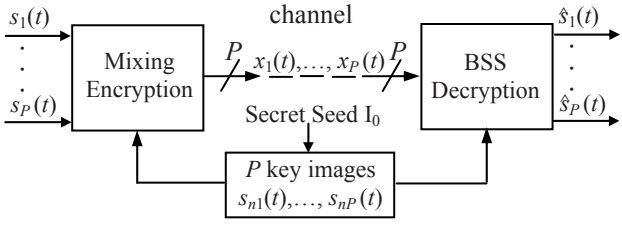


Fig. 1. Block diagram of BSS-based image encryption

The central idea of the BSS-based encryption is to construct the underdetermined BSS problem in the encryption process, and then solve the problem in the decryption process by using the key images.

2.1 Encryption

The P key images $s_{n1}(t), ..., s_{nP}(t)$ are first formed by pseudorandom values with uniform distribution, which are generated by a pseudorandom number generator initialized with the secret seed I_0 (see Fig. 1). Then, a $P \times 2P$ mixing matrix for encryption is constructed as follows [7]:

$$\mathbf{A}_e = [\mathbf{B} \quad \beta\mathbf{B}] \tag{1}$$

where $\mathbf{B}$ is a $P \times P$ matrix of full rank, which are pseudo-randomly generated with uniform distribution between -1 and 1, β is a scalar value and $\beta \geq 10$.

The BSS-based method encrypts the P original images under the mixing matrix for encryption $\mathbf{A}_e$ according to the following BSS mixing equation:

$$\mathbf{x}_P(t) = \mathbf{A}_e \, \mathbf{s}_{2P}(t) \tag{2}$$

where $\mathbf{s}_{2P}(t) = [s_1(t), ..., s_P(t), s_{n1}(t), ..., s_{nP}(t)]^T$, $\mathbf{x}_P(t) = [x_1(t), ..., x_P(t)]^T$ includes the P encrypted images, in which the original images are well masked by the key images with much higher level of energy to achieve the security goal.

Since there are $2P$ source images but P mixed images, the encryption process turns BSS into the difficult case of the underdetermined problem, in which the number of the mixed signals is less than that of the source signals. In such a case, the complete separation is out of the question [9]. The mixing matrix for encryption $\mathbf{A}_e$ in (1) has been shown inseparable by using the source inseparability theorem, i.e., Theorem 1 in [9]. As a result, the P original images cannot be recovered from the P encrypted images through BSS without the P key images [7].

2.2 Decryption

At the receiving side, when the P encrypted images $x_1(t), ..., x_P(t)$ are available, and the P key images $s_{n1}(t), ..., s_{nP}(t)$ are regenerated with the secret seed I_0, they are first combined to form $2P$ mixed images $\mathbf{x}_d(t) = [x_1(t), ..., x_P(t), s_{n1}(t), ..., s_{nP}(t)]^T$, on which the BSS algorithm is then performed. Obviously, by using the P key images, the underdetermined BSS problem in the encryption process becomes the simplest BSS in which the number of the mixed signals $\mathbf{x}_d(t)$ is equal to that of the source signals $\mathbf{s}_{2P}(t)$. That is, the equivalent mixing matrix for decryption $\mathbf{A}_d$ is a $2P \times 2P$ square matrix of full rank:

$$\mathbf{A}_d = \begin{bmatrix} \mathbf{B} & \beta\mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{3}$$

where $\mathbf{I}$ is a $P \times P$ identity matrix, $\mathbf{0}$ is a $P \times P$ zero matrix. Therefore, the P original images can be fully recovered to give the P decrypted images $\hat{s}_1(t), ..., \hat{s}_P(t)$ (see Fig. 1).

3 Fast BSS Decryption

The BSS decryption mentioned above, which is called the regular BSS decryption hereafter for distinction, is simple and straightforward, yet it is time-consuming when many original images are simultaneously encrypted to further enhance the security. The reason is that the computational load of the BSS algorithms usually has nonlinear relation (e.g., square, cubic, etc) with the number of the source signals. Therefore, to

decrease the number of the source signals for the BSS decryption is a direct solution to speeding up the decryption of multiple images.

It is known that the P key images are available at the receiving side. If they can be canceled from the encrypted images, the number of the source signals for the BSS decryption will reduce in half. As a result, the fast BSS decryption is achieved. Consider the P encrypted images $x_1(t), ..., x_P(t)$ as follows:

$$x_i(t) = \sum_{j=1}^{P} a_{ij} s_j(t) + \sum_{j=1}^{P} a_{i(P+j)} s_{nj}(t), \quad i = 1, \cdots, P \tag{4}$$

From (4), it can be seen that the P key images $s_{nj}(t)$ can be subtracted from the P encrypted images $x_i(t)$ by estimating $P \times P$ mixing coefficients $a_{i(P+j)}$, $i = j = 1, \ldots, P$, the estimation values of which are denoted by $\hat{a}_{i(P+j)}$. Then, P new mixed signals for the fast BSS decryption are

$$\hat{x}_i(t) = x_i(t) - \sum_{j=1}^{P} \hat{a}_{i(P+j)} s_{nj}(t), \quad i = 1, \cdots, P. \tag{5}$$

Consequently, the P new mixed signals $\hat{x}_i(t)$ are approximately mixtures of the P original images $s_j(t)$ as follows:

$$\hat{x}_i(t) \approx \sum_{j=1}^{P} a_{ij} s_j(t), \quad i = 1, \cdots, P. \tag{6}$$

Comparing the signals for the fast BSS decryption in (6) and those for the regular BSS decryption in (4), we can see that half number of the source signals is reduced. Hence, the computational load for the fast BSS decryption will considerably decrease, and the decryption is significantly speeded up.

The optimum $\hat{a}_{i(P+j)}$ can be estimated by using the method for adaptive noise cancellation as [10]

$$\hat{a}_{i(P+j)} = \frac{\sum_{t=1}^{T} x_i(t)\, s_{nj}(t)}{\sum_{t=1}^{T} s_{nj}^2(t)}, \quad i = j = 1, \cdots, P. \tag{7}$$

It is noted that the estimation of $\hat{a}_{i(P+j)}$ may have small errors. Thus, the P new mixed signals $\hat{x}_i(t)$ may have a little residual noise. Correspondingly, the performance of the recovered images by the fast BSS decryption may slightly degrade compared with that of the recovered images by the regular BSS decryption. However, owing to the characteristic of the human perception, a little degradation in the decrypted images is hard to identify.

4 Computer Simulations and Performance Analyses

To illustrate the efficiency of the proposed method, we carried out extensive computer simulations with nature images. Fig. 2 shows one example of encrypting two original images $s_1(t)$ and $s_2(t)$. They are "Lena" and "Girl" with 256 grey levels, the size of which is 256×256, as shown in Fig. 2(a). In the experiments, the Comon's algorithm

[11] is used for the BSS decryption. Other popular BSS algorithms suitable for separating images can also be used [3], [4]. Fig. 2(b) shows two independent key images $s_{n1}(t)$ and $s_{n2}(t)$ generated with the secret seed I_0.

One example of mixing matrix for encryption $\mathbf{A}_e$ satisfying (1) is

$$\mathbf{A}_e = \begin{bmatrix} -0.1098 & 0.6924 & -0.1098\times 10 & 0.6924\times 10 \\ 0.8636 & 0.0503 & 0.8636\times 10 & 0.0503\times 10 \end{bmatrix} \tag{8}$$

where $\beta = 10$, i.e., the mixing coefficients of the two key images $s_{n1}(t)$ and $s_{n2}(t)$ are considerably greater that those of the two original images $s_1(t)$ and $s_2(t)$. By using the encryption equation (2), two encrypted images $x_1(t)$ and $x_2(t)$, as shown in Fig. 2(c), are finally obtained, in which the two original images $s_1(t)$ and $s_2(t)$ have been well covered by the two key images $s_{n1}(t)$ and $s_{n2}(t)$ with much higher level of energy.

At the receiving side, after the two encrypted images $x_1(t)$ and $x_2(t)$ are available, and the two key images $s_{n1}(t)$ and $s_{n2}(t)$ are regenerated, both the regular BSS decryption and the fast BSS decryption are performed for comparison. Fig. 3(a) shows two decrypted images $\hat{s}_1(t)$ and $\hat{s}_2(t)$ by the regular BSS decryption, in which Comon's algorithm is used to separate the mixtures $\mathbf{x}_d(t) = [x_1(t), x_2(t), s_{n1}(t), s_{n2}(t)]^T$.

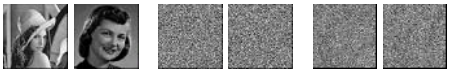
For the fast BSS decryption, the four mixing coefficients $\hat{a}_{13}$, $\hat{a}_{14}$, $\hat{a}_{23}$, and $\hat{a}_{24}$ in (5) are first estimated by using (7). They are –1.1000, 6.9198, 8.6357, and 0.5087, respectively. Compared with the real values shown in the last two columns of $\mathbf{A}_e$ in (8), these estimated values have only very small errors. By using (5), two new mixed signals $\hat{x}_1(t)$ and $\hat{x}_2(t)$ are then obtained, upon which the Comon's algorithm is performed to output two decrypted images $\hat{s}_1(t)$ and $\hat{s}_2(t)$, as shown in Fig. 3(b). From Fig. 3(a) and Fig. 3(b), it can be seen that the decrypted images by the two decryption methods are all of good quality. Owing to the characteristic of the human perception, the differences of the decrypted images and the original images are hard to identify, and the degradation in Fig. 3(b) is also hard to identify.

To quantify the performance of the proposed algorithm, we mainly analyzed two aspects of performance: signal-to-noise ratio (*SNR*) index and the average decryption times. Specifically, the *SNR* index for the original images $s_i(t)$, i=1,…,P, in the decrypted images is defined as follows:

$$SNR_i = 10\log \frac{\sum_{t=1}^{T} \left[(\mathbf{WA})_{ii}\, s_i(t)\right]^2}{\sum_{j=1}^{2P}\sum_{t=1}^{T} \left[(\mathbf{WA})_{ij}\, s_j(t)\right]^2}, \quad i \neq j \tag{9}$$

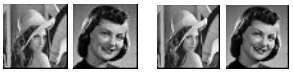
where $\mathbf{W}$ is the de-mixing matrix obtained from the BSS decryption algorithm used. $\mathbf{A}$ is the mixing matrix corresponding to the mixed signals for the BSS decryption. It is the same with (3) for the regular BSS decryption, but is $\mathbf{A} = [\mathbf{B} \quad \beta\mathbf{B} - \hat{\mathbf{F}}]$ for the fast BSS decryption, where $\hat{\mathbf{F}}$ is a $P \times P$ matrix formed with $\hat{a}_{i(P+j)}$, $i = j = 1,\ldots,P$, in (5). $(\mathbf{WA})_{ij}$ denotes the ith row and the jth column of matrix $\mathbf{WA}$. SNR_i was computed for the two original images $s_1(t)$ and $s_2(t)$ in the two decrypted images

$\hat{s}_1(t)$ and $\hat{s}_2(t)$ by the two decryption methods, respectively. Table 1 shows the results. We can find that the *SNR* indexes are all very high for both methods though *SNR*s of the fast BSS decryption are slightly smaller than those of the regular BSS decryption due to the small estimation errors of $\hat{a}_{i(P+j)}$.



(a) (b) (c)

Fig. 2. Example of encrypting two original images. (a) Two original images $s_1(t)$ and $s_2(t)$. (b) Two key images $s_{n1}(t)$ and $s_{n2}(t)$. (c) Two encrypted images $x_1(t)$ and $x_2(t)$.



(a) (b)

Fig. 3. Decrypted images by two decryption methods. (a) Two decrypted images $\hat{s}_1(t)$ and $\hat{s}_2(t)$ by regular BSS decryption. (b) Two decrypted images $\hat{s}_1(t)$ and $\hat{s}_2(t)$ by fast BSS decryption.

Besides, we compared the average times consumed by the two decryption methods. The running time of the BSS algorithm (e.g., Comon's algorithm) contributes most of the decryption time, while the time for estimating $\hat{a}_{i(P+j)}$ is also included for the fast BSS decryption. Specifically, the time consumed by the regular BSS decryption is 1.35s, whereas the time used by the fast BSS decryption is only 0.27s in this experiment. That is to say, the fast BSS decryption is roughly five times faster than the regular BSS decryption when encrypting two original images simultaneously.

To compare the running times for both regular and fast decryption methods as a function of the number of images encrypted, we also compute the average decryption times for both methods with different number of images (P=2,3,4,5) to be simultaneously encrypted. Table 2 shows the results. It can be seen that the decryption become slower with an increase of the images number P. But the fast BSS decryption is usually 2-3 times faster than the regular BSS decryption for different P. Therefore, the fast BSS decryption efficiently shortens the average decryption time by reducing the number of the source signals for the BSS algorithm in half.

Table 1. Comparison of *SNR*s (dB) of two original images $s_1(t)$ and $s_2(t)$ in two decrypted images $\hat{s}_1(t)$ and $\hat{s}_2(t)$ for two BSS decryption methods

	Regular BSS decryption	Fast BSS decryption
$s_1(t)$	72.32	71.46
$s_2(t)$	74.67	73.94

Table 2. Comparison of average times (second) for two BSS decryption methods with different number of images to be simultaneously encrypted

	Regular BSS decryption	Fast BSS decryption
P =2	1.35	0.27
P =3	3.34	0.91
P =4	4.85	1.71
P =5	6.48	2.93

5 Conclusion

The BSS-based method is more suitable for encrypting multiple images simultaneously since the multiple images can mask mutually besides the key images cover them. But the decryption time will become considerably longer with an increase of the images number. To achieve much higher level of security while retaining the rapid decryption speed, this paper presents a fast BSS decryption algorithm by using the knowledge of the key images, which are subtracted from the encrypted images based on adaptive noise cancellation. As a result, the number of the source signals for the BSS decryption is decreased in half, and the proposed method is as least twice times faster than the regular decryption.

The exact estimation of parameters $\hat{a}_{i(P+j)}$ is very important for the proposed method to have satisfied quality of the recovered images. In fact, the $\hat{a}_{i(P+j)}$ estimation errors are very small when the P original images $s_1(t), ..., s_P(t)$ are independent. Therefore, a practical solution to further decreasing the estimation errors is to permute the pixel values of each original image randomly before encryption to give more independent images for encryption, and then to recover the original images after decryption by restoring the original order of the pixel values for each decrypted image. In such a case, the proposed method can have both fast decryption speed and good image quality.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant No. 60402013 and No. 60372082.

References

1. Chang, C.C., Hwang, M.S., Chen, T.S.: A New Encryption Algorithm for Image Cryptosystems. The Journal of Systems and Software, 58 (2001) 83–91
2. Cheng, H., Li, X.B.: Partial Encryption of Compressed Images and Videos. IEEE Trans. Signal Processing, 48 (2000) 2439–2451
3. Hyvärinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. John Wiley (2001)

4. Cichocki, A., Amari, S.: Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications. John Wiley (2003)
5. Kasprzak, W., Cichocki, A.: Hidden Image Separation from Incomplete Image Mixtures by Independent Component Analysis. Proc. 13th Int. Conf. Pattern Recognition, II (1996) 394–398
6. Lin, Q.H., Yin, F.L.: Blind Source Separation Applied to Image Cryptosystems with Dual Encryption. Electronics Letters, 38 (2002) 1092–1094
7. Lin, Q.H., Yin, F.L., Liang, H.L.: Blind Source Separation-Based Encryption of Images and Speeches. J. Wang, X. Liao, and Z. Yi (Eds.): Advances in Neural Networks-ISNN 2005, Lecture Notes in Computer Science, 3497 (2005) 544–549
8. Menezes, A., Van Oorschot, P., Vanstone, S.: Handbook of Applied Cryptography. CRC Press Inc. (1996)
9. Cao, X.R., Liu, R.W.: General Approach to Blind Source Separation. IEEE Trans. Signal Processing, 44 (1996) 562–571
10. Haykin, S.: Adaptive Filter Theory. Englewood Cliffs, New Jersey: Prentice-Hall (1996)
11. Comon, P.: Independent Component Analysis, A New Concept?. Signal Processing, 36 (1994) 287–314

A Novel Cryptographic Scheme Based on Wavelet Neural Networks

Guo Chen[1], Feng Tan[2], and Degang Yang[3]

[1] Department of Modern Educational Technology, Chongqing Normal University, Chongqing 400047, China
[2] Department of Computer Science and Engineering, Chongqing University Chongqing 400044, China
[3] Department of Mathematics and Computer Science, Chongqing Normal University, Chongqing 400047, China
andychennet@163.com

Abstract. A novel method for encryption based on a wavelet neural network (WNN) is presented. The WNN is trained by a heuristic algorithm and can generate a random sequence which is used for encrypting and decrypting. Furthermore, some simulated experiments, including key space analysis, key sensitivity test, statistical analysis, are performed to substantiate that our scheme can make cipher-text more confusion and diffusion and that the method can resist several attacks, effectively.

1 Introduction

Cryptography based on algebra has been researched for many years and people have proposed many excellent cryptographic algorithms, such as International Data Encryption Algorithm (IDEA), Data Encryption Standard (DES), and Advanced Encryption Standard (AES) [1]. In recent years, people pay much attention to study the behavior of chaotic systems, whose highly unpredictable property and sensitivity to initial conditions and system parameters are attractive features that have been applied to many fields. Chaotic systems and cryptography have some similarities. Shannon once said [2], "in a good mixing transformation, functions are complicated and involving all the variables in a sensitive way and a small variation of anyone changes, the output changes considerably". Despite a number of papers published in the field of chaos-based cryptography in the past decade, the impact is rather marginal comparing with traditional cryptography. One of important factor is that these researches mainly concentrate on theory and these chaos-based cryptographic systems are low in efficiency because they are usually based on an iterated chaotic map which could only obtain an iterated value once a time. On the other hand, a neural network system is an information processing paradigm which is famous as a highly nonlinear dynamics system, a highly fast parallel system and is applied to many fields greatly [4], which enlightens us to use neural networks to generate an iterated sequence in a fast way after being trained and employ the sequence to our cipher scheme.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 326 – 331, 2006.

2 Chaotic Chebyshev Map

The Chebyshev map of degree k with I=[-1,1] is defined as

$$\tau(x) = \cos(k\cos^{-1}x) \tag{1}$$

The Chebyshev map is an ergodic chaotic map with many perfect properties, such as absolutely continuous invariant (ACI) measure, equi-distributivity, symmetric property. In this paper, we employ the following method to obtain a random binary sequence. Denote a floating value x in a binary representation as

$$x = 0.b_1(x)b_2(x)\ldots b_i(x)\ldots, \qquad x \in [0,1], b_i(x) \in \{0,1\}. \tag{2}$$

The ith bit $b_i(x)$ can be expressed as

$$b_i(x) = \sum_{r=1}^{2^i-1} (-1)^{r-1} \Theta_{(r/2^i)}(x) \tag{3}$$

Where $\Theta_t(x)$ is a threshold function which is defined as

$$\Theta_t(x) = \begin{cases} 0 & x < t \\ 1 & x \geq t \end{cases} \tag{4}$$

It can be proved that these random variables coming from $\left\{b_i(\tau^n(x))\right\}_{n=0}^{\infty}$ are independent and identically distributed (i.i.d), which is discussed in detail in [6].

3 Our Model Using a Wavelet Neural Network

3.1 The Wavelet Basis Neural Network

Wavelet basis neural networks are based on wavelet transform, in which the wavelet basis functions are activation functions of the neurons, instead of common nonlinear activation functions and they can be also seen as the development of radial basis function neural networks. In this paper, we adopt three-layer structure for our wavelet neural network, namely an output layer, a hidden layer and an input layer, which however do not affect the performance in approximating. In addition we choose sigmoid functions as activation functions of the output layer and wavelet basis functions as those of the hidden layer for the purpose that they have the stronger performance of approximating. Furthermore we use entropy function instead of energy function as error function for increasing the efficiency of convergence. The wavelet neural network model has m values of input and n values of output, which can be denoted as:

$$y_i(t) = \sigma\left[\sum_{j=0}^{n} W_{ij}\psi_{a,b}\left[\sum_{k=0}^{m} V_{jk}x_k(t)\right]\right] \quad (i = 1,2,\cdots n) \tag{5}$$

Where $\psi_{a,b}(t)$ is the wavelet basis function coming from the mother wavelet ψ(t) and a and b are dilation and translation scale factors, respectively. The x_k is the kth input value of the input layer, and y_i is the ith output value of the output layer. The W_{ij} is the weights of connection from output layer node i to hidden layer node j, and V_{jk} is the weights from hidden layer node j to input layer node k. The m is the number of the input layer node and n is the number of output layer node.

3.2 The Heuristic Algorithm Based on Improved B-P for Training

The standard B-P algorithm has some faults such as: encountering local minimum, slow convergence speed and converging instability. So in the paper, we propose a heuristic algorithm by introducing a momentum and a variable learning rate parameter as improved B-P algorithm, which can overcome these shortcomings to some extend. Moreover, we calculate the error (entropy) function after inputting a group of samples (each sample contains m values) and it can be defined as:

$$E = -\sum_{k=1}^{p}\sum_{i=1}^{n}[d_i^k \ln y_i^k + (1-d_i^k)\ln(1-y_i^k)] \tag{6}$$

where y_i^k is the ith truth output value of the kth sample and d_i^k is the ith expect output value of the kth sample. In addition we introduce a momentum parameter μ and alter the weights and scale coefficients of the wavelet neural network like this:

$$W_{ij}(t) = W_{ij}(t-1) - \eta\frac{\partial E}{\partial W_{ij}} + \mu\Delta W_{ij}(t-1) \tag{7}$$

$$V_{jk}(t) = V_{jk}(t-1) - \eta\frac{\partial E}{\partial V_{jk}} + \mu\Delta V_{jk}(t-1) \tag{8}$$

$$a_j(t) = a_j(t-1) - \eta\frac{\partial E}{\partial a_j} + \mu\Delta a_j(t-1) \tag{9}$$

$$b_j(t) = b_j(t-1) - \eta\frac{\partial E}{\partial b_j} + \mu\Delta b_j(t-1) \tag{10}$$

Where η is the variable learning rate parameter and the algorithm description of altering η is as follows.

(a) Assume an initialized learning rate parameter η_0.
(b) If after an iteration, the error function increases over a bound (usually 1%—5%) , cancel the current updates of weights and scale coefficients; the learning rate parameter multiplies a real number t (0<t<1) and at the same time the momentum parameter μ is set to zero.
(c) If the error function decreases, accept current updates; the learning rate parameter η multiplies t (t>1) and if μ was set to zero, recover it to the initial value.
(d) If the error function increases in the range of being accepted, accept the current updates and keep the learning rate parameter.

We use chaotic sequences to train the neural network. Firstly, generate a large number of chaotic sequences by the Chebyshev map and then divide these sequences into R groups. Each group has P samples. The training algorithm is as follows.

(a) Initiate V, W and let r=1, t=1.
(b) Input a sample.
(c) Calculate the output of each layer. r=r+1, If r<P go to (b).
(d) Calculate E and alter weights and scale coefficients, r=1,t=t+1, If t<R, go to (b).
(e) If $E>E_{min}$, t =1,go to (b) else stop.

When the training of the neural network has been accomplished, the real or pseudo Chebyshev chaotic sequence can be acquired from it in a fast speed, and we can use formula (3) to generate random sequences.

4 The Scheme of Encryption and Decryption

4.1 The Design of Algorithm

We adopt stream cipher system and the encrypting and decrypting scheme is illustrated in figure1.

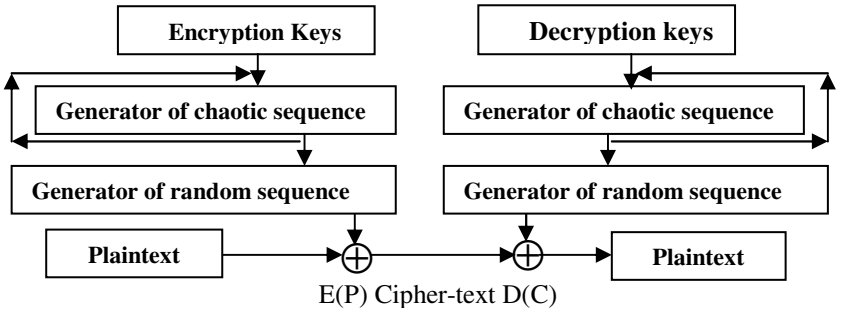


Fig. 1. Encryption and decryption scheme

Where the keys are the input initiate values of the neural network and the "Generator of chaotic sequence" is our wavelet neural network model and the "generator of random sequence" employs the way (3) to obtain random sequences.

4.2 Experiment of the Encryption System

In our experiment we choose

$$\psi(t) = \cos(1.75t)\exp(-t^2/2) \tag{11}$$

as the wavelet basis function. Choose x=0.400 as the initiate iterating value and generate m initiate input values. We test the algorithm by VC++ 6.0 and input an image plaintext. The result of encryption is graphically shown in figure 2, 3.

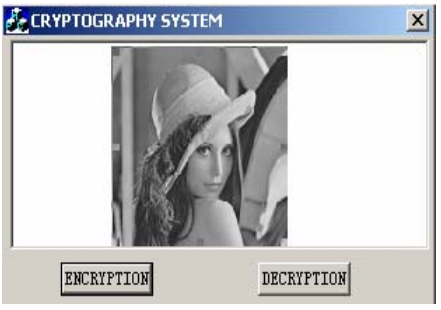


Fig. 2. Image plaintext

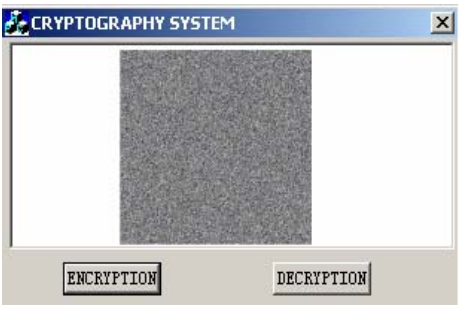


Fig. 3. Image cipher-text

4.3 Security Analysis

A good encryption scheme should resist several kinds of known attacks [1].Some security analyses have been performed on the proposed encryption scheme, including key space analysis, key sensitivity test, statistical analysis, cipher-text only attack, known-plaintext attack, and chosen-plaintext attack.

4.3.1 Key Space and Key Sensitivity Analysis

We can provide a large range of key set because the chaotic system is very sensitive to the initiate conditions and system parameters and a slight change of key, the sequence of iteration will become great different. In addition, the key distributing in the key space is uniform owing to the ergodic property of the chaotic system.

4.3.2 Statistical Analysis

Shannon once said [2], "It is possible to solve many kinds of ciphers by statistical analysis". So in this paper, the gray-scale distribution of the image for the plaintext and cipher-text are tested respectively. They are graphically shown in figure 4 and figure 5.

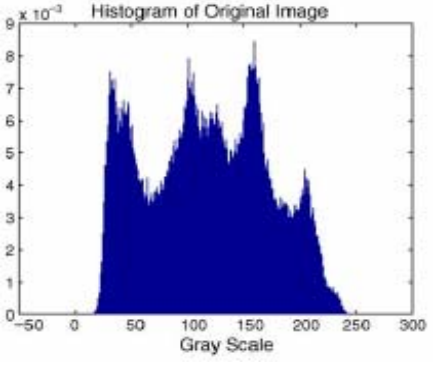


Fig. 4. Histogram of original Image

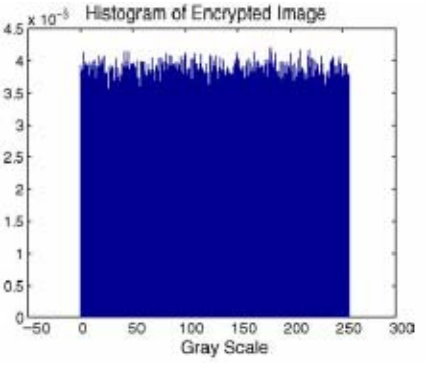


Fig. 5. Histogram of Encrypted Image

From above, it can be realized that the histogram of the ciphered image is fairly uniform and is significantly different from that of the original image. The analysis shows that our scheme can effectively resist the attacker using statistic analysis.

5 Conclusion

In this paper, a new cryptographic scheme based on wavelet neural networks is presented, in which we make use of the advantage of properties of chaotic systems and neural networks, respectively and overcome some shortcoming in chaos-based cryptography to some extend. In addition, the results of numerical analysis show that our method can resist manifold attacks effectively.

References

1. Shneier, B.: Applied Cryptography: Protocols Algorithms, and Source Code in C. John Wiley & Sons, Inc (1996)
2. Shannon, C.E.: Communication Theory of Secrecy Systems, Bell Systems Technical Journal. 28 (1949) 656-715.
3. Szu, H., Telfer, B., *et al*.: Neural Network Adaptive Wavelets for Signal Representation and Classification. Optical Engineering 36 (1992) 1907-1916.
4. Simon, H.: Neural Network a Comprehensive Foundation. Prentice Hall, Inc (1999).
5. Van Ooyen, A., Nienhuis, B.: Improving the Convergence of Back Propagation Algorithm. Neural Networks.5 (1992) 465- 471.
6. Kohda, T., Tsuneda, A.: Statistics of Chaotic Binary Sequences. IEEE Transactions on Information Theory 43 (1997) 104-112.

Combining RBF Neural Network and Chaotic Map to Construct Hash Function

Pengcheng Wei[1,2], Wei Zhang[1,2], Huaqian Yang[1,2], and Jun Chen[2]

[1] Department of Computer Science and Engineering, Chongqing University 400044, China
[2] Department of Computer and Modern Education Technology,
Chongqing Education College, Chongqing 400067, China

Abstract. How to design an efficient and security keyed hash function is always the point in modern cryptography researches. In this paper, A better chaos sequence is generated by RBF neural network through training the known chaotic sequence generated by a piecewise nonlinear map, then the sequence is used to construct keyed hash function. One advantage of the algorithm is that the hidden-mapping model of neural network makes it difficult to get the direct mapping function of the ordinary chaos hash algorithm. Simulation results show that the keyed hash function based on the neural network has good one-way, weak collision, better security property and it can be realized easily.

1 Introduction

Hash Function compresses a string of arbitrary length to a string of fixed length. In cryptography, hash functions are typically used for digital signatures to authenticate the message being sent so that the recipient can verify that the message is authentic and that it came from the right person. The requirements for a cryptographic hash function are [1]:

① Given a message m and a hash function H, it should be easy and fast to compute the hash $h = H(m)$.

② Given h, it is hard to compute m such that $h = H(m)$ (property of one way).

③ Given m, it is hard to find another message m' such that $H(m') = H(m)$ (property of collision free).

In recent years, there has been an increased interest in hash algorithm based on chaotic map [2-5], but under the finite precision in practical application conditions, the chaotic sequences have drawbacks of short period and strong correlation. Hence, it is necessary to research a new hash approach with strong security. In this paper, by investigating the nonlinear and parallel computing properties of neural network and integrating chaos with them, a hash scheme based on chaotic neural network is proposed. Firstly, A better chaotic sequence is generated by the neural network through training the known chaotic sequence generated by a piecewise nonlinear. Then the sequence is used to construct keyed hash function. The scheme can securely, efficiently complete hash in this combined mode. It is practicable and reliable, with high potential to be adopted for Ecommerce.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 332 – 339, 2006.

2 RBFNN Training the Known Chaotic Sequences

2.1 RBFNN

We consider multiple inputs single output system without loss of generality. A structure of RBF(Fig.1)comprises three layers. The hidden layer possesses an array of neurons, referred to as the computing nodes. Each hidden node in the network has two parameters called a center vector $C_i = [c_{i1}, c_{i2}, \cdots c_{in}]$ and a width σ_i associated with it. The activation function of the hidden nodes is Gaussian function in general. The output of each hidden node depends on the radial distance between the input vector $X = [x_1, x_2 \cdots x_n]^T$ and the center vector C_i. The response of the ith hidden node to the network input is expressed as:

$$\Phi_i(X_n) = \exp(-\|x_n - C_i\| / 2\sigma^2) \tag{1}$$

Where $\|\bullet\|$ denotes the Euclidean norm. The response of each hidden node is scaled by its connecting weights ϖ_i to the output node and then summed to produce the overall network output. The overall network output is expressed as:

$$y_n = f(X_n) = b_0 + \sum_{i=1}^{h} w_i \Phi_i(X_i) \tag{2}$$

Where h is the number of hidden nodes in the network b_0 is the bias term for the output node. The leaning process of RBF involves allocation of new hidden node as well as adaptation of network parameters. The network begins with no hidden node. As input-output (X_n, y_n) data are received during training, some of them are used to generate new hidden nodes.

2.2 Selection Sample District

A piecewise nonlinear map $f: I \to I, I_i = [0,1], I_i (i = 0,1,\cdots N)$ and denote the sub-interval of[0,1] as well as the length of this region(Fig.2), which can be presented [6,7]:

$$F(x_{k+1}) = \begin{cases} \left(\dfrac{1}{I_{i+1} - I_i} + a_i\right)(x_k - a_i) - \dfrac{a_i}{I_{i+1} - I_i}(x_k - a_i)^2 & \text{if} \quad x_k \in [I_i, I_{i+1}) \\ 0 & \text{if} \quad x_k = 0.5 \\ F(x_k - 0.5) & \text{if} \quad x_k \in (0.5,1) \end{cases} \tag{3}$$

Where $x_k \in [0,1]$, $0 = I_0 < I_1 < \cdots I_i \cdots I_{N+1} = 0.5$, $N \geq 2$, $a_j \in (-1,0) \bigcup (0,1)$, $j = 0,1,\cdots N$ and

$$\sum_{i=0}^{N-1} (I_{i+1} - I_i) a_i = 0 \tag{4}$$

We find such map F(.) has the following interesting properties:

i). The iterating system $x_{k+1} = F(x_k) \quad (k \geq 0)$ is chaotic;

ii). The sequence $\{x_k\}_{k=1}^{\infty}$ is ergodcity in [0,1] as well as with a uniform probability distribution function $f(x) = 1$.

iii). The sequence $\{x_k\}_{k=1}^{\infty}$ with a δ–like auto-correlation function:

$$R_F(r) = \lim_{j\to\infty} \frac{1}{J} \frac{\sum_{k=1}^{J} x_k x_{k+r}}{\sum_{k=1}^{J} x_k^2}, \quad r \geq 0 \tag{5}$$

Proof. i) For $x \in [I_i, I_{i+1})$ and $0 \leq i \leq N$,

$$\left|F^{'}(x)\right| = \left(\frac{1}{I_{i+1} - I_i} + a_i\right) - \frac{2a_i}{I_{i+1} - I_i}(x - I_i) > \frac{1}{I_{i+1} - I_i} - a_i > 1$$

By noting the F(x) is even-symmetric map, then $\left|F^{'}(x)\right| > 1$ holds for all $x \in [0,1]$, that is, the Lyapunov exponent:

$$\lambda = \lim_{J\to\infty} \frac{1}{J} \log_2 \left(\prod_{k=1}^{J} \left|F^{'}(x_k)\right| \right) > 0 \tag{6}$$

According to the definition of chaotic system [HBL], a positive Lyapunov exponent means the iterating system is chaotic, so property 1) holds.

ii) As the same as (i), when $x \in [I_i, I_{i+1})$ and $0 \leq i \leq N$

$$\begin{aligned} \left|F^{''}(x)\right| / \left|F^{'}(x)\right|^2 &= 2a_i (I_{i+1} - I_i) / [(1 + a_i (I_{i+1} - I_i)) - 2a_i (x - I_i)]^2 \\ &< 2a_i (I_{i+1} - I_i) / (1 - (I_{i+!} - I_i) a_i)^2 \\ &< 2 / (1 - (I_{i+1} - I_i) a_i)^2 \end{aligned} \tag{7}$$

Note that F(x) is even-symmetric map, we can obtain $\left|F^{''}(x)\right| / \left|F^{'}(x)\right|^2 < +\infty$ for all $x \in [0,1]$. According to above theory, $P_r \circ f(x) = \sum \left| f(y_i) \right| (I_{i+1} - I_i)(1 + a_i x), \quad x \in [0,1]$

where *Pr* is the Frobenius-Perron operator of map f(x) defined as [kohda]

$$P_r \circ f(x) = \frac{d}{dy} \int_{F^{-1}([0,x])} f(x) dx \tag{8}$$

where $y_i = I_i + (I_{i+1} - I_i)(1 + a_i I_{i+1} + a_i I_i - 2a_i x)$.

According to the Eq.(3), f(x)=1 is the unique solution of $f(x) = P_r \circ f(x)$, so property 2) holds.

iii) By the same ensemble average technique [kohda], we rewrite the auto-correlation function as

$$\rho_F(r) = \frac{\int_0^1 xF^r(x)f(x)dx}{\int_0^1 x^2 f(x)dx} \tag{9}$$

When $r=0$, $Fr(x)=x$, $\rho_{F(r)}=1$; when r>0, x is a odd and $Fr(x)=x$ is even with respect to the center $x=0.5$, then $\rho_F(r)=0$, so property ii) holds.

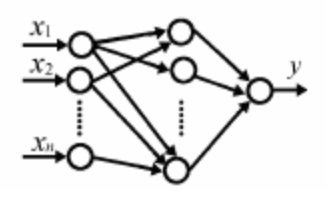


Fig. 1. RBFNN

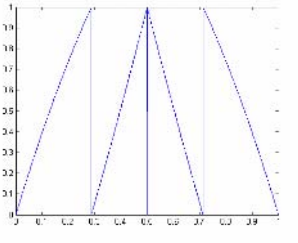


Fig. 2. Piecewise nonlinear map

2.3 The Process of RBFNN Training

The structure of RBFNN training the known chaotic sequence generated by the piecewise nonlinear is shown in Fig. 3(CS-RBFNN). In order to predict $x(n+1)$ by time serial $x(1), x(2)\cdots x(n)$, we take $x(n-k+1),\cdots x(n)$ as input of RBFNN, and the output of RBFNN is $\hat{x}(n+1)$ (predictable value). Fig. 3 is the process of RBFNN training.

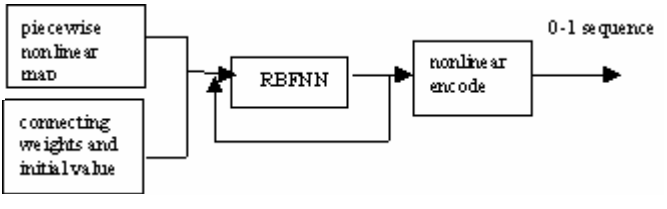


Fig. 3. The structure of CS-RBFNN

The RBFNN initial input is $x(1), x(2)\cdots x(n)$ which generated by the piecewise nonlinear, the chaotic sequences $\hat{x}(n+1), \hat{x}(n+2), \cdots \hat{x}(n+m)$ is the RBFNN output by the RBFNN training, So we can induce how to predict $\hat{x}(n+k)$ using $x(1), x(2)\cdots x(n)$, where $k \geq 1$. In this paper, we adopt recursion to solve the above problem,

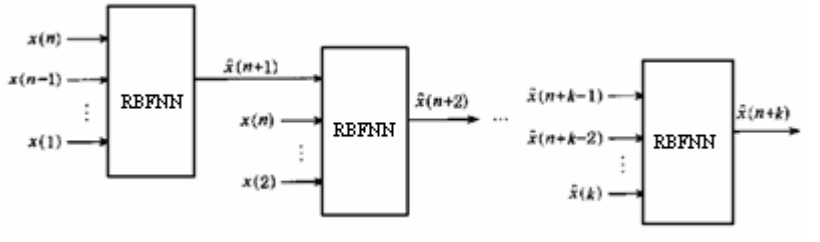


Fig. 4. The relation between input and output of RBFNN

each time predict one step only, that is $\hat{x}(n+k)$, k=1, then the feedback of $\hat{x}(n+k)$ will predict next step. By this means we can obtain all chaotic sequences, Fig. 4 is the relation between input and output of RBFNN.

3 Realization of Hash Function

Combining RBFNN and chaotic maps to construction hash function is show in Fig. 5, it conclude three steps.

(a) Suppose message M is binary sequences, and the length of hash value is N $(N = 128 * i, i = 1,2,\cdots)$, $|M| = kN, k = 1,2,3\cdots$, if $|\mathrm{M}|$ isn't satisfied with $|M| = kN, k = 1,2,3\cdots$, it will be connected some random sequences to content the above equation. Then divide message M into N bits separate portion, that is $M = M_1, M_2, \cdots, M_k$, denoted $M_i = M_i^1 M_i^2 \cdots M_i^N$.

(b) Select random sequences N as the initial vector H_0, and compute $H_0 \oplus M_1$.

(c) Taken the initial value $x(0)$ of the piecewise nonlinear map and RBFNN parameter as hash system private key, CS-RBFF train chaotic sequence and obtain N bits random sequence. Using N bits random sequences encrypt $H_0 \oplus M_1$, the N bits ciphertext H_1 is hash value of message M_1, compute $H_1 \oplus M_2$, encrypt $H_1 \oplus M_2$ by CS-RBFNN, the N bits ciphertext H_2 is hash value of message $\mathrm{M}_1\mathrm{M}_2$. Repeat the process until all message being encrypt, finally H_k is hash value of message M.

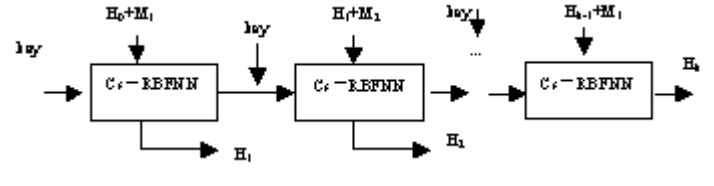


Fig. 5. Hash algorithm structure

4 Performance Analysis

4.1 One-Way

One-way property means that it only allows a digest to be created from the original, yet the inverse is very hard [8]. In order to describe this characteristic explicitly, we compress the below plaintext by our scheme.

Plaintext

Cryptographist the science of overt secret writing (cryptography), of its authorized decryption (cryptanalysis), and of the rules which are in turn intended to make that unauthorized decryption more difficult (encryption security).

For distinguishing plaintext from hash valve, we employ two dimensions graph express the result, in Fig.6 the ASCII code of plaintext distribute over small zone, after compressed by our scheme, in Fig .7 the HEX code of hash value decentralize among large zone uniformly. So our scheme has good one-way property.

4.2 Collision Resistance

Collision resistance is the difficulty of finding two different inputs hashing to the same [10]. Each neuron in a layer is connected to all in the next layer, this inherent structure expedites the avalanche effect; furthermore, piecewise nonlinear map is sensitivity to initial value, these ensure that even a single bit change in input will definitely result in great changes in the output.

4.3 Auto-correlation and Cross-Correlation Function with Least Different Keys

Repeated plaintext such as "QQ……" is digested and the corresponding auto-correlation function is analyzed. It is a δ function and has no repeated period. In Fig.8, for the sensitivity of chaos, least different keys will result in the great changes in the output of RBFNN, which will totally change the hash value.

4.4 Efficiency

The length of hash value is changeable, so system needn't be restarted for the fixed problem [9], which greatly reduces communication and computation load, furthermore,

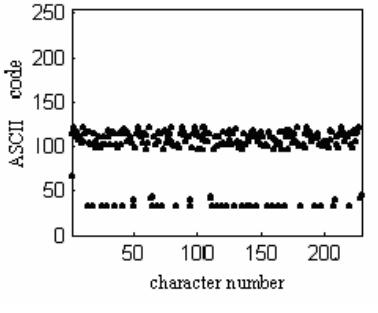


Fig. 6. ASCII distribution of plaintext

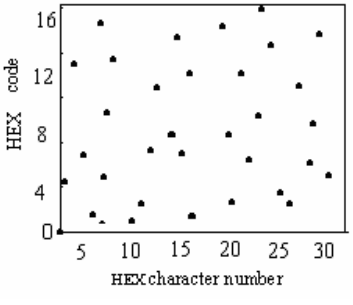


Fig. 7. HEX distribution of hash value

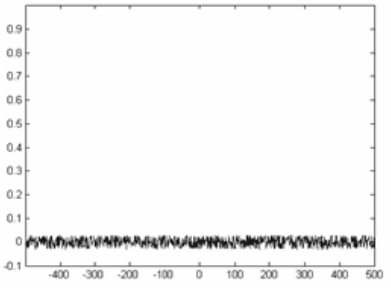


Fig. 8. Cross-correlation function

our scheme is not involved in any time-consuming modular exponential computing so the efficiency of the whole scheme is boosted much more.

5 Conclusions

In this paper, we investigate the nonlinear and parallel computing properties of neural network and integrate chaos with them, propose a hash scheme by RBFNN and piecewise nonlinear map, the scheme can securely and efficiently complete hash. It is practicable, flexible and reliable with high potential to be adopted for E-commerce and E-government.

Acknowledgement

The work described here was supported by National Science Foundation of China (Project No. 60573047), by National Science Foundation of Chongqing Science and Technology Committee (Project No. CSTC, 2005B2286), by the Applying Basic Research of Chongqing Education Committee (No.kj051501).

References

1. Schneier, B.: Applied Cryptography. 3rd edn. China Machine Press, Beijing (2001)
2. Pieprzyh, J.: Design of Hashing Algorithm. Springer-Verlag, Berlin Heidelberg New York (1993)
3. Xiaomin, W., Jianhu, Zh., Wenfang, Zh.: One Way Hash Function Construction Based on the Extended Chaotic Maps Switch. Acta Physica Sinica 52(6) (2003) 273-275
4. Junning, L., Jiecheng, X., Pu, W.: One Way Hash Function Construction Based on Chaotic Mappings. J Tsinghua Univ. 40(3) (2000) 55-58
5. Hongda, L., Dengguo, F.: Composite Nonlinear Discrete Chaotic Dynamical Systems and Keyed Hash Functions. Chinese Journal of Computers 26(5) (2003) 460-465
6. Tao, S., Ruli, W., Yixun, Y.: Generating Binary Bernoulli Sequences Based on a Class of Even-Symmetric Chaotic Maps. IEEE Tran. Communications 49(8) (2001) 620-623
7. Guojie, H., Zhengjin, F.: Theoretical Design for a Class of New Chaotic Stream Cipher. Communications Technology 24(4) (2003) 73-74

8. Di, X., Xiaofeng, L.: A Combined Hash and Encryption Scheme by Chaotic Neural Network. Vol. 3174. Springer-Verlag, Berlin Heidelberg New York (2004) 633–638
9. Di, X., Xiaofeng, L., Guoping, T., Chuandong, L.: Using Chebyshev Chaotic Map to Construct Infinite Length Hash Chains. 2004 International Conference on Communications Circuits and Systems (2004) 11-12
10. Kwok-Wo, W.: A Combined Chaotic Cryptographic and Hashing Scheme. Physics Letters A 307(10) (2003) 292–298

Multiple-Point Bit Mutation Method of Detector Generation for SNSD Model

Ying Tan

Department of Computer Science and Technology,
University of Science and Technology of China, Hefei 230027, P.R. China
ytan@ustc.edu.cn

Abstract. In self and non-self discrimination (SNSD) model, it is very important to generate a desirable detector set since it decides the performance and scale of the SNSD model based task. By using the famous principle of negative selection in natural immune system, a novel generating algorithm of detector, multiple-point bit mutation method, is proposed in this paper. It utilizes random multiple-point mutation to look for non-self detectors in a large range in the whole space of detectors, such that we can obtain a required detector set in a reasonable computation time. This paper describes the work procedure of the proposed detector generating algorithm. We tested the algorithm by using many datasets and compared it with the Exhaustive Detector Generating Algorithm in details. The experimental results show that the proposed algorithm outperforms the Exhaustive Detector Generating Algorithm both in computational complexities and detection performance.

1 Introduction

Biological immune system can protect the body against harmful diseases and infections by recognizing and discriminating non-self from self, which ability plays very important adjustment functions in the whole lifetime of biological creatures. Recently, many algorithms and architectures are proposed [1]-[5] in many engineering fields based on natural immune principles. The prevalent model of the immune system is Self and Non Self discrimination (SNSD) which has been used for the last 50 years in immune system community. This SNSD model uses the idea that the immune system functions by making a distinction between bodies native to the system, and foreign bodies. Many computer researchers use it as a metaphor to develop a number of artificial immune algorithm and systems for intrusion motoring and detection, resource change detection, pattern recognition as well as computation of complex problems. Among them, the negative selection algorithm (NSA) was proposed in [2] as a detector generating algorithm by simulating the negative selection process of T cells generating in thymus. However, the generation of detector is a vital computation burden in NSA and attracts a lot of researchers to explore an efficient and useful method or algorithm for it. In this research environment, this paper studies the detector-generating algorithm and proposes a detector generating algorithms based on SNSD model and negative selection principles from the perspective of optimizing detector generation. The procedure of the proposed algorithm is presented in details and thoroughly compared with the famous Exhaustive Detector Generating Algorithm (EDGA).

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 340 – 345, 2006.

2 Current Algorithms of Detector Generation

As we know that, for biological immune system, T cells with essentially random receptors are generated in thymus and their generation and maturation undergo a negative selection process. Before they are released to the rest of body, those T-cells that match self are deleted. This process is called as negative selection process. According to this principle, a number of detector generating algorithms are created in artificial immune systems (AIS) and practical applications. For example, in 1994, Forrest, et al., proposed so-called negative selection algorithm (NSA) inspired by this negative selection principle [2] and applied it to intrusion detection and change detection. NSA primarily consists of two stages of *censoring and monitoring*. The censoring phase caters for the generation of change-detectors. Subsequently, the system being protected is monitored for changes using the detector set generated in the censoring stage. In the censoring phase, the generation of change detectors needs a large of computational amount which limits the use of NSA in practical application. Since then, a number of detector-generating algorithms with different matching rules applied for 'self' and 'non-self' matching methods are proposed and tested for promoting the application of NSA. For binary code of the detector, there are mainly three kinds of matching rules: perfect matching, r-contiguous bits matching and Hamming distance matching [2]. Exhaustive detector generating algorithm (EDGA) is suitable for all of the three matching rules and is used to repeat the negative selection process till the number of detectors meets a presetting requirement or candidate set is vanished. However, negative selection algorithm with mutation [5] has different evaluating rules in the negative selection process from EDGA. With r-contiguous bits matching rule, the related algorithms mainly include Liner Time Detector Generating Algorithm [4] and Greedy Detectors Generating Algorithm [5].

3 Growth Algorithms

The major difference between growth algorithm and NSA is the generating method of candidate detectors. For NSA, each detector candidate in EDGA is randomly selected from whole candidate space. Thus, after the number of detector candidate space N_{r0} is determined, EDGA randomly selects N_{r0} detectors to construct the detector candidate set R_0. Then, the algorithm generates detector set R through negative selection process. On the other hand, growth algorithm does not need to maintain a huge detector candidate set R_0. It directly generates the detector set R by utilizing detector mutation or growth in whole shape space and combining with negative selection process. Its flow chart is shown in Fig.1.

Growth Algorithm

Step 1. Generate self set S with its number N_s.
Step 2. Generate one detector generating seed which is randomly selected from the whole shape space.
Step 3. Match the new detector candidate with S.
Step 4. Experience a negative selection process. If the candidate is not matched with S, then a new one is generated and added into R.
Step 5. If the stop criterion is met, then exit.
Step 6. Mutate the candidate and going to step3.

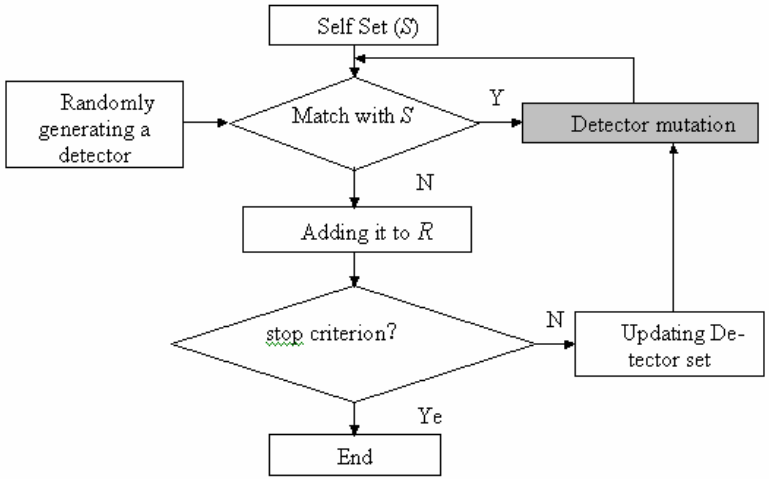


Fig. 1. Flow chart of detector generation growth algorithm

3 Multiple Point Bit Mutation Method

As the name indicates, the proposed method is somewhat similar to the bit mutation mechanism of Genetic Algorithm (GA). But this algorithm mutates multiple bits of detector candidate simultaneously, not only one bit. If we let the string length be l and the maximum number of mutation bits be N_m ($N_m = m$), then, at one time, the mutated bits of detector candidate is less than or equal to m.

Multiple-Point Bit Mutation Algorithm

Step 1. Set the maximum number of mutation bits N_m.
Step 2. Input detector DetectorM to be mutated.

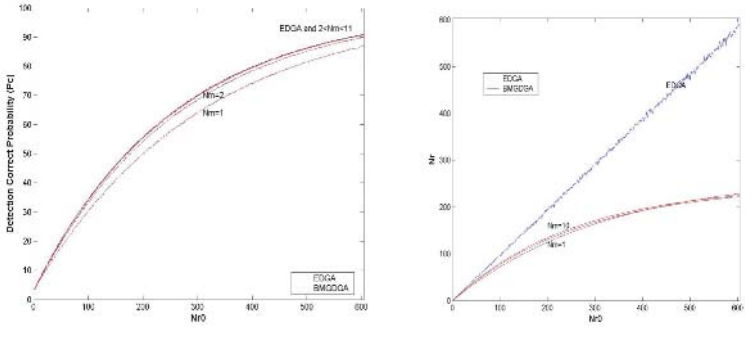


Fig. 2. Experimental results of Detection rate P_c versus N_{r0} and when N_s is constant, where N_s =8, N_{r0} increases from 1 to 604, size of test set is 256, and 1000 runs. And also curves of N_r versus N_{r0}.

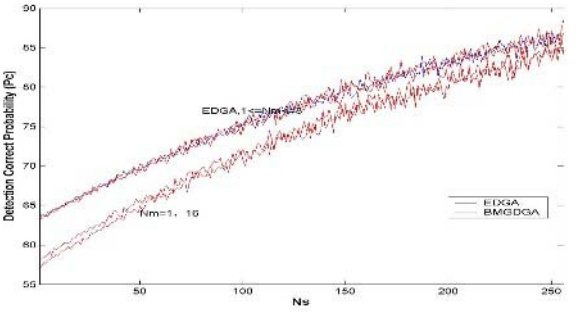


Fig. 3. Experimental result comparisons of EDGA to BMGDGA when Nr0 =256 and Ns changing from 1 to 256

Step 3. Generate mutation seed MutationBits (its length is same as detector): N_m bits of MutationBits are randomly generated at range of 1 and l and set to 1, and others are set to zero.

Step 4. Detector mutation: an exclusive OR operation performed on the corresponding bits of arrays DetectorM and MutationBits.

Once the proposed multiple-point bit mutation methods is applied to the detector mutation part of the growth algorithm in Fig.1, we can obtain the detector generating algorithm just called multiple-point bit mutation detector generation algorithm (BMDGA, for short).

4 Experiments

We set two kinds of experiments. One chooses random dataset with string length 8, 16 and 24 bits, respectively. Another experiment is to detect the changes of static files.

4.1 Random Dataset Experiments

4.1.1 8-Bit Dataset Experiments

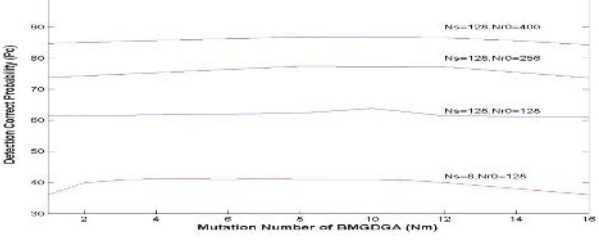


Fig. 4. Experimental results of detection rate P_c versus mutation number N_m, where N_s, N_{r0} are constants

4.2.2 16-Bit and 24-Bit Dataset Experiments

Table 1. Computational complexity comparison for 16-bit dataset

Algorithm	Parameters setting	Computational time (ms)	Note
EDGA	N_{r0}=65536, N_S=2048	730	
BMGDGA	N_{r0}=65536, N_S=2048 N_m=8	594.5	10 runs
	N_{r0}=65536, N_S=2048 N_m=4	585	10 runs

Table 2. Computational complexity comparison for 24-bit dataset

Algorithm	Parameters setting		Detection rate	Computational time (ms)
EDGA	N_S=2048, N_{r0}=16384 N_t=16384		0.1081542968	183.7
BMGDGA	N_S=2048	N_m=1	0.1094360351	128.11
	N_{r0}=16384	N_m=16	0.1115112304	151.86
	N_t=16384	N_m=24	0.1052246093	188.44

Notice that our experimental platform is Intel Pentium 993M CPU, 256M memory and Windows Me.According to above experimental results, we can notice that the detection rate P_c of BMGDGA is not lower than EDGA's under all circumstances, and when N_m=l/2, the detection rate reaches the best. Its computational complexity is also superior to EDGA's, with almost same memory space. So the overall performance of proposed BMGDGA is better than EDGA.

4.2 Change Detection of Static Files

We conduct two experiments with the algorithm to validate their detection abilities for the change of static files, and compare to EDGA. First of all, we compare two different files by using the algorithm. We select two files of 'FTP.exe' and 'Ping.exe' and define 'Ping.exe' as self (S). The experimental results of anomaly number are listed in the first sub-column of Anomaly number column of Table 3. Secondly, it is used to detect anomaly detection of the file infected by a virus, where we define a benign file "dbeng50.exe" as protected file and detect its changing when infected by "Worm Concept" virus. The experimental results of anomaly number are listed in the second sub-column of Anomaly number column of Table 3. Experimental results are all compared to EDGA and listed in Table 3 for convenience.

It can be seen from the experimental results that we can obtain the same results as random dataset experiments. To BMGDGA, when N_m=1, the quality of detector set generated by this algorithm is worst. When N_m=l/2~ l, the performance of BMGDGA

Table 3. Results of anomaly numbers for file comparison (first sub-column of last column), and anomaly detection (second sub-column of last column)

Algorithms	Parameters setting		Anomaly number	
EDGA	Detector length l=16bits		3775.2	596.9
BMGDGA	Detector length l=16bits	N_m=1	3389.5	537
		N_m=10	3724.8	561.7
		N_m=16	3709.6	606

is almost same to EDGA. It turns out from our experiments that it is possible to detect the changes of static files with these two algorithms.

5 Conclusions

This paper proposes a novel detector-generating algorithm on the basis of negative selection principle of natural immune system. Extensive experimental results show that the proposed algorithm outperforms EDGA in both detection performance and computational complexity.

References

1. D'haeseleer, P.: An Immunological Approach to Change Detection: Theoretical Results. In: Proceedings of the 9th IEEE Computer Security Foundations Workshop, IEEE Computer Society Press (1996)
2. Forrest, S., Perelson, A.S., Allen, L., Cherukuri, R.: Self-Nonself Discrimination in a Computer. In: Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy, Los Alamitos, CA (1994).
3. Kim, J., Bentley, P.: Immune Memory in the Dynamic Clonal Selection Algorithm. In Proceedings of ICARIS'02 (2002).
4. Ayara, M., Timmis, J., de Lemos, R., de Castro, L., Duncan, R.: Negative Selection: How to Generate Detectors. In: Proceedings of ICARIS'02 (2002).
5. Singh, S.: Anomaly Detection Using Negative Selection Based on the r-contiguous Matching Rule. In: Proceedings of ICARIS'02 (2002).

An Erotic Image Recognition Algorithm Based on Trunk Model and SVM Classification

Qindong Sun[1,2], Xinbo Huang[3,*], Xiaohong Guan[2], and Peng Gao[2]

[1] School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, Shaanxi 710048, China
sqd@xanet.edu.cn
[2] School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China
[3] School of Electro-Mechanical Engineering, Xi'an University of Engineering Science & Technology, Xi'an, Shaanxi 710048, China

Abstract. The characteristics of erotic images are analyzed in this paper and a novel algorithm for erotic images recognition is proposed. The algorithm first obtains the mask images of the recognized image by skin color detecting and texture analyzing, and then locates the possible position of human trunk in mask image according to the established model of trunk, based on which the characteristics of erotic images are extracted. Furthermore, the SVM classifier is used to recognize the erotic images based on those extracted characteristics. The experimental results show that the recognition accuracy rate of the proposed algorithm is higher than other algorithms and the proposed algorithm is efficient and effective.

1 Introduction

Increasingly network eroticism affects people's normal life seriously, especially to the growth of teenagers, and brings lots of society and moral problem. How to recognize and eliminate network eroticism has been one of hot problems in network information security field and the content-based erotic image recognition has received intensive attention.

Some excellent works were carried out on recognition of erotic images [1][2][3]. Fleck makes use of the computer vision and image recognition technique to recognize erotic images, which made the beneficial exploring in erotic image recognition, but the approach is of lower recognition accuracy and higher time complexity [1]. Duan adopts the proportion of skin color pixels in images as criterion, blending various methods of classification to recognize erotic images, which is simple, but cannot handle complicated situation [2]. Fan makes use of the characteristics of the skin color mask of images to recognize erotic images, which cannot distinguish whether the mask is limb or trunk clearly and could lead to misclassification [3].

A novel algorithm is proposed to recognize erotic images in this paper. By establishing the model of human trunk, the characteristics of erotic images are extracted and support vector machine classifier is used to recognize them.

* Corresponding author.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 346 – 351, 2006.

2 Characteristic Analysis of Erotic Images

The recognition of erotic images belongs to the scope of the content-based image filtering, but also has the special feature. Different from other image contents recognition problems, such as face detection [4], fingerprint recognition [5] etc., the background of erotic images is more complex.

According to statistics, the area of naked skin in erotic images will usually exceed 10% of whole image area. So it is necessary to make out whether there is large quantity naked skin region in images, which is the basis of recognition.

The skin color is a very valid feature for naked skin detection, which is a kind of important vision information attribute. The color feature stabilizes very much, not sensitive to the rotation, moving, dimensions variety, even pose changing. Furthermore, the computing of the color feature is simple. We can quickly expel other color background by making use of the skin color information, zooming out the searching scope.

Moreover, by texture analyzing, the region whose color is close with skin color can be removed effectively, and identify the skin color region more accurately. The simple color space YCrCb, which does not need complex computing [6], is used for skin color detection, and IRgBy model used for texture analysis in this paper.

Through skin color detection and texture analysis, the mask image is obtained. How to extract the characteristics of mask images is the key problem of erotic images recognition.

3 The Model of Human Trunk

The skin color detection can recognize the skin region in the image, but that is not enough for erotic images recognition. Because the marked characteristic of erotic images is naked trunk, but usually the face and limb of human is naked, taking them for human trunk will cause false identification. So how to locate the trunk and make sure whether it is naked is the basis of erotic images recognition.

Because the pose of human trunk in the image is usually unknown, we wish to locate its possible position based on a model. According to the physiological feature of human trunk, the trunk can be predigested as two fold lines whose meeting point is near the center of waist. As shown in Fig. 1, O is the center of human waist, and the fold lines, OC and OD, can represent the approximate position and orientation of the trunk.

In erotic images, the region of naked trunk usually occupies the most part of naked skin area, so we can suppose that the center of waist should be near the center of whole naked area. If the naked trunk exists in the image, we can consider it is near the center of waist.

Assuming the height and width of the image are x and y, and $x \geq y$; the number of skin color connected regions is i, and indexed by $1,2,\cdots i$; the length and width of the tangent rectangle of connected regions are $m_1, m_2, \cdots, m_i$ and $n_1, n_2, \cdots, n_i$ respectively.

Fig. 1. The simplified representation of trunk

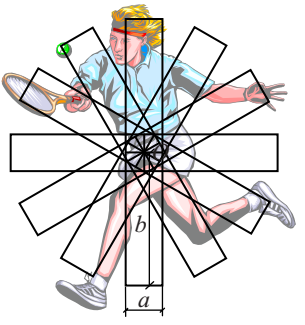


Fig. 2. The model of human trunk

Firstly, we get the center O of skin color region, and then create $k \times 2$ rectangles whose width and length are a and b by rotating around the point O, usually $k \geq 3$, as shown in Fig. 2. Here a and b are defined as follows,

$b = \lambda \max(\max(m_1, m_2, \cdots, m_i), \max(n_1, n_2, \cdots, n_i))$, and $a = b / \eta$

Where $\lambda \geq 1$ and η is an integer, usually $\eta \geq 3$. The value of λ and η could be determined during experiments with experiences. After the above preprocessing, the algorithm of locating trunk is given as follow,

Setp1: Two adjacent rectangles are regarded as one group, and $Rt_1, Rt_2, \cdots, Rt_{2k}$ are divided into k groups, i.e. $Gp_i = \{Rt_{2i-1}, Rt_{2i}\}, \quad 1 \leq i \leq k$

Setp2: Calculate the proportion of skin color pixels in each group $Gp_1, Gp_2, \cdots, Gp_k$, and get the proportion sequence $r_1, r_2, \cdots, r_k$;

Step3: Suppose that r_i and r_j are the two largest value in the sequence, and Gp_i and Gp_j are not adjacent, i.e. $|i-j| \neq 1$; if $|i-j| = 1$, then select the thirdly larger value to instead the little of r_i and r_j, until the condition is satisfied.

Step4: If the r_i and r_j are greater than the threshold M, we can consider that the naked trunk exists in the image, and the image is most likely to be erotic, and more processing is needed.

By using the established model, we can estimate preliminary that whether the naked trunk exists in images or not. If the naked trunk exists, further work is necessary to extract more complex classification characteristics. Then the following characteristics are extracted,

(1) The proportion of skin color pixels in whole image, ra; (2) The proportion of skin color pixels in Gp_i and Gp_j, rb; the proportion of skin color pixels in Gp_i and Gp_j to that in the whole image, rc. (3) The proportion of skin color pixel in the circular region with the center of o and radius of b, rd. (4) The proportion of the largest length and width of connected regions to that of the image, re and rf,

$$re = \frac{\max(\max(m_1, m_2, \cdots, m_i), \max(n_1, n_2, \cdots, n_i))}{x} \tag{1}$$

$$rf = \frac{\min(\max(m_1, m_2, \cdots, m_i), \max(n_1, n_2, \cdots, n_i))}{y} \tag{2}$$

(5) The angle φ between Gp_i and Gp_j, which is also an important classification attributes, because there is an obvious difference about it between the erotic images and normal images.

Then the recognition of erotic images is converted to a problem of classifying seven-dimension vector $R = [ra, rb, rc, rd, re, rf, \varphi]^{\mathrm{T}}$.

4 Recognition of Erotic Images Based on SVM

4.1 Support Vector Machine

Support Vector Machine is a kind of statistical learning theory, which is based on structural risk minimization, proposed by Vapnik [7]. If the classification problem is nonlinear separable, the data set will be mapped to a higher dimension space with nonlinear mapping method (kernel function), and the date set will turn to linear separable. The problem of classifying seven-dimension vector R is nonlinear. The nonlinear SVM can be expressed as,

$$f(R) = \sum \alpha_i y_i K(R_i, R) - b \tag{3}$$

where, $f(R)$ is the output of SVM, α_i is weight, R_i is training sample, R is the classified samples, $y_i \in \{-1,1\}$ is the target value for each training sample, $K(R_i, R)$ is kernel function, and b is error factor.

With the method of nonlinear mapping, the classification problem can be transformed to a quadratic programming problem in higher dimension space as following,

$$\begin{aligned} &\min_{\alpha} \ W(\alpha) = \min_{\alpha} \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(R_i, R_j) - \sum_{i=1}^{n} \alpha_i \\ &s.t. \ 0 \le \alpha_i \le C, i = 1,2,\cdots,n \quad and \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned} \tag{4}$$

where, C is a constant, greater than zero served as penalty factor. The classification results are decided with the value of $f(R)$ in formula (3).

4.2 Training of SVM Classifier

The computation complexity and memory requirement of training SVM are huge and some algorithms are proposed to solve the training problem of SVM, such as decomposition algorithm [8], sequential minimal optimization (SMO) algorithm [9] etc. In this paper, we will apply SMO to train SVM.

There are two important parameters of SVM based on radial basis kernel function, the radial basis function parameter δ, and the penalty factor C in formula (4). δ denotes the effect area of radial basis function. Punishment factor C reflects the penalty strength to those wrong classified results. Currently, the kernel function and relevant parameters only can be selected with experiences, without the guide of theory. In this paper, we will provide a group of testing samples, and decide the value of parameters according to analyzing the rule of the changing from the parameters and the practical data.

For the sake of a good classification effect, we need a set of samples not only enough in number but apt in distribution. It is easier to get the positive images, i.e. erotic images. Conversely, the representative negative samples are in various forms and difficult to be defined. If we choose samples randomly, the amounts of positive and negative samples may be badly unbalanced, which will make the derived SVM be too complex to use.

This paper adopts a method that gradually appends negative samples, which are classified wrongly, to resolve the above problem. By training repeatedly, two groups of samples are confirmed: one group is erotic images, the quantity is 2961; the other group is normal images, the quantity is 3573, where, 2132 images contain normal human body, 875 images contain animal, others contain scenery. All images are multicolor.

5 Experimental Results and Analysis

The experimental results are given in this section. In order to compare the performance of different algorithms, a comparison experiment, the method based on threshold, is implemented on the same sample set.

Through testing repeatedly, the parameters of SVM are confirmed, $\delta = 2.1$ and $C = 120$. Select 295 erotic images and 213 normal images as samples to be recognized. The accurate recognition rate (ARR), false negative rate (FNR), false positive rate (FPR) and classification error rate (CER) of different methods are shown in table 1.

Table 1. The performance of two methods

	ARR (%)	FNR(%)	FPR (%)	CER (%)
Threshold	71.18	28.82	23.36	29.33
SVM	90.17	9.83	7.32	10.63

According to the analysis of the above results, we can conclude that:

(1) The classification vector extracted by the model of trunk preferably reflects the difference between erotic images and normal images. The performance of the methods based on the proposed trunk model is better than that of the method based on threshold.

(2) The model of trunk is very effective, which help to extract the characteristics of erotic images greatly and all of the processing for the algorithm can be finished in once scan of original image, which does not increase too much payload of system. The shortage of the trunk model lies on the confirmation of the center of trunk, which

is effectively, but may lead to the wrong selection of trunk sometime, especially in some most complex images.

6 Conclusions

The characteristics of erotic images are analyzed in this paper and a novel algorithm for erotic images recognition is proposed. The algorithm first obtains the mask images of the image, and then locates the possible position of human trunk in mask image according to the established trunk model, based on which the characteristics of erotic images are extracted. Furthermore, the SVM classifier is used to recognize the erotic images. Experiments are implemented to manifest the performance of the proposed algorithm and experimental results show that the recognition accuracy rate of the proposed algorithm is higher than that of the other algorithm and the proposed algorithm is effective.

References

1. Fleck, M.M., Forsyth, D.A., Bregler C.: Finding Naked People. Proceedings of 4th European Conference on Computer Vision. Cambridge, UK: Springer (1996)591–602
2. Duan, L., Cui, G., Gao, W., Zhang, H.: Journal of Computer Aided Design and Computer Graphics 14(5) (2002) 404–409
3. Fan, X., Shen, X.: Filter of Erotic Images based on IE. Journal of Jilin University (Information Science Edition) 22(6) (2004) 631–637
4. Schwerdt, K., Crowley, J.: Robust Face Tracking using Color. Proceedings of Fourth IEEE International Conference on Automatic Face and Gesture Recognition. Grenoble, France (2000)
5. Dorai, C., Ratha, N., Bolle, R.: Detecting Dynamic Behavior in Compressed Fingerprint Videos: Distortion. Proc. of the Computer Vision and Pattern Recognition. Hilton Head: IEEE Computer Society. (2000) 13–15
6. Brand, J., Mason, J.S.: A Comparative Assessment of Three Approaches to Pixellevel Human Skin-Detection. Proceedings of the International Conference on Pattern Recognition. Barcelona, Spain (2000) 1056–1059
7. Vapnik, V.N.: The Nature of Statistical Learning Theory. NY: Apringer –Verlag (1995)
8. Osuna, E., Freund, R., Girosi, F.: Training Support Vector Machines: An Application to Face Detection. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Puerto Rico. (1997) 130–136
9. Platt, J.C.: Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Technical Report. MSR–TR–98–14 (1998)

Sensor Validation Using Nonlinear Minor Component Analysis

Roger Xu[1], Guangfan Zhang[1], Xiaodong Zhang[2], Leonard Haynes[1], Chiman Kwan[1], and Kenneth Semega[3]

[1] Intelligent Automation, Inc., 15400 Calhoun Drive, Suite 400, Rockville, MD 20855
[2] GM R & D and Planning, 30500 Mound Road, Warren, MI 48090-9055, USA
[3] 1950 Fifth Street, Building 18, RM D036, WPAFB, OH 45433-7251, USA

Abstract. In this paper, we present a unified framework for sensor validation, which is an extremely important module in the engine health management system. Our approach consists of several key ideas. First, we applied nonlinear minor component analysis (NLMCA) to capture the analytical redundancy between sensors. The obtained NLMCA model is data driven, does not require faulty data, and only utilizes sensor measurements during normal operations. Second, practical fault detection and isolation indices based on Squared Weighted Residuals (SWR) are employed to detect and classify the sensor failures. The SWR yields more accurate and robust detection and isolation results as compared to the conventional Squared Prediction Error (SPE). Third, an accurate fault size estimation method based on reverse scanning of the residuals is proposed. Extensive simulations based on a nonlinear prototype non-augmented turbofan engine model have been performed to validate the excellent performance of our approach.

1 Introduction

While FDI provides significant potential in improving safety and performance of future advanced jet engines, clearly the success of this method is highly dependent upon the accuracy of the sensor signals used to drive FDI approaches. The approaches to sensor fault detection/isolation are usually categorized into two types: model-based and data-driven. Model-driven approaches are preferable when a physical model of the system is available [1 and 2]. However, in many applications, physical models may not be available or may be inaccurate, especially for systems with complex nonlinear dynamics. Data-driven approaches do not require physical models and therefore do not have the limits aroused from the model-based approach. Recent research advances in data-driven FDI and fault accommodation methods, for example, Fuzzy Logic inference, Neural Networks (NN), Case-Base Reasoning (CBR) [3]. However, all these methods require thorough information about system behaviors in different fault modes for fault diagnostics. In practice, such data are not always available. Therefore, data-driven approaches are usually criticized due to this strict requirement.

In this paper, a data-driven sensor fault detection and isolation (FDI) scheme is presented. This novel data-driven approach is based on Nonlinear Minor Component

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 352 – 357, 2006.

Analysis (NLMCA) and is designed for nonlinear system FDI. Compared with other sensor FDI methods, the proposed approach does not require a physical model and only needs training data in normal conditions that is usually easily accessible. This property distinguishes our approach from many other data-driven approaches that require faulty data in training. Meanwhile, a reverse scan method is used to reconstruct the faulty signal. Finally, extensive simulations were performed to illustrate the effectiveness of the proposed scheme with a nonlinear engine model from NASA.

2 FDI Based on Nonlinear Minor Component Analysis Architecture

The Nonlinear Minor Component Analysis (NLMCA) based Fault Detection and Isolation (FDI) approach is developed for the sensor validation of nonlinear dynamic systems. The architecture of NLMCA for sensor validation is shown in Fig. 1. This architecture contains three main modules: detection, isolation, and size estimation.

The fault detection module is used for detecting sensor faults and it is based on the NLMCA method and Squared Weighted Residual (SWR) generation. After a fault is detected, the fault isolation estimator is activated. The fault isolation estimator contains a bank of m NLMCA structure, where m is the number of sensors. With the fault isolated, a reverse scan method is used to estimate the degradation status, i.e., the fault size of the faulty sensor.

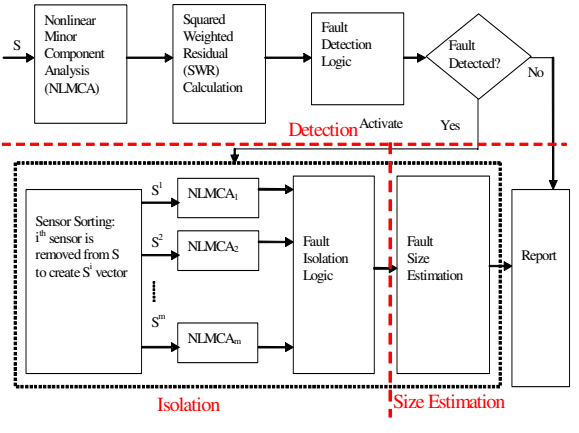


Fig. 1. NLMCA for sensor fault detection/isolation architecture

2.1 Fault Detection Scheme

2.1.1 Nonlinear Minor Component Analysis

The fault detection scheme is based on Nonlinear Minor Component Analysis (NLMCA) technique. Various Neural Network (NN) methods have been developed

for performing the NLPCA [5, 6, and 7]. NLMCA can be performed using the same structure as NLPCA. The first principal component can be extracted using a NN structure in Fig. 2. The cost function is defined in the following equation.

$$J^1 = \| e^1 \|^2 = \| X - X^{'} \|^2$$

If the cost function is minimized, then u can be regarded as the first principal component. To obtain the second principal component and other principal components for nonlinear systems, we can feed the residual e^1 into the same NLPCA structure. Also, the nonlinear minor components can be extracted based on the nonlinear principal components: $MC^1 = PC^{n-k+1}$, $MC^2 = PC^{n-k+2}$, …, $MC^k = PC^n$. The overall structure of NLMCA is shown in Fig. 3.

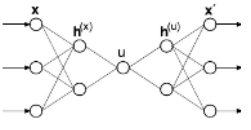


Fig. 2. NLPCA structure [4]

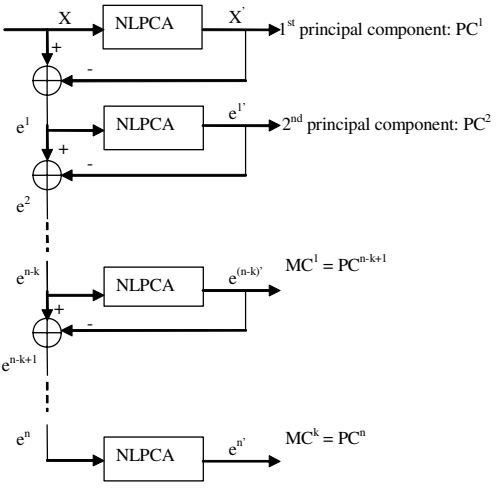


Fig. 3. NLMCA Architecture

2.1.2 NLMCA for Sensor Fault Detection

Usually, a so-called Square Predicted Error (SPE) is used to indicate the presence of a fault. The SPE is defined as

$$d_{SPE} = e^T e$$

where e is extracted from the minor components. In our approach, instead of using SPE, Squared Weighted Residual (SWR) ([8]), is used as the fault detection index in our approach. The SWR is given by $d_{SWR} = e^T R_s^{-1} e$, where R_s is derived from training data, $R_s = E(e*e^T)$. It has been proved that the revised index is more sensitive to faults and more robust to noises. The normal dynamics is captured in the minor component space, and therefore any abnormally increase in the SWR indicates an abnormal situation. Based on this feature, the approach of NLMCA to sensor fault detection is accordingly developed.

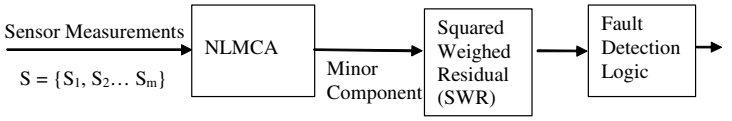


Fig. 4. NLMCA for fault detection architecture

2.2 Fault Isolation Scheme

After a fault is detected, the fault isolation estimator is activated. To isolate sensor fault, a bank of NLMCA structures is built, as shown in Fig. 5. Each NLMCA structure (NLMCAS) is used to monitor one sensor only and uses training data from all the other sensors. Assuming that the original NLMCAS is modeling the sensors $S=\{S_1, S_2, \ldots, S_m\}$ and a fault, bias or drift, occurs on sensor S_i, then the minor components will be detected to be larger than predefined thresholds. If we use a subset of sensors S, $S_i=\{S_1, \ldots, S_{i-1}, S_{i+1}, \ldots, S_m\}$ to build NLMCAS, it is easy to see that the residual will remain small. Meanwhile, all the other isolation NLMCAS will most likely produce a high SWR since they are affected by sensor S_i.

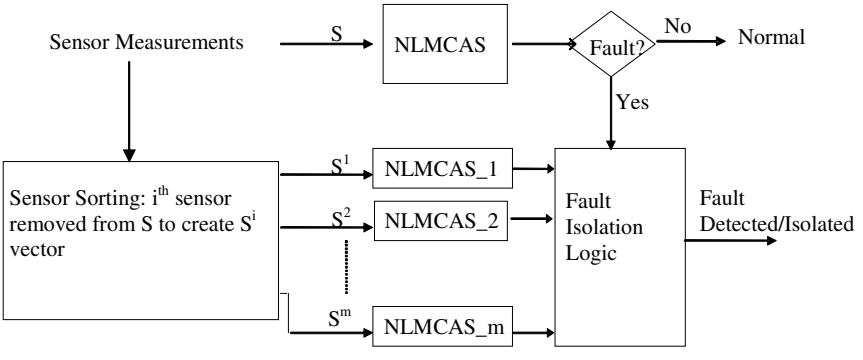


Fig. 5. Fault isolation logic

2.3 Fault Size Estimation

After a fault is isolated, it is critical to estimate the fault size and reconstruct the faulty sensor outputs. We use a reverse scan method to reconstruct the faulty sensor signals. We substitute the faulty sensor measurement with a value selected from a certain range, for example, ±p% of the measurement value of the faulty sensor, and calculate the SWR for each substituted sensor measurements. The sensor value with the minimum SWR is assumed to be the "true" value of the faulty sensor.

3 Simulation Results

This sensor validation approach was validated and verified using a nonlinear NASA engine model. The nonlinear engine model is a prototype non-augmented turbofan engine model [9]. The Table 1 shows the sensors in the engine model.

Table 1. Sensor list ([9])

ID	1	2	3	4	5	6	7	8	9	10
Sensor Name	T2	P2	NL	NH	T27	P27	PS3	T3	PS5	T5

To generate baseline training data, a thorough simulation is run by varying the four set-point inputs:

– Altitude: an ambient input (from sea level to 70,000 feet).
– Mach number: from 0 to 0.65.
– DTamb: difference in ambient temperature from that of a standard day. (in deg F)
– PLA: N1 demand. It is compared to the feedback N1 and this error is used to adjust the fuel demand.

To illustrate our approach, a bias fault is first initiated on sensor T2 and the fault size is 3% of its nominal value. A white noise with a variance of 2% of the nominal value of each sensor is added in this model. The bias fault is initiated at time t = 90 sec. The fault detection result is shown in Fig. 6.

Clearly, by examining the fault detection index in Fig. 6, we can detect the fault soon after it is initiated. Following the detection of a fault, the isolation logic is activated, and the fault on sensor T2 is isolated according to the fault isolation indices.

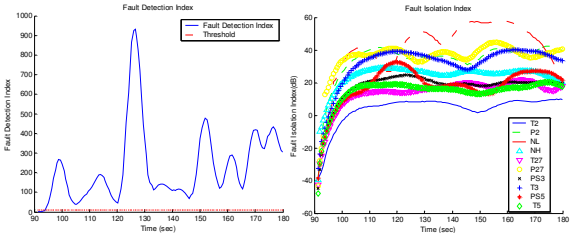


Fig. 6. Fault detection/isolation results

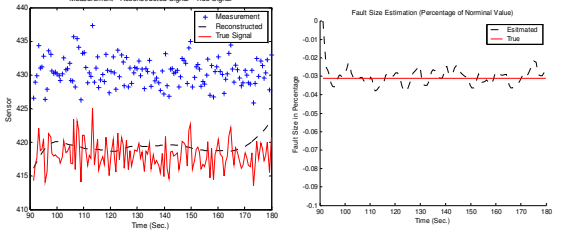


Fig. 7. Fault size reconstruction

As shown in Fig. 7, the fault size is accurately estimated based on the proposed fault size estimation technique.

4 Conclusion

In this paper, the Nonlinear Minor Component Analysis (NLMCA) based FDI approach is presented. This approach is especially suitable when the model of a dynamic system is either unavailable or inaccurate, and when the system is non-linear (all real systems contain more or less nonlinearities). Our innovative approach requires only normal operational data for training. With this approach, fault detection and isolation is accomplished through a fault detection NLMCA model and a bank of fault isolation NLMCA models. Meanwhile, a reverse scan method is utilized for the fault size estimation purpose and simulation results with performance analysis are reported to illustrate the proposed approach. In the future, it is practical to perform diagnostic reasoning in transient periods.

Acknowledgements

This research was supported by the Air Force under contract number FA8650-05-M-2582. We would like to thank Mr. Semega for his continuous support.

References

1. Chen, J. and Patton, R.J.: Robust Model-Based Fault Diagnosis for DynamicSystems. London, U.K.: Kluwer (1999)
2. Zhang, X., Parisini, T., and Polycarpou, M. M.: A Sensor Bias Fault Isolation Scheme for a Class of Nonlinear Systems. IEEE Transactions on Automatic Control, to appear
3. Mark, D. and Bill, C.: Case-based Reasoning for Gas Turbine Diagnostics. American Association for Artificial Intelligence., 2005..
4. Hines, J.W., Uhrig, R.E. and Wrest, D.J.: Use of Autoassociative Neural Networks for Signal Validation. Proceedings of NEURAP 97 Neural Network Applications, Marseille, France, March 1197
5. Kramer, M.A.: Nonlinear Principal Component Analysis Using Autoassociative Neural Networks. AICHE Journal 37 (1991) 233-243
6. William W. Hsieh: Nonlinear Principal Component Analysis by Neural Networks. Tellus 53A (2001) 599-615.
7. Oja, E., Wang, L.: Neural fitting: Robustness by Anti-Hebbian Learning. Neurocomputing 12 (1996) 155-170
8. IAI: A Novel Health Monitoring Approach for Hydraulic Pumps and Motors. Phase 2 SBIR Proposal, June 26, 2002
9. Mink, G.: ICF Generic Engine Model Documentation. Scientific Monitoring, Inc.

Fault Diagnosis with Enhanced Neural Network Modelling

Ding-Li Yu and Thoon-Khin Chang

Control Systems Research Group, Liverpool John Moores University, UK
d.yu@livjm.ac.uk

Abstract. A neural network (NN) based fault detection and isolation (FDI) approach for unknown non-linear system is proposed to detect both actuator and sensor faults. An enhanced parallel (independent) NN model is trained to represent the process and used to generate residual. A mean-weight strategy is developed to overcome the un-modelled noise and disturbance problem. A signal pre-processor is also developed to convert the quantitative residual to qualitative form and applied to a NN fault classifier to isolate different faults. The developed techniques are demonstrated with a multi-variable non-linear tank process.

1 Introduction

The model-based approaches to FDI in automated processes have received considerable attention for the last two decades [1]-[2]. Since an accurate mathematical model is difficult and costly to develop for a complex non-linear dynamic system, NNs have been applied for this purpose [3]. Most of the NN-based FDI approaches for unknown, non-linear systems contain detection and isolation parts [4]-[5]. In their approaches, a series-parallel NN is implemented to perform one-step-ahead prediction of the system output, and another NN, known as fault classifier is trained to learn the fault patterns. Because series-parallel NN model uses the past process output with fault effect as input, it will be influenced by the actuator or sensor faults. Therefore it will lead that series-parallel NN follows the faulty system output. The residual generated by this series-parallel NN is small and sensitive to modelling error.

In this paper, the proposed NN-based FDI scheme employ an enhanced parallel NN model. The parallel NN model contain two NNs, the first NN is trained by the series-parallel model method, then implemented as a parallel model.

The difference between the parallel NN output and process output is used as residual to detect faults. For fault isolation, an information pre-processor is used to convert the quantitative residual to qualitative form. This qualitative information is then used to a NN, known as fault classifier, which is trained to learn the fault patterns, which could be obtained from the past fault records. The developed NN-based FDI techniques are demonstrated in a multi-variable, non-linear tank process. The MLPN is used in this NN-based FDI simulation.

2 NN-Based FDI Scheme

An NN-based FDI scheme with an enhanced parallel NN model is proposed as shown in Fig.1. The parallel NN model is trained as a healthy model of the non-linear

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 358 – 363, 2006.

process. The residual, *e(t)* is defined as the difference between parallel NN output, $y_{NN}(t)$ and process output, *y(t)*

$$e(t) = y(t) - y_{NN}(t) \tag{1}$$

The residual vector, *e(t)* is used to generate fault alarm. Hence, if $|e(t)| > \delta$ then a fault is detected, where δ is a pre-specified threshold which is signed according to modelling error. This residual, *e(t)* is also used in an information pre-processor, which converts the quantitative of *e(t)* to qualitative form. The qualitative output of information pre-processor, *q(t)* is used to isolate the faults by a NN-based fault classifier. The reason for employing such qualitative information is the fault classifier will be trained faster and more easily.

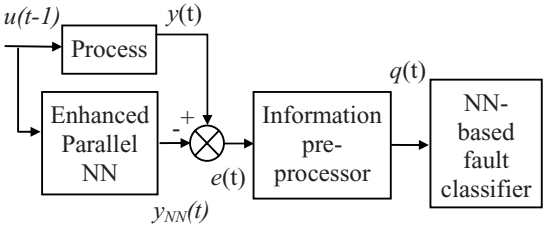


Fig. 1. NN-based FDI scheme

3 Enhanced Parallel NN Model Structure

An NN model is trained in series-parallel mode as (2),

$$y_{SP}(t) = NN_{SP}[u(t-1), \cdots, u(t-n_u), y(t-1), \cdots y(t-n_y)] \tag{2}$$

where $y_{SP}(t)$ is the series-parallel NN output, *u(t)* and *y(t)* are the system input and output respectively. n_u and n_y are the maximum time delay for u and y respectively, which are determined by simulation experiment. After learning, this series-parallel NN is implemented as parallel NN, which has the following structure:

$$y_P(t) = NN_{SP}[u(t-1), \cdots, u(t-n_u), y_p(t-1), \cdots, y_p(t-n_y)] \tag{3}$$

The series-parallel NN modelling error, $e_{SP}(t)$, and parallel NN modelling error, $e_P(t)$ are defined by:

$$e_{SP}(t) = y(t) - y_{SP}(t) \tag{4}$$

$$e_P(t) = y(t) - y_P(t) \tag{5}$$

hence eq. (3) can be rewritten as

$$y_P(t) = NN_{SP}[u(t-1), \cdots, u(t-n_u), y(t-1) - e_p(t-1), \cdots, y(t-n_y) - e_p(t-n_y)] \tag{6}$$

The $NN_{SP}(.)$ is trained to track the non-linear system output trajectory, *y(t)* with input pattern in eq. (2). Hence, $e_P(t)$ is greater than $e_{SP}(t)$. In this scheme, another NN with the parallel NN output as an input is trained to track the non-linear system output trajectory, *y(t)*.

$$y_{ER}(t) = NN_{ER}[u(t-1), y_p(t)] \tag{7}$$

The mean-weight strategy is an approach using the average of the outputs of three network models. Hence, the overall unmodelled noise and disturbances in the network model are reduced.

$$y_{mean}(t) = \frac{y_{ER1}(t) + y_{ER2}(t) + y_{ER3}(t)}{3} \tag{8}$$

where $y_{mean}(t)$ is the mean weight of the three $NN_{ER}(.)$ output. $y_{ER1}(t)$, $y_{ER2}(t)$ and $y_{ER3}(t)$ are the outputs of three $NN_{ER}(.)$ respectively.

4 FDI Applications to a Process

The tank has two inlet flows with time varying flow rates $f_1(t)$ and $f_2(t)$. Both feeds contain dissolved material with constant concentrations c_1 and c_2 respectively. The outlet flow has a flow rate $f(t)$, and concentration $c(t)$. Assume that the tank is stirred well so that the concentration of the outgoing flow equals the concentration in the tank. The process can be mathematically described as:

$$\frac{dv(t)}{dt} = f_1(t) + f_2(t) - k\sqrt{\frac{v(t)}{b}} \tag{9}$$

$$\frac{dc(t)}{dt} = \frac{c_1 - c(t)}{v(t)} f_1(t) + \frac{c_2 - c(t)}{v(t)} f_2(t) \tag{10}$$

where $v(t)$ and $c(t)$ are the process outputs, $f_1(t)$ and $f_2(t)$ are the inputs of the process.

Based on eqns. (9) and (10), a simulation model is developed. The physical parameters are k=0.02, c_1=1M, c_2=2 M, b=1m^2. Input, $u_1(t)$ and $u_2(t)$ have amplitude in the interval [0, 0.02]. The sample time is chosen as one second.

In this paper, four faults were simulated on $f_1(t)$, $f_2(t)$, $v(t)$ and $c(t)$ of the process to evaluate the NN-based FDI scheme's performance. Consider that the process is mathematically described as,

$$y(t) = \begin{bmatrix} v(t) \\ c(t) \end{bmatrix} = F\left[f_1(t), f_2(t)\right] \tag{11}$$

Two actuator faults are simulated as multiplicative faults with increasing rate $p_1(t)$ and $p_2(t)$. Two sensor faults are simulated as multiplicative faults with increasing rate $p_3(t)$ and $p_4(t)$. Then, when faults occur, the process sensor output, $y_f(t)$ with fault can be described as,

$$y_f(t) = \begin{bmatrix} p_3(t) \\ p_4(t) \end{bmatrix} F[p_1(t) f_1(t),\ p_2(t) f_2(t)] \tag{12}$$

If no fault occurs in the process, all fault parameters equal to one.

In this study, the MLPN is implemented as the parallel NN model and parallel modelling error reducer. After the investigation, the best parallel NN and parallel modelling error reducer structures are found as below:

$$y_P(t) = NN_{SP(6:10:2)}[u(t-1), y_p(t-1), y_p(t-2)] \tag{13}$$

$$y_{AER}(t) = NN_{AER(6:10:2)}[u(t-1), y_p(t), y_p(t-1)] \tag{14}$$

Table 1 shows the sum-squared-error (SSE) of 4000 samples of testing data by the three enhanced parallel NNs with and without parallel modelling error reducer. It can be observed that the average model with parallel error reducer has the smallest SSE.

Table 1. SSEs of three enhanced models

	Without error reducer	With error reducer
SSE_1	1.1205	0.95053
SSE_2	0.84876	0.51109
SSE_3	1.9926	1.8673
SSE_{mean}	**0.70034**	**0.48445**

Consider the residual, *e(t)* in eq. (1) defined as $e(t) = [e_v(t)\ e_c(t)]^T$. The difference between the enhanced parallel NN mean estimation and process output is used as a residual vector, *e(t)* to generate a fault alarm.

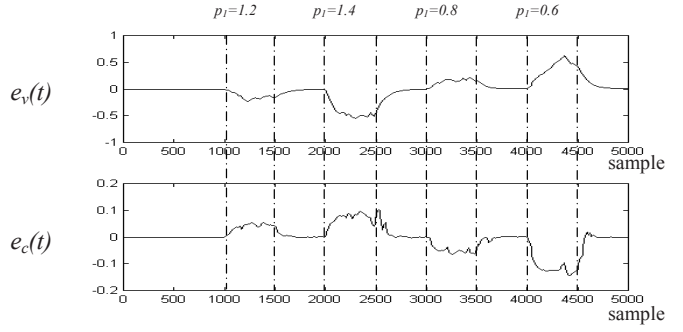


Fig. 2a. Residual of f_1 fault

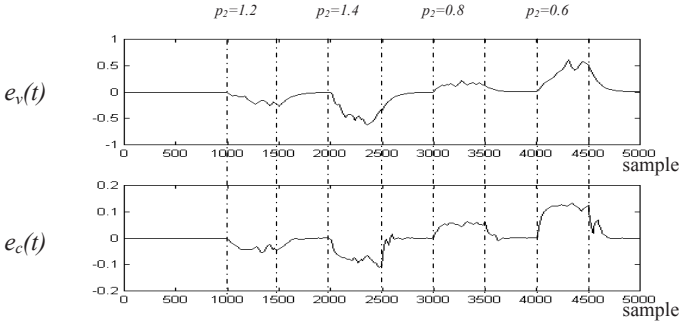


Fig. 2b. Residual of f_2 fault

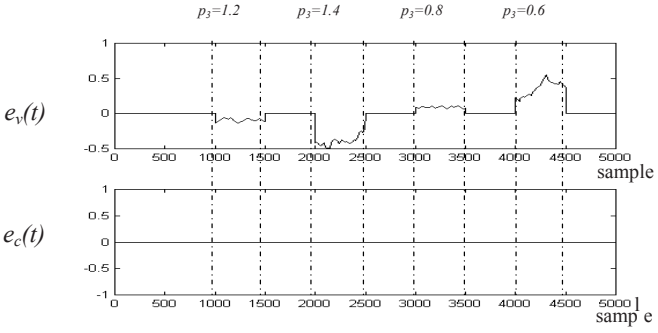


Fig. 2c. Residual of *v* fault

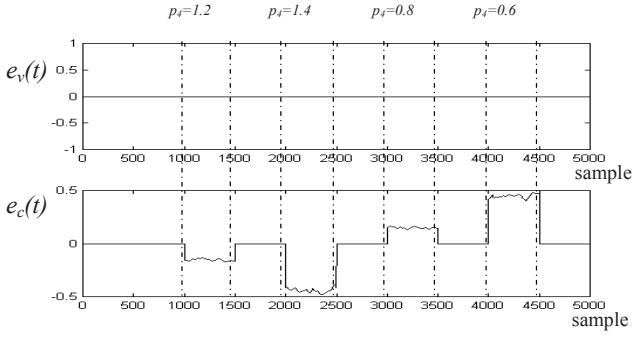


Fig. 2d. Residual of *c* fault

Ideally, if $|e_v(t)|>0$ or $|e_c(t)|>0$ then fault is detected. However, the realistic situation is different due to the present of unmodelled noise or disturbance and other uncertain factors. Therefore, a threshold is used to avoid the setting of an incorrect alarm. In our study, a threshold of [0.055 0.0135] is used. Hence, if $|e_v(t)|>0.055$ or $|e_c(t)|>0.0135$ then a fault alarm is active. Fig.2 shows four faults with fault parameter 1.2, 1.4, 0.8 and 0.6 are detected. Each fault occurs at sample interval [1000, 1499], [2000, 2499], [3000, 3499] and [4000, 4499] respectively.

Table 2. Fault pattern of the tank process

	q_v	q_c	y_{FC1}	y_{FC2}	y_{FC3}	y_{FC4}
f_1	1	3	1	0	0	0
fault	3	1	1	0	0	0
f_2	1	1	0	1	0	0
fault	3	3	0	1	0	0
v	1	2	0	0	1	0
fault	3	2	0	0	1	0
c	2	1	0	0	0	1
fault	2	3	0	0	0	1

4.5 Fault Isolation Results

In the fault isolation, the residual vector, *e(t)* is converted to *q(t)* with qualitative form: more than zero, zero and less than zero. Here we use value 1, 2 and 3 to represent these grades. A threshold is applied due to the modelling error. Thus, the conversion can be described as

$$q_v(t) = \begin{cases} 1, & e_v(t) < -0.055 \\ 2, & -0.055 \le e_v(t) \le 0.055 \\ 3, & e_v(t) > 0.055 \end{cases}, \quad q_c(t) = \begin{cases} 1, & e_c(t) < -0.0135 \\ 2, & -0.0135 \le e_c(t) \le 0.0135 \\ 3, & e_c(t) > 0.0135 \end{cases} \tag{15}$$

The qualitative, *q(t)* used to isolate the faults by a NN-based fault classifier. In this simulation, a MLPN is used as fault classifier. The fault classifier is designed and trained such that each output represents one fault when it is one. After the investigation, the best structure of the fault classifier is found as below:

$$y_{FC}(t) = NN_{FC(2:5:4)}[q(t)] \tag{16}$$

A fault pattern of the tank process is determined in Fig.2. Table 2 shows the fault pattern used in this application to train the NN classifier. After training, the classifier is used to isolate these four faults and the results are same as displayed in table 2.

5 Conclusions

An enhanced FDI scheme based on NN model and classifier is proposed in this paper. The reliability of the FDI scheme depends on the accuracy of the parallel NN model, and the past fault records of the process. The proposed parallel modelling error reducer is able to improve the accuracy process modelling with parallel MLPN, and the mean weight strategy is able to compensate the un-modelled noise by three models. Thus, combine the parallel modelling error reducer with the mean weight strategy is advantageous in improving parallel NN modelling, reducing the threshold of fault detection, and improves the accuracy of fault detection.

References

1. Frank, P.M.: Fault Diagnosis in Dynamic Systems Using Analytical and Knowledge-Based Redundancy - A Survey and Some New Results, Automatica 26 (3) (1990) 459-474.
2. Isermann, R.: Fault Diagnosis of Machines via Parameter Estimation and Knowledge Processing - Tutorial Paper, Automatica 29 (5) (1993) 815-835.
3. Patton, R.J., Chen, J. and Siew, T.M.: Fault Diagnosis in Non-Linear Dynamic Systems via Neural Networks, Proc. IEEE Int. Conf. CONTROL 94, Coventry, UK, March 21-24 (1994) 1346-1351.
4. Yu, D.L., Gomm, J.B. and Williams, D.: Sensor Fault Diagnosis in A Chemical Process via RBF Neural Networks, Control Eng. Practice 7 (1) (1999) 49-55.
5. Kolvo, H.N.: Artificial Neural Networks in Fault Diagnosis and Control, Control Eng. Practice 2 (1) (1994) 89-101.

Fault Detection and Diagnosis Using Neural Network Design

Kok Kiong Tan, Sunan Huang, and Tong Heng Lee

National University of Singapore, 4 Engineering Drive 3, Singapore 117576
{eletankk, elehsn, eleleeth}@nus.edu.sg

Abstract. In this work, a fault detection method is designed based on neural networks. The proposed method is that a neural network is built on-line for the normal mode, while other one is used to diagnose the faults. The simulation shows the effectiveness of the proposed method.

1 Introduction

Recently, various methods have been developed for fault detection and diagnosis. The application of statistical algorithms to associate patterns is found in [1]. In [2], acoustic emission sensor and accelerometers are used to monitor progressive stages. In [3], linear model is built to detect the tool wear and breakage. However, due to the nonlinear nature of the industrial proess, linearized model is difficult to apply. Neural networks (NNs) have been considered as general tools for modeling unknown functions. In [4], a NN is used to approximate the unknown cutting process by on-line learning and a nonlinear observer is designed based on the built neural network model. Unforunately, the states are assumed to be known. In this work, model-based process supervision with fault detection and diagnosis is presented. There are two NNs used in the design. The first NN is to learn an appropriate system's characteristic when it is operating normally. The unknown parameters (neural network weights) are obtained by update learning laws. The model thus obtained is then used as a state observer to monitor the process. Thereafter, a diagnosis technique is developed using the second NN to capture the nonlinear characteristics of faults.

2 Problem Statements and Fault Detection

Consider the following nonlinear model

$$\begin{aligned} \dot{x} &= Ax + bf(x,u), \\ y &= C^T x, \end{aligned} \tag{1}$$

with $f(x,u)$ is unknown and smooth function. An observer that estimates states in (1) is given by

$$\dot{\hat{x}} = A\hat{x} + b[\hat{f}(\hat{x},u)] + K(y - C^T\hat{x}) \tag{2}$$

$$\hat{y} = C^T\hat{x} \tag{3}$$

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 364–369, 2006.

where $\hat{x}$ denotes the estimate of the state x, $K = [k_1, k_2, ..., k_n]^T$ is the observer gain vector. The function $\hat{f}(\hat{x}, u)$ is estimate of $f(x, u)$ in terms of $\hat{x}$. Define the state and output estimate errors as $\tilde{x} = x - \hat{x}$ and $\tilde{y} = y - \hat{y}$. Thus, the error dynamics is given by

$$\dot{\tilde{x}} = (A - KC^T)\tilde{x} + b[f(x, u) - \hat{f}(\hat{x}, u)], \tag{4}$$

$$\tilde{y} = C^T\tilde{x}. \tag{5}$$

As the function $f(x, u)$ is smooth, the following approximation holds on a compact set $(x, u) \in \Omega_{xu}$:

$$f(x, u) = W^{*T}\Phi(x, u) + \epsilon(x, u), \tag{6}$$

with bounded function approximation error $\epsilon(x, u)$, i.e., $||\epsilon(x, u)|| \leq \epsilon_M$. The magnitude of ϵ depends on the choices for basis function and node number. Let $\hat{W}$ be estimates of the ideal W^* and the estimation errors are defined as $\tilde{W} = W^* - \hat{W}$. Then,

$$f(x, u) - \hat{f}(\hat{x}, u) = \tilde{W}^T\Phi(\hat{x}, u) + d_{xu} + \epsilon \tag{7}$$

where $d_{xu} = W^{*T}[\Phi(x, u) - \Phi(\hat{x}, u)]$. For the radial basis activation function, every element of $\Phi(x, u) - \Phi(\hat{x}, u)$ is bounded. Thus,

$$|d_{xu}| \leq 2||W^*||\phi_m. \tag{8}$$

NN observer design. The following adaptive observer is given by

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + b[\hat{W}^T\Phi(\hat{x}, u)] + K(y - C^T\hat{x}) \\ \hat{y} &= C^T\hat{x} \end{aligned} \tag{9}$$

Considering (9), the error equation (4) becomes

$$\dot{\tilde{x}} = \bar{A}\tilde{x} + b[\tilde{W}^T\Phi(\hat{x}, u) + d_{xu} + \epsilon], \tag{10}$$

where $\bar{A} = A - KC^T$. Consider the nonlinear system described by (1), the observer (9), and the following weight adaptation laws:

$$\dot{\hat{W}} = k_0^2\tilde{y}F\Phi(\hat{x}, u) - k_wF(\hat{W} - W_0), \tag{11}$$

where $k_0 > 0, F > 0, k_w > 0$ and W_0 are designed by users. We are now in a position to state the main result of the work.

Assumption 1. $H(s) = C^T(sI - \bar{A})^{-1}b$ with $\bar{A}$ being a Hurwitz matrix that is SPR.

Theorem 1. Consider the nonlinear system described by (1), the observer (9), and the weight adatation laws (11). If Assumption 1 holds, $x_0 \in \Omega_{x0}$, then all the signals are bounded and the state estimate $\hat{x}$ still remains in the compact set $\Omega_{\hat{x}} = \{\hat{x}| \, ||x - \hat{x}|| \leq M_\epsilon, x \in \Omega_x\}$.

In Theorem 1, it is supposed that the output error equation is SPR, which is restrictive assumption. In the following analysis, the SPR imposed on the output error equation is relaxed in part. The estimation error e is given by

$$e = H(s)[\tilde{W}^T \Phi(\hat{x}, u) + d_{xu} + \epsilon] \tag{12}$$

where $H(s)$ is realized by $(A - KC^T, b, C)$. It is quite difficult to satisfy the SPR condition for $H(s)$. To reduce the constraint, we introduce a transfer function $L(s)$ which is chosen so that $H(s)L(s)$ is SPR, and $L^{-1}(s)$ is stable transfer function. The output estimation equation is re-written as

$$e = H(s)L(s)[\tilde{W}^T \hat{\Phi}(\hat{x}, u) + \hat{d}_{xu} + \hat{\epsilon}] \tag{13}$$

where $\hat{\Phi} = L^{-1}(s)\Phi(\hat{x}, u), \hat{d}_{xu} = L^{-1}(s)d_{xu}, \hat{\epsilon} = L^{-1}(s)\epsilon$. This error dynamics is only for analysis purpose. We will show later that only the NN basis function is filtered by $L^{-1}(s)$. The error equation (13) may be written as a state equation described by

$$\dot{\tilde{z}} = A_c \tilde{z} + b_c[\tilde{W}^T \hat{\Phi}(\hat{x}, u) + \hat{d}_{xu} + \hat{\epsilon}] \tag{14}$$

$$e = C_c^T \tilde{z} \tag{15}$$

where (A_c, b_c, C_c) is a minimal state representation of $H(s)L(s)$ with $C_c = [1, 0, ..., 0]^T$. The following theorem is given to establish the stability under the proposed design.

Theorem 2. Consider the nonlinear system described by (1), the observer (9), and the weight adatation laws provided by

$$\dot{\hat{W}} = k_0^2 \tilde{y} F \hat{\Phi}(\hat{x}, u) - k_w F(\hat{W} - W_0), \tag{16}$$

If $x_0 \in \Omega_{x0}$, then all the signals are bounded and the state estimate $\hat{x}$ still remains in the compact set $\Omega_{\hat{x}} = \{\hat{x} \mid ||x - \hat{x}|| \leq M_\epsilon, x \in \Omega_x\}$. In addition, arbitrarily small error of $||e||$ may be achieved by selecting a large k_0.

Model-based fault detection. After the NN is trained based on a normally working system, the NN can be used as a state observer. Subsequently, the adaptive learning law can be switched off and the NN parameters are fixed. From (10), we have

$$\tilde{x} = e^{\bar{A}t}\tilde{x}(0) + \int_0^t e^{\bar{A}(t-\tau)} b[\tilde{W}^T \Phi(\hat{x}, u) + d_{xu} + \epsilon] d\tau \tag{17}$$

Thus, the residual $e = y - C^T \hat{x}$ is given by

$$||e|| \leq \| C \| \| \tilde{x} \| \leq \mu e^{-\pi t} \epsilon_{x0} ||C|| + \frac{\mu ||b|| (\zeta_w + g)}{\pi} (1 - e^{-\pi t}) ||C|| \tag{18}$$

This implies that the upper bound for e can be chosen as $\varpi = \mu \epsilon_{x0} ||C|| + \frac{\mu ||b|| (\zeta_w + g)}{\pi} ||C||$. The fault detection problem is to generate a robust residual signal ϖ that satisfies

$$||e|| \leq \varpi \text{ if no fault,} \tag{19}$$

$$||e|| > \varpi \text{ if fault occurs.} \tag{20}$$

3 Fault Diagnosis with Parameter Estimation

We begin by introducing abrupt fault and incipient fault. Then, another neural network is used as the approximator to estimate these faults with the proposed learning algorithm. Consider the system (1) with a fault function described by

$$\begin{aligned} \dot{x} &= Ax + b[f(x,u) + I(t-T)\zeta(x,u)], \\ y &= C^T x, \end{aligned} \tag{21}$$

with

$$I(t-T) = \begin{cases} 0 & t < T \\ 1 - e^{-\theta(t-T)} & t \geq T \end{cases} \tag{22}$$

The term $\zeta(x,u) \in R^n$ is a vector which represents the fault in the system, $I(t-T) \in R$ represents the time profile of the fault, $\theta > 0$ is an unknown constant that represents the rate at which the fault in states and actuators evolves and T is the time of occurrence of the fault.

When the nonlinear function $\zeta(x,u)$ is unknown, the fault function $\beta(t-T)\zeta$ is not available. As the function $\zeta(x,u)$ is smooth, the following approximation holds:

$$\zeta(x,u) = \Xi^{*T}\Phi(x,u) + \xi, \tag{23}$$

with bounded function approximation error ξ satisfying $|\xi| \leq \xi_M$ and the ideal weight Ξ^* defined as:

$$\Xi^* := arg\min_{\Xi \in \Omega_v} \{ \sup_{(x,u)\in\Omega_{xu}} |\Xi^T\Phi(x,u) - \zeta(x,u)| \}, \tag{24}$$

The magnitude of ϵ_z depends on the choice of the basis function and the number of nodes. We consider an estimated model of the form

$$\left.\begin{aligned} \dot{\hat{x}} &= A\hat{x} + b[\hat{W}^T\Phi(\hat{x},u)] + K(y - C^T\hat{x}) + b\hat{\zeta}(\hat{x},u;\hat{\Xi}) \\ \hat{y} &= C^T\hat{x} \end{aligned}\right\} \tag{25}$$

where $\hat{\zeta}$ is a NN and $\hat{\Xi}$ represents the weights of the network. $\hat{\zeta}$ is given by $\hat{\zeta}(\hat{x},u) = \hat{\Xi}^T\Phi(\hat{x},u)$, where the weights $\hat{\Xi}$ is provided by the on-line tuning algorithm. The construction of an appropriate estimation model is a key step in designing the fault diagnosis scheme. The output of the above nonlinear estimation model is used to adaptively handle system failures. From (21), we can obtain

$$I(t-T)\zeta(x,u) - \hat{\zeta}(\hat{x},u) = \tilde{\Xi}^T\Phi(\hat{x},u) + I(t-T)\xi + z_{xu} \tag{26}$$

where $\tilde{\Xi} = I(t-T)\Xi^* - \hat{\Xi}$, and $z_{xu} = I(t-T)\Xi^{*T}[\Phi(x,u) - \Phi(\hat{x},u)]$. Note that Ξ^* is constant vector, $\Phi(x,u) - \Phi(\hat{x},u)$ is bounded and $I(t-T)$ is bouned, therefore, $||z_{xu}|| \leq ||\Xi^*||\phi_m$.

Based on the estimation model (25), we present an adaptive algorithm for tuning the weights of the neural network. We start by computing the error

equation from the measured vector and its estimate. Using (21) and (25), the following error dynamics is obtained

$$\dot{\tilde{x}} = (A - KC^T)\tilde{x} + b[f(x,u) - \hat{W}^T\Phi(\hat{x},u) + I(t-T)\zeta(x,u) - \hat{\Xi}^T\Phi(\hat{x},u)]. \quad (27)$$

$$e = y - C^T\hat{x} \quad (28)$$

Based on the error dynamics given by the above equation, we propose the following adaptive laws for the weights of the neural network:

$$\dot{\hat{\Xi}} = \Upsilon\Phi(\hat{x},u)D[e] - \eta||D[e]||\hat{\Xi} \quad (29)$$

where $\Upsilon = \Upsilon^T$ is a positive definite adaptation matrix and $D[.]$ is the dead-zone operator, defined as

$$D[e] = \begin{cases} 0_l \text{ if} ||e|| \leq \varpi \\ e \text{ otherwise} \end{cases} \quad (30)$$

where 0_l is an l-dimensional vector of zeros. We now examine the stability properties of the designed fault diagnosis algorithm. The stability involves the boundedness of all the signals in the fault detection system.

Theorem 3. In presence of faults, the nonlinear fault diagnosis algorithm described by (25) and (29) can ensure that $e(t), \tilde{x}$ and $\tilde{\Xi}$ are uniformly bounded.

If the SPR condition is not satisfied in Theorem 3, the proposed algorithm has to be modified. Let us analyze the output error equation.

$$e = H(s)[\tilde{W}^T\Phi(\hat{x},u) + d_{xu} + \epsilon + \tilde{\Xi}\Phi(\hat{x},u) + z_{xu} + I(t-T)\xi] \quad (31)$$

where $H(s)$ is realized by $(A - KC^T, b, C)$. Similar to Theorem 2, we introduce a transfer function $L(s)$ which is chosen so that $H(s)L(s)$ is SPR. The output estimation equation is re-written as

$$e = H(s)L(s)[\tilde{W}^T\hat{\Phi}(\hat{x},u) + \hat{d}_{xu} + \hat{\epsilon} + \tilde{\Xi}\hat{\Phi}(\hat{x},u) + \hat{z}_{xu} + I(t-T)\hat{\xi}] \quad (32)$$

where $\hat{\Phi} = L^{-1}(s)\Phi(\hat{x},u), \hat{d}_{xu} = L^{-1}(s)d_{xu}, \hat{\epsilon} = L^{-1}(s)\epsilon, \hat{z}_{xu} = L^{-1}(s)z_{xu}, \hat{\xi} = L^{-1}(s)\xi$. This error dynamics is only for analysis purpose. We will show later that only the NN basis function is filtered by $L^{-1}(s)$. The state equation (31) is given by

$$\dot{\tilde{z}} = A_c\tilde{z} + b_c[\tilde{W}^T\hat{\Phi}(\hat{x},u) + \hat{d}_{xu} + \hat{\epsilon} + \tilde{\Xi}\hat{\Phi}(\hat{x},u) + \hat{z}_{xu} + I(t-T)\hat{\xi}] \quad (33)$$

$$e = C_c^T\tilde{z} \quad (34)$$

where (A_c, b_c, C_c) is a minimal state representation of $H(s)L(s)$ with $C_c = [1, 0, ..., 0]^T$. The following theorem is given to establish the stability under the proposed design.

Theorem 4. In presence of faults, the nonlinear fault diagnosis algorithm described by the observer (25) and adaptive law provided by

$$\dot{\hat{\Xi}} = \Upsilon\hat{\Phi}(\hat{x},u)D[e] - \eta||D[e]||\hat{\Xi} \quad (35)$$

can achieve that $e(t), \tilde{x}$ and $\tilde{\Xi}$ are uniformly bounded.

4 Simulation Example

Consider the following second order nonlinear system

$$\dot{x} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} x + \begin{bmatrix} 3 \\ 1 \end{bmatrix} [u - 7x_1 sin^2(x_1)] \tag{36}$$

$$y = [0 \quad 1]x \tag{37}$$

The observer control gain is chosen as $K = [100, 20]^T$ such that the SPR condition is satisfied. The total number of the NN nodes is 20. Figure 1 presents the simulation results for the designed NN observer, where the solid line represents the actual response and the dashed line represents the estimated response. Thus, the established observer can be used as a monitor to detect the fault. Assume that the system after 4s has a fault in the form of $I(t-4)\zeta(x) = (1-e^{-0.6(t-4)})0.2x_2^2$. Thus, we trigger the second NN to estimate the fault function. Figure 1 also shows the time histories of the fault function and the second NN outputs. It can be observed that the second NN remains zero prior to the fault and changes to a certain value as the occurrence of the fault.

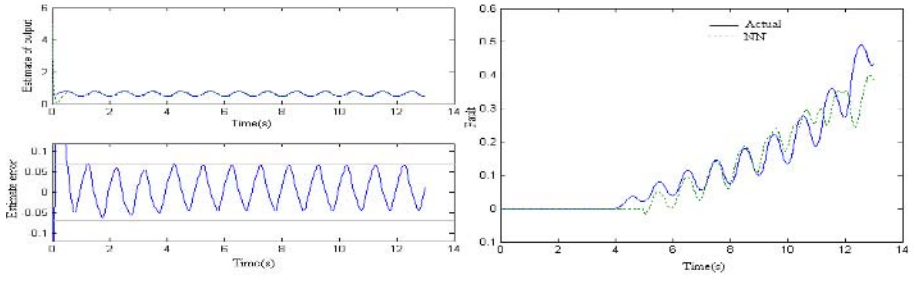


Fig. 1. Observer performance using NN

References

1. Rangwala,S., Liang,S., Dornfeld,D.: Pattern Recognition of Acoustic Emission Signals During Punch Stretching. Mechanical Systems and Signal Processing 1(1987)321-332
2. Prateepasen,A., Au,Y.H.J., Jones,B.E.: Acoustic Emission and Vibration for Tool Wear Monitoring in Single-point Machining Using Belief Nework. In:IEEE Instrumentation and Measurement Technology Conference. Hungary (2001)1541-1546
3. Isermann,R., Ayoubi,M., Konrad,H., Reiβ,T.: Model Based Detection of Tool Wear and Breakage for Machine Tools. In: Int. Conf. on Syst. Man & Cyber. Vol. 3. Le Touquet (1993)72-77
4. Colbaugh,R., Glass,K.: Real-time Tool Wear Estimation Using Recurrent Neural Networks. In: Proc. of the 1995 IEEE Int. Symp. on Intelligent Contr. Monterey, USA (1995) 357-362

Certainty Improvement in Diagnosis of Multiple Faults by Using Versatile Membership Functions for Fuzzy Neural Networks

Yuan Kang[1], Chun-Chieh Wang[1], Yeon-Pun Chang[1],
Chien-Ching Hsueh[2], and Ming-Chang Chang[3]

[1] Department of Mechanical Engineering,
Chung Yuan Christian University, Chung Li 320, Taiwan
[2] Power System Dept., Industrial Technology Research Institute, Hsinchu 310, Taiwan
[3] Magnate Aeronautical Industry Co.,Ltd, Kaohsiung 811, Taiwan
yk@cycu.edu.tw

Abstract. Because the relationship between frequency symptoms and fault causes are different, this study uses fuzzy neural network (FNN) with versatile membership functions to diagnose multiple faults in rotary machinery. According to the frequency symptom values for each fault causes, three kinds of membership functions are used. Besides, the structure of the FNN is large which spend much training time. Thus, when the matrix between frequency symptoms and fault causes can decoupled, the relational matrix decomposed into several sub-matrixes and the structure of the FNN can also divided into several sub-networks. In this study, two above-mention approaches are combined to diagnose multiple faults and compared with neural network (NN), FNN with single/versatile membership functions in two actual cases.

1 Introduction

The definition of the rotary machinery diagnosis is using vibration signals to distinguish faults by expert knowledge or experiments when the machine broke down. Recently, NN is widely used to solve these problems. Li [1] extracted frequency symptoms to diagnose faults by using NN for motor bearing system. Kuo [2] collected vibration symptoms of the power spectrum density for network inputs from ship propulsive system and classifying faults by using NN. It is better convenient then traditional method and has high diagnosis ability for faults. However, the diagnosis result is not exact by using NN when the weak characteristic signals in multiple faults.

Fuzzy reasoning is capable of dealing with imprecise information when NN is capable of learning form samples. FNN combines the advantages with fuzzy reasoning and NN. Goode [3] used membership function to represent each of the inputs for bearing fault detection by using FNN. Wu [4] extracted the frequency symptoms from the steam turbines, and applied FNN to infer faults. Tang [5] separated the symptoms including the temperature, pressure and combustion airflow, respectively, into three levels by utilizing membership functions and faults detection by using FNN.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 370 – 375, 2006.

The shortcomings of NN were improved by FNN with single membership function. Due to the different degree of relationship of inputs for each fault, it could not get ideal results by using single membership function of FNN. Thus, this study proposes versatile membership functions for different inputs of FNN. Besides, the structure of FNN is large which spend much time to train. This study proposed the sub-network approach to reduce the structure of FNN and saving much training time. Combining abovemention approaches can improve the certainty for multiple faults by using FNN.

2 Fuzzy Neural Network

The frame of the FNN retains the structure of the NN and increases the fuzzy layer between input and hidden layer. The derivation formula is detailed in Freeman and Skapura [6]. In this study, the input layer of FNN has 14 neurons corresponding to the 14 frequency symptoms and Table 1 (the explanation is listed in Appendix 1) shows the sets of frequency samples for 11 faults. The output layer has 11 neurons associated with the 11 faults. The training output samples are listed in Table 2. The first row corresponds to the fault type, and the (i+1)th row corresponds to the ith fault, i=1,2…,11. The fuzzy layer has 42 neurons arranged in 14 groups corresponding to the 14 frequency symptoms with each group containing 3 neurons.

2.1 Sub-networks/Sub-matrixes

The structure of FNN is so large that spend much training time. In order to saving the training time, the frame of the FNN was divided into three sub-networks including rotor, rolling bearing and electricity sub-networks as shown in Fig.1. And the training samples were divided followed as sub-matrixes. When the fault is not relative to the frequency symptom, the coefficient define zero. Therefore, the non-zero coefficients can concentrate into three sub-matrixes by rearrangement. The first matrix is rotor module including input neurons from S1 to S8 (Table 1) and output neurons from O1 to O5 (Table 2). The second matrix is rolling bearing module including input neurons from S2, S9 to S11 and output neurons from O1, O6 to O8. The third matrix is electricity module including input neurons from S2, S12 to S14 and output neurons from O1, O9 to O11. Consequently, this method can reduce the training time for dealing with non-relationship between frequency symptoms and faults.

3 Case Studies

The traditional FNN only use single membership function. But the degree of relationship of inputs is different for each fault. For example, the relation is different between S1 and S2 for unbalance fault. Therefore, the diagnosis results are not always satisfactory, even for multiple faults.

In this study, three kinds of membership functions were used and were showed in Fig.2. The input neurons (s1, s2, s4 and s8) belong to high saturation function and the membership function can be expressed by Eq. (1) and showed in Fig.2 (a). The input neurons (s3, s5, s6 and s7) belong to medium saturation function and the membership function can be expressed by Eq. (2) and showed in Fig.2 (b). The input neurons (s9

to s14) belong to low saturation function and the membership function can be expressed by Eq. (3) and showed in Fig.2 (c).

$$y_L = \begin{cases} -5x+1 & ,0 \le x \le 0.2 \\ 0 & ,0.2 \le x \le 1 \end{cases}, y_M = \begin{cases} 10x & ,0 \le x \le 0.1 \\ -10x+2 & ,0.1 \le x \le 0.2 \\ 0 & ,0.2 \le x \le 1 \end{cases}, y_H = \begin{cases} 5x & ,0 \le x \le 0.2 \\ 1 & ,0.2 \le x \le 1 \end{cases} \quad (1)$$

$$y_L = \begin{cases} -10x/3+1 & ,0 \le x \le 0.3 \\ 0 & ,0.3 \le x \le 1 \end{cases}, y_M = \begin{cases} 10x/3 & ,0 \le x \le 0.3 \\ -10x/3+2 & ,0.3 \le x \le 0.6 \\ 0 & ,0.6 \le x \le 1 \end{cases}, y_H = \begin{cases} 5x/3 & ,0 \le x \le 0.6 \\ 1 & ,0.6 \le x \le 1 \end{cases} \quad (2)$$

$$y_L = \begin{cases} -2x+1 & ,0 \le x \le 0.5 \\ 0 & ,0.5 \le x \le 1 \end{cases}, y_M = \begin{cases} 2x & ,0 \le x \le 0.5 \\ -2x+2 & ,0.5 \le x \le 1 \end{cases}, y_H = \begin{cases} 0 & ,0 \le x \le 0.5 \\ 2x-1 & ,0.5 \le x \le 1 \end{cases} \quad (3)$$

y_L, y_M, y_H are low, medium, high level, respectively. The value x is the coefficient of frequency symptom. The measured frequency symptoms set transform to fuzzy frequency symptoms set by membership function and obtain faults by computing with the FNN when faults diagnosing. Each of the output neuron represents a kind of fault and the output value represents the degree of certainty for corresponding fault. If the value is closed to 1, which represent the possibility of the fault is high.

This study uses versatile membership function and sub-networks separation to improve shortcomings of the traditional FNN and prove the DVFNN has good diagnosis usability for multiple. In this study, the abbreviated VFNN means FNN with using versatile membership function. The abbreviated DVFNN means FNN with using versatile membership function and sub-networks separation.

Case 1 is the induction motor that using fluid-film bearing. In Fig.3, it shows vibration frequency diagram and the rotating frequency is over twice then certain frequency (38Hz). Therefore, this case have oil whirl and unbalance by artificial diagnosing. The diagnosis results by using NN and FNN are listed in Table3 and the major fault is unbalance. But the degree of certainty of the oil whirl is low and is different with artificial diagnosis. The major reasons are unbalance and oil whirl by using VFNN and DVFNN. The degree of certainty of oil whirl fault is close to 1 so that the possibility for oil whirl is determined. The results are better than using NN and FNN.

Case 2 is the induction motor that using ball bearing. In Fig.4, it shows vibration frequency diagram. The vibration distributes over 60Hz, 2760Hz and 3600Hz. It's sure that have inner race damage and broken rotor bars, even has the misalignment by actual investigation. The diagnosis results by using NN and FNN are listed in Table4. The network output value of unbalance and broken rotor bars is closed to 1 and 0.9, respectively. And all of the others output values are not over 0.4 and it can not obtain accurate diagnosis result even by using FNN. Although, the network output value of the inner race defect rise to 0.623 by using VFNN, the result was not satisfied with the artificial diagnosis. In Fig.4, the twice rotary speed frequency is appeared beside the rotary speed frequency, broken rotor bars frequency and inner race defect frequency. In Table 4, the major reasons are unbalance, misalignment, inner race defect and broken rotor bars by using DVFNN. This method not only rise the degree of certainty for faults but also has more fault for misalignment than other methods. The results are accurate better than the foregoing methods and conform to artificial diagnosis.

4 Conclusions

In this study, diagnosis utilizing DVFNN for multiple faults in two cases can obtain satisfied results by comparing with NN, FNN and VFNN. These results infer some conclusions and following as:

1. Although VFNN can improve the shortcomings of FNN and rise the degree of certainty, the diagnosis results are still not the same with artificial reasoning. For the complex faults or the weak characteristic signals, using DVFNN can get better accurate results then using NN, FNN and VFNN. Consequently, this study proposes DVFNN is superior then the others method in complex rotating machinery system.
2. The DVFNN divides relation matrix into three modules according to different characteristic between frequency symptoms and faults. This method not only improves the difficulty for defining samples but saves the training times effectively.

Acknowledgements

This study was supported by the National Science Council, the Republic of China, under grant number NSC 89-TPC-7-033-008.

References

1. Li, B., Chow, M.Y., Tipsuwan, Y., Hung, J.C.: Neural-Network-Based Motor Rolling Bearing Fault Diagnosis. IEEE Transactions on Industrial Electronics 47 (2000) 1060-1069
2. Kuo, H.C., Wu, L.J., Chen, J.H.: Neural-Fuzzy Fault Diagnosis in a Marine Propulsion Shaft System. Journal of Materials Processing Technology 122 (2000) 12-22
3. Goode, P.V., Chow, M.Y.: Neural/Fuzzy Systems for Incipient Fault Detection in Induction Motors. Proceedings of IECON (1993) 332-337
4. Wu, C.Z., Yan, H., Ma, J.F.: Method Research of Noise Diagnosis Based on Fuzzy Neural Network. Fourth International Conference on ICSP '98. 2 (1998) 1370-1373
5. Tang, T., Zhu, Y., Li, J., Chen, B., Lin, R.: A Fuzzy and Neural Network Integrated Intelligence Approach for Fault Diagnosing and Monitoring. UKACC International Conference on Control '98. 2 (1998) 975-980
6. Freeman, J.A., Skapura, D.M.: Neural Networks Algorithms, Applications, and Programming Techniques. Addison Wesley (1992)
7. Rao, J.S.: Vibratory Condition Monitoring of Machines. Alpha Science Int. LTD (2000)

Appendix 1: Symbols for Frequency Symptoms and Fault Causes

$S_1 \sim S_8$: 0.5, 1, 1.5, 2, 2.5, 3, 4 and below 0.5X rotary frequency, respectively.

$S_9 \sim S_{11}$: The defect frequency of inner race, outer race and ball spin, respectively.

S_{12} : Two times of power supply frequency. S_{13} : The defect frequency of broken rotor bar.

S_{14} : Two times of the power supply frequency $\pm$ (n/3) times of the power supply frequency (n=0,1,2)

O_1 : Unbalance	O_2 : Misalignment	O_3 : Looseness	O_4 : Oil whirl
O_5 : Oil whip	O_6: Inner race defect	O_7 : Outer race defect	O_8 : Ball spin defect
O_9 : Air gap eccentricity	O_{10}: Broken rotor bar	O_{11}: Phasing problem	

Table 1. Relational matrix [7]

Symptoms / Faults	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}
O_1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
O_2	0	1	0	1	0	0.4	0.1	0	0	0	0	0	0	0
O_3	0.3	1	0.2	1	0.2	0.6	0.6	0	0	0	0	0	0	0
O_4	1	1	0	0.2	0	0	0	0	0	0	0	0	0	0
O_5	0	1	0	0.2	0	0	0	1	0	0	0	0	0	0
O_6	0	1	0	0	0	0	0	0	1	0	0	0	0	0
O_7	0	1	0	0	0	0	0	0	0	1	0	0	0	0
O_8	0	1	0	0	0	0	0	0	0	0	1	0	0	0
O_9	0	1	0	0	0	0	0	0	0	0	0	1	0	0
O_{10}	0	1	0	0	0	0	0	0	0	0	0	1	1	0
O_{11}	0	1	0	0	0	0	0	0	0	0	0	1	0	1

Table 2. The training output samples

Output / Faults No.	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}	O_{11}
1	**1**	0	0	0	0	0	0	0	0	0	0
2	**1**	**1**	0	0	0	0	0	0	0	0	0
3	**1**	0	**1**	0	0	0	0	0	0	0	0
4	**1**	0	0	**1**	0	0	0	0	0	0	0
5	**1**	0	0	0	**1**	0	0	0	0	0	0
6	**1**	0	0	0	0	**1**	0	0	0	0	0
7	**1**	0	0	0	0	0	**1**	0	0	0	0
8	**1**	0	0	0	0	0	0	**1**	0	0	0
9	**1**	0	0	0	0	0	0	0	**1**	0	0
10	**1**	0	0	0	0	0	0	0	0	**1**	0
11	**1**	0	0	0	0	0	0	0	0	0	**1**

Table 3. Diagnosis result (the degree of certainty)

Faults / method	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}	O_{11}
NN	0.998	0.001	0.003	0.169	0.006	0.008	0.007	0.006	0.021	0.004	0.003
FNN	0.999	0.002	0.016	0.108	0.010	0.017	0.007	0.010	0.008	0.004	0.005
VFNN	0.998	0.002	0.020	0.959	0.007	0.006	0.007	0.005	0.001	0.006	0.004
DVFNN	0.996	0.001	0.018	0.953	0.006	0.010	0.010	0.010	0.027	0.006	0.008

Table 4. Diagnosis result (the degree of certainty)

faults / method	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}	O_{11}
NN	0.996	0.193	0.001	0	0.002	0.083	0.002	0.004	0.027	0.878	0.001
FNN	0.997	0.310	0.007	0.001	0.001	0.367	0	0.007	0.029	0.903	0.004
VFNN	0.998	0.208	0.010	0.003	0.019	0.623	0.001	0.001	0.008	0.876	0.002
DVFNN	0.997	0.729	0.005	0.012	0.017	0.908	0.004	0.051	0.009	0.987	0.005

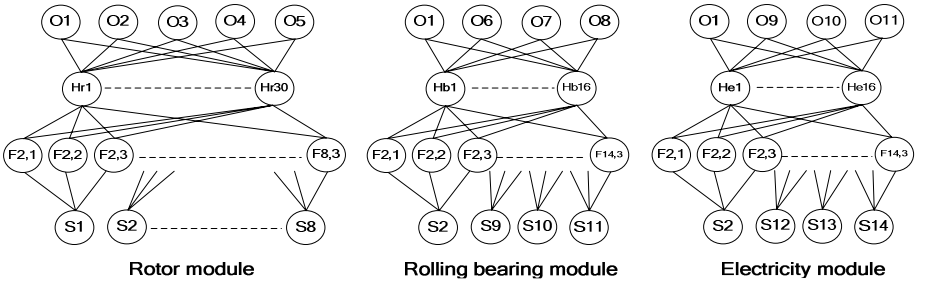


Fig. 1. The sub-networks structure of DVFNN

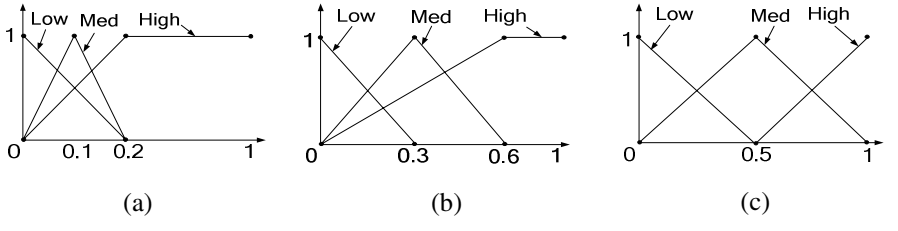


Fig. 2. Membership functions. (a) Membership function for s1, s2, s4 and s8. (b) Membership function for s3, s5, s6 and s7. (c) Membership function for s9 to s14.

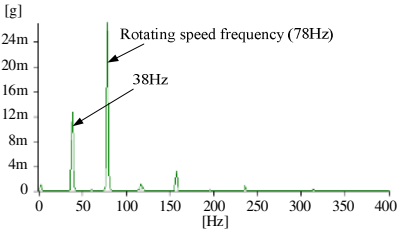


Fig. 3. Frequency diagram

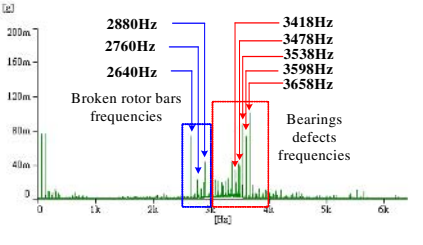


Fig. 4. Frequency diagram

Fault Detection of Reactive Ion Etching Using Time Series Neural Networks

Kyung-Han Ryu[1], Song-Jae Lee[2], Jaehyun Park[1],
Dong-Chul Park[2], and Sang J. Hong[3]

[1] Department of Electronic Engineering, Myongji University, Yongin, Korea
{kevin, jhpark}@mju.ac.kr
[2] Department of Information Engineering, Myongji University, Yongin, Korea
{songjae, parkd}@mju.ac.kr
[3] Department of Electronic Engineering & Nano-Bio Research Center,
Myongji University, Yongin, Korea
samhong@mju.ac.kr

Abstract. Maximizing the productivity in semiconductor manufacturing, early detection of process and/or equipment abnormality. Since most of the key processes in semiconductor production are performed under extremely high vacuum condition, no other action can be taken unless the undergoing process is terminated. In this paper, time series based neural networks have been employed to assist the decision for determining potential process fault in real-time. Principal component analysis (PCA) for the dimensionality reduction of the data is first performed to handle smoothly in real-time environment. According to the PCA, 11 system parameters were selected, and each of them were then classified using modeled and tested in time series. Successful detection on different types of process shift (or fault) was achieved with 0% false alarm.

1 Introduction

Issues of previous semiconductor manufacturing was mainly focused on the development of smaller and better performing microelectronics devices for supporting other engineering technology; however, it started to turn its concentration to making chips with lower manufacturing cost while maintaining what they have established in term of device performances. One way to reduce manufacturing cost in high volume manufacturing industry is minimizing process fault or at least detection of faulty process before the following processes are performed. The key success is maintaining stringent process control. When unreliable equipment leads operating conditions to vary beyond an acceptable level, overall product quality is jeopardized. Thus, timely and accurate fault detection is highly desirable for minimizing manufacturing cost [1].

RIE is a plasma etching process that involves chemical reactions and mechanical collisions of molecules in vacuum chamber. Due to the complexity of the plasma process, fault detection and classification (FDC) as a preliminary of advanced process control (APC) remains a challenge. In this study, neural networks are employed as a tool for the time series modeling of various process

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 376–381, 2006.

parameters and further utilized as a method for FDC using tool data. RIE tool data acquired from a production line consists of more than what we can handle in real time fashion. Utilizing principal component analysis, some significantly varying parameters were driven, and they were continuously used for process modeling and fault detection.

2 Fault Detection in Advanced Process Control

The identification of process excursions has traditionally been addressed by statistical process control (SPC) techniques, in which statistically established control limits define an acceptable process variation range, and process behavior beyond these limits is defined as out of control. Such shifts require that investigation and corrective action be undertaken to remove the source of variability [2]. Since classical SPC has the drawback of being incapable of detecting such shifts until the completion of process step in question, real-time SPC for plasma etching has been proposed [3].

To alleviate some of the limitations inherent in SPC, advanced process control (APC) techniques have been undertaken in the semiconductor industry. APC includes both model based process control and FDC [4]. FDC has been rigorously applied to plasma etching, utilizing tool state data [5]. and optical emission spectroscopy (OES) data as an in-situ metrology [6]. In each of these cases, principal component analysis (PCA) has been employed for the reduction of data dimensionality. Model-based PCA has also been applied to FDC in NMOS fabrication [7].

2.1 Data Acquisition

The experimental etch runs consisted of 10 baseline (nominal) runs, 3 unperturbed (acceptable) control runs, and 7 perturbed (unacceptable) runs. The perturbed runs were recorded under conditions where the chamber hardware or calibration constants were intentionally adjusted. At no time was the recipe adjusted, and none of the perturbations were sufficient to create a recipe fault.

In order to demonstrate the proposed fault detection approach, data streams from 59 system variables have simultaneously been collected at 10 Hz. Each individual run consisted of 11 different steps, including an initial stabilization step and a final de-chuck step.

Table 1. Description of 10 baseline runs, 3 control runs, and 7 perturbed runs

No	Selected parameters	No	Selected parameters
5	Pressure	21	RF probe current
7	Throttle valve percentage open	22	RF probe phase
14	Source RF match: source reading	51	RF probe peak to peak voltage
18	Source RF match: bias series	52	RF probe DC bias
19	Source RF match: bias shut	57	E-chuck voltage
20	RF probe voltage		

2.2 Principal Component Analysis

A primary challenge in this study was accounting for the voluminous multivariate tool data to establish relationships between the acquired data and the etch process state. To alleviate this concern, principal component analysis (PCA) was used as a tool to reduce the dimensionality of the data set. PCA is a well-established statistical technique that has been previously used for this purpose [8, 9].

Consider a vector $\mathbf{x}$ that consists of p random variables. Let Σ be the covariance matrix of $\mathbf{x}$. Then, for k = 1, 2,, r, the k^{th} principal component (PC) is given by $\mathbf{t}_k = \mathbf{u}_k^T\mathbf{x}$, Where $\mathbf{u}_k$ is an eigenvector of Σ corresponding to its k^{th} largest eigenvalue, and T represents the transpose operation. Dimensionality reduction through PCA is achieved by transforming the data to a new set of coordinates (i.e., the selected eigenvectors), which are uncorrelated and ordered such that the first few retain most of the variation present in the original data set. Generally, if the eigenvalues are ordered from largest to smallest, then the first few PCs will account for most of the variation in the original vector x.

3 Experiment

To enhance the performance of the proposed fault detection system, SOM first classified the input data into several groups depending on the number of steps in the baseline runs. Consequently, back-propagation neural network was employed to learn the baseline run which is represented in the form of time series.

Among the 10 baseline runs which are known for good process, empirically chosen one baseline run was used for the modeling of each parameter. Once neural network models for 11 different parameters were established, acceptable control limits for individual parameters were established. As an example of constructing process control limits of the Pressure, the Pressure data from the rest of 9 baseline runs were fed into the Pressure model as they are tested, and the maximum and minimum of the output were recorded. In this manner, the rest of parameters were repeated in the same fashion.

It seems that the testing the established network with the data from the same type of baseline runs, but these control limits will includes random variability of the etching process. Even though all the baseline runs can be regarded as the same, actually they are not the identical, but they all acceptable for the baseline. The difficulty of fault detection and classification of RIE comes from the avoidable random variability associated with chemical and mechanical process of molecules inside the chamber. In SPC, control limits can be calculated utilizing $\pm 3\sigma$ (standard deviation) criteria and it has been widely used in semiconductor manufacturing over the past decades. The process control limits presented in this paper seems rather empirical than statistical way, but this approach may useful for the following cases, which are prevail semiconductor process. At the transition point, great amount of sudden peaks of responses are very usual due to transition of tool parametric value. However, empirically established control limits are rather smoothing the peaks at the step to step transition which were

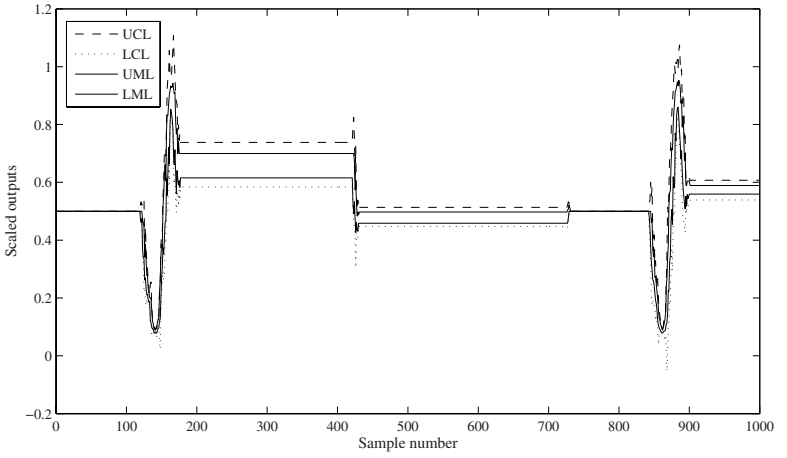


Fig. 1. Comparison between statistically established $\pm 3\sigma$ process control limits and empirically found baseline process model. UCL and LCL denote upper control limits for process control limits, and UML and LML stands for upper model limits and lower model limits.

known for disadvantage of neural networks. Right after the transition, there are generally large amount of innate random variability exist until the parametric values approach to the desired level. In this case, neural network would incorporate the random variability and showed empirically outperforming result than statistical way even though the full proof version of this is still remained as a future work.

4 Result

Once appropriate control limits of all 11 parameters were successfully established, the remaining 10 experimental runs were further tested for fault detection of reactive ion etching. The testing was performed on the run-to-run basis rather than in real time, and the simulation results are presented in this section. Fig. 1 show two types of control limits established by statistical $\pm 3\sigma$ approach and neural time series model respectively. As it is depicted in Fig. 1, statistically established control limits are wider than empirical model limits, which can be translated that statistical limit can increase the chance of missed alarm. On the other hands, empirical control limits is rather provide tighter control limit, thus it may increase the false alarm. However, the Type I can be reduced by multiplying appropriately weighting factor so as to modify the control limits depending on the process engineer.

As it is pointed out earlier, empirical model limits are meaningful for peaks at the step to step interface. In this type of work, very high peaks at the step to step interface is inevitable, thus they have been either neglect or major factor for false alarm. Detailed model verification result is provided in Table 2. In the

Table 2. Description of 10 baseline runs, 3 control runs, and 7 perturbed runs

Parameters	1	2	3	4	5	6	7	8	9	10
Pressure	◯	⊕	◯	◯	◯	◯	◯	◯	◯	◯
Throttle valve open	◯	⊕	⊕	◯	◯	◯	◯	◯	◯	◯
RF source reading	◯	◯	◯	◯	◯	⊕	◯	◯	◯	◯
RF bias series	◯	◯	◯	◯	⊕	⊕	◯	TypeII	◯	⊕
RF bias shut	◯	◯	◯	TypeII	TypeII	⊕	◯	⊕	◯	⊕
RF probe voltage	◯	◯	◯	◯	◯	⊕	◯	⊕	◯	⊕
RF probe current	◯	◯	◯	◯	◯	⊕	◯	◯	◯	⊕
RF probe phase	◯	◯	◯	◯	◯	⊕	◯	⊕	◯	⊕
RF probe voltage	◯	◯	◯	◯	◯	⊕	◯	⊕	◯	⊕
RF probe DC bias	◯	◯	◯	◯	◯	⊕	◯	⊕	◯	⊕
E-chuck voltage	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯

table, runs without fault is presented with open circle (◯: normal), and runs with fault with closed circle(⊕: fault). All of ◯ and ⊕ represents the correction action, in other words, the fault detector classified normal operations as normal and fault operations as faults. Other than symbols, "TypeII" shows missed alarm in quality control. Type I error (or false alarm) is a type of error that the fault detector detected fault even though there was no fault. Type II error (or missed alarm) is induced by not detecting faults when there exist faults. In this exercise, no evidence of a fault in the three controlled runs (1, 7, and 9) was found, which indicates zero false alarms. For the runs with 4, 5, and 8, Type II errors were found; however, only run number 4 is accounted for actual missed alarm since the rest of runs were already found to be fault in other parameters.

Tracking the experimental condition simulating the fault scenario, run number 4 and 5 are only $\pm 0.1\%$ of mass flow controller fault. Considering the fact that about %5 of system fluctuation is innate to normal equipment operation, only $\pm 0.1\%$ of fault can be regarded as no fault, and the missed alarm generated in those runs are not very surprising.

5 Conclusion

Considering productivity in high volume semiconductor manufacturing, detection of abnormal process is a crucial to minimize the cost from scrap. To show the potential of neural networks for efficient fault detector for fault detection and classification (FDC) module in advanced process control (APC). Reactive Ion Etching (RIE) tool data acquired from a production line consist of 59 variables, and each of them collected 10 data points per second. Process model limits, represented by upper/lower model limits, were proposed, and fault detection on RIE tool parameters were performed according to the proposed process model limits. According to the 10 testing runs, 0% of false alarm was achieved, and there were only one experimental run that contribute to missed alarm. Currently, refining of process model with more evident faulty data is under development, and

multivariate modeling is also valuable to improve the performance of the fault detector.

Acknowledgement

This work was supported by 2005 Research Fund of Myongji University (PID: 20051124), and authors are grateful to members at Intelligent Micro/Nano Process Lab. for sharing their knowledge for fault detection.

References

1. Hong, S., May, G., Park, D.: Neural Network Modeling of Reactive Ion Etch Using Optical Emission Spectroscopy Data. IEEE Trans. Semi. Manufac. 16(4) (2003) 1-11
2. Montgomery, D.: Introduction to Statistical Quality Control. Wiley, New York NY (1991)
3. Guo, H., Spanos, C., Miller, A.: Real Time Statistical Process Control for Plasma Etching. In: Semiconductor Manufacturing Science Symposium. (1991) 113-118
4. Barna, G.: APC in the Semiconductor Industry, History and Near Term Prognosis: In Proc. IEEE/SEMI Adv. Semi. Manufac. Conf. (1996) 364-369
5. Goodlin, B., Boning, D., Sawin, H.: Simultaneous Fault Detection and Classification for Semiconductor Manufacturing Tools: In Proc. Int. Symposium on Plasma Processing XIV. (2002)
6. Yue, H., Qin, s., Markle, R., Nauert, C., Gatto, M.: Fault Detection of Plasma Etchers Using Optical Emission Spectra, IEEE Trans. Semi. Manufac. 13(3) (2000)
7. Lachman-Shalem, S., Haimovitch, N. Shauly, E. Lewin, D.: MBPCA Application for Fault Detection in NMOS Fabrication. IEEE Trans. Semi. Manufac. 15(1) (2002) 60-69
8. Haykin, S.: Neural Networks. Macmillan College Publishing Company. New York NY, (1994)
9. Anand, R. Mehrotra, k., Mohan, c. Ranka, s.: Efficient Classification for Multiclass Problems Using Modular Neural Networks. IEEE Trans. Neural Networks. 6(1) (1995) 117-124

Intelligent Diagnostics for Sound Reproduction System by the Use of PEAQ

Byung Doo Jun[1], Nakjin Choi[2], Hyun-Woo Ko[2], and KoengMo Sung[2]

[1] R&D Lab., NEX1 FUTURE Co., Ltd., 148-1 Mabuk Gusung,
Yongin Kyunggi, Korea, 449-910
bdjeon@nex1.co.kr
[2] School of Electrical Engineering and Computer Science, Seoul National University,
Shinlim-dong, Kwanak-gu, Seoul, Korea, 151-742
{nakjin, kenshin, kmsung}@acoustics.snu.ac.kr

Abstract. After finishing the design of the sound reproduction system, it is necessary and important to evaluate how much each component of the sound reproduction system affects the final quality of audio. In this paper, we present an intelligent diagnostics for sound reproduction system. It is based on PEAQ(ITU-R BS. 1387) and artificial neural networks. This method makes it possible to consider the human preferences for various genres of music which differ from nation to nation and culture to culture.

1 Introduction

Sound reproduction system generally consists of source(e.g., CD player), amplifier system, and loudspeakers. It is necessary and important to audio engineers to know how much each component of the system affects the final quality of sound. The reason is that the final sound quality is determined by the contribution of each component [1].

There are two ways to evaluate the quality of sound reproduction system. One way is a subjective assessment in which listeners subjectively evaluate the sound quality of a system. The other way is an objective assessment in which the sound quality is evaluated from the measured data.

It is well known that there are various kinds of subjective listening test methods and associated standards in existence. Single-stimulus, pair-comparison, ABX, and triple stimulus with hidden reference - have been widely used and still popular in subjective assessment [2][3]. However, none of them are infallible. Therefore, a new methodology called MUSHRA (MUlti Stimulus with Hidden Reference and hidden Anchor) is seriously considered as an alternative [4].

However, Human ears surpass any other measurement system in ability. However, the subjective evaluation by human ear has some disadvantages. For example, the results of evaluation change according to listener's feeling from day to day. And listeners can lose their keen sense of hearing with growing age. In addition, listening testers are required considerable amount of training for reliable test results. If it is possible to assess loudspeakers with fidelity from the measured data by computational method, then we can save both time and cost.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 382–389, 2006.

In the late of 1990s, a new perceptual measurement called PEAQ(Perceptual Evaluation of Audio Quality) in ITU-R is presented for objective measurement of perceived audio quality(BS. 1387) [5]. PEAQ is based on generally accepted psychoacoustic principles. It includes measures of nonlinear distortions, linear distortions, harmonic structure, distance to masked threshold, and changes in modulation. These variables are mapped to a single measure of audio quality by a neural network [6].

In this paper, we present an intelligent diagnostics for sound reproduction system by the use of PEAQ. Single-layer perceptrons are used to reflect the consumers' preferences for various genres of music which differ from nation to nation and culture to culture.

In Section 2, we will explain the PEAQ(ITU-R BS.1387) and in Section 3, propose an intelligent diagnostics for sound reproduction system. In Section 4, the results and analysis of our experiment are present and in Section 5 are present our conclusions.

2 ITU-R: Perceptual Evaluation of Audio Quality (PEAQ)

From 1994 to 1996, ITU-R evaluated several existing methods for objective measurement of perceived audio quality. These methods were called DIX, NMR, OASE, PAQM, PERCEVAL, POM, and Toolbox. The conclusion by the ITU-R committee that none of the existing perceptual measurement methods was sufficiently reliable for an international standard led to the cooperative development of PEAQ. PEAQ is a new perceptual measurement scheme, jointly developed by all parties that were involved in the development of measurement methods and combines features of all these methods [5][6]. PEAQ includes ear models based on the fast Fourier transform (FFT) as well as on a filter bank. The model output values are partly on the masked threshold concept and partly on a comparison of internal representations. In addition, it also yields output values based on a comparison of linear spectra not processed by an ear model. The model outputs the partial loudness of nonlinear distortions, the partial loudness of linear distortions (signal components lost due to an unbalanced frequency response), a noise to mask ratio, measures of alterations of temporal envelopes, a measure of harmonics in the error signal, a probability of error detection, and the proportion of signal frames containing audible distortions.

Selected output values are mapped to a single quality indicator by an artificial neural network with one hidden layer.

The model consists of two versions: one is intended for applications that require high processing speed (that is, low computational complexity) and is called the basic version of PEAQ. The other version is intended for applications requiring the highest achievable accuracy and is called the advanced version. Both versions generate many of the same types of MOVs, but the advanced version takes advantage of the increased temporal resolution of the filter bank ear model. Fig. 1 shows a schematic of the overall model.

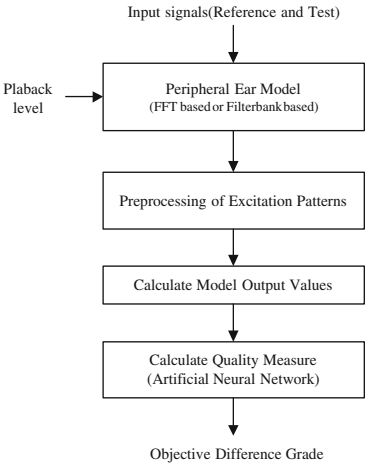


Fig. 1. Block diagram of measurement scheme

The Advanced Version uses both the filter bank-based ear model and the FFT-based ear model. It uses the Model Output Variables $\text{RmsModDiff}_\text{A}$, $\text{Rms NoiseLoudAsym}_\text{A}$, $\text{AvgLinDist}_\text{A}$, $\text{SegmentalNMR}_\text{B}$, and EHS_B. These five Model Output Variables are mapped to a single quality index using a neural network with five nodes in the hidden layer. The parameters of the mapping are given in Table 1.

Table 1. Model output variables used in the advanced version

MOV	Purpose
$\text{RmsNoiseLoudAsym}_\text{A}$	Loudness of the distortion
$\text{RmsModDiff}_\text{A}$	Changes in modulation (related to roughness)
$\text{AvgLinDist}_\text{A}$	Linear distortions (frequency response etc.)
Segmental NMR_B	Noise-to-mask ratio
EHS_B	Harmonic structure of the error

3 Proposed Intelligent Diagnostics for Sound Reproduction System

3.1 Model for Sound Reproduction System

Sound reproduction system generally consists of source equipment, amplifier, and loudspeaker. Fig. 2 shows the block diagram of sound reproduction system. Continuous electrical signal $x(t)$ from source equipment(e.g., CD player) grows up after passing through the amplifier. The amplified electrical signal $y(t)$ is converted to an acoustic signal $z(t)$ by loudspeaker.

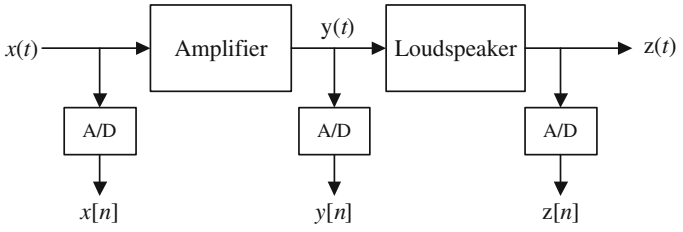


Fig. 2. Sound reproduction system model

In this paper, we assume that the purpose of sound reproduction system is to reproduce the small electrical signal $x(t)$ to the acoustic signal $z(t)$ without any distortion. Signals $x(t)$, $y(t)$, and $z(t)$ should be sampled and quantized to measure the performance of each component - amplifier, loudspeaker - in digital domain.

3.2 The Basic Idea of the Proposed Method

The basic concept for making objective measurements with PEAQ is illustrated in Fig. 3 below. The output of PEAQ is ODG(Objective Difference Grade) which means how much the test signal $s_{test}[n]$ resembles the reference signal $s_{ref}[n]$. Fig. 4 describes the basic idea of the proposed method. In this figure, $\mathrm{ODG}_{x,y}$ means how much the input signal $x[n]$ of the amplifier resembles the output signal $y[n]$ of the amplifier, that is, how much the amplifier degrades the input signal $x[n]$. That is,

$$\mathrm{SQ}_{\mathrm{Amplifier}} = \mathrm{ODG}_{\mathrm{x,y}} \tag{1}$$

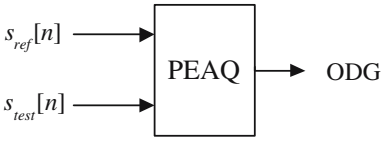


Fig. 3. The basic concept of PEAQ

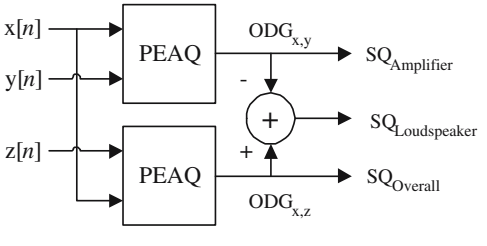


Fig. 4. The basic idea of the proposed method

When we test the performance of the loudspeaker, $\mathrm{ODG}_{x,z} - \mathrm{ODG}_{x,y}$ is more meaningful than $\mathrm{ODG}_{y,z}$ only. It is because $\mathrm{ODG}_{x,z} - \mathrm{ODG}_{x,y}$ means how much the original signal $x[n]$ is degraded by passing through the loudspeaker only. That is,

$$\mathrm{SQ}_{\mathrm{Loudspeaker}} = \mathrm{ODG}_{\mathrm{x,z}} - \mathrm{ODG}_{\mathrm{x,y}}. \tag{2}$$

3.3 Proposed Intelligent Diagnostics for Sound Reproduction System

It is not desirable to assess the sound reproduction system by means of only single sound source like in Fig. 4. It is because the intelligent diagnostics should evaluate the sound quality under various consumer favorite genres of music.

Fig. 5 describes the block diagram of the proposed method. The first ODGs are calculated from the $x_i[n]$, $y_i[n]$, and $z_i[n]$. After that, the sound quality of each component is evaluated by passing the ODGs through the second-stage artificial neural networks. These neural networks will be explained in Section 3.4.

3.4 The Second-Stage Artificial Neural Networks

Artificial neural network in Fig. 5 consists of single-layer perceptron as in Fig. 6. The synaptic weight w_i of this perceptron represents the people's preference for the i-th sound source. These preferences are determined by the statistical research like enquête. They differ from nation to nation and culture to culture.

4 Experimental Results

In this experiment, there were six sound sources. Each sound source has its unique perceptual dimensions. Table 2 is the match-up list of the sound source and the perceptual dimensions. In this experiment, we assume that the relationship between input $x(t)$ and output $y(t)$ of the amplifier system is

$$y(t) = \tanh(3.5x(t)) \tag{3}$$

and the loudspeaker system is LTI(Linear Time-Invariant) system with following impulse response.

$$h(t) = \delta(t) - 0.2\delta(t-1) \tag{4}$$

ODGs and MOVs between $x(t)$ and $y(t)$ are summarized in Table 3. In the same way, those for $x(t)$ and $z(t)$ are summarized in Table 4. Fig. 7 shows the sound quality of each component.

It is shown that the amplifier and the loudspeaker degrade the sound quality by percentages of 88 and 87, respectively. Eventually, the overall system degrades the original sound source by a percentage of 80.

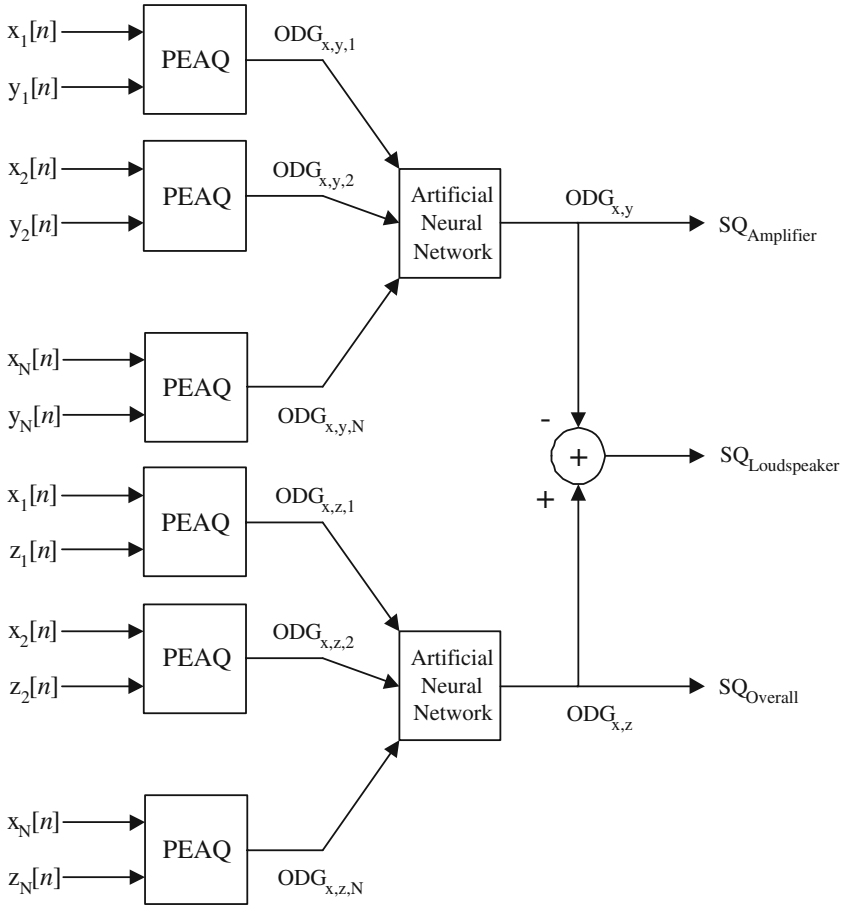


Fig. 5. Block diagram of the proposed method

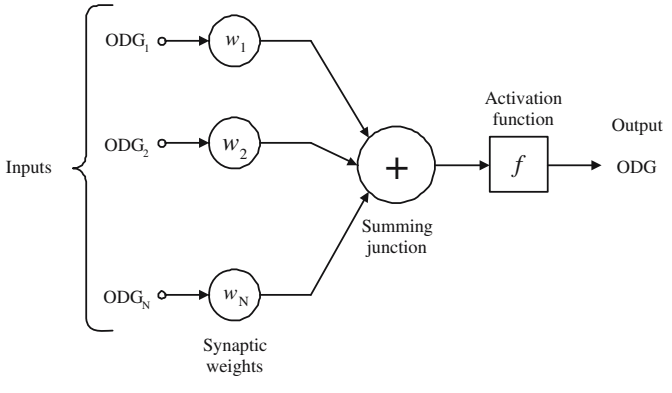


Fig. 6. Single-layer perceptron

Table 2. Model output variables used in the advanced version

No.	Sound source	Perceptual dimension
1	Spanish Harlem	Transparency, Diffuseness, Fullness
2	Dunk Shot	Extraneous sounds, Fullness, Sharpness
3	Enter Sandman	Brightness, Fullness, Sharpness
4	Drum	Depth, Diffuseness, Brightness
5	Zarathustra	Dynamic Balance, Depth, Brightness
6	Viola fora de moda	Transparency, Extraneous sounds, Diffuseness

Table 3. ODGs and MOVs between $x(t)$ and $y(t)$

No.	NLA	RModDiff	ALD	SNMR	EHS	ODG
1	1.25049	98.0865	0.29362	-16.541	0.413336	-0.536812
2	0.69694	92.6141	0.31791	-16.168	0.199392	-0.376001
3	1.52799	119.189	1.68246	-15.504	0.210566	-0.854159
4	0.90336	28.5959	0.34827	-19.794	0.212564	-0.008974
5	0.41706	56.7729	0.68792	-22.769	0.187879	-0.109033
6	0.33292	41.2663	0.15278	-19.790	0.253626	-0.052436

Table 4. ODGs and MOVs between $x(t)$ and $z(t)$

No.	NLA	RModDiff	ALD	SNMR	EHS	ODG
1	1.27058	103.825	0.14585	-11.613	0.536768	-1.13534
2	0.73337	100.819	0.17762	-11.885	0.248275	-0.73305
3	1.55321	125.549	1.25814	-12.279	0.293215	-1.17486
4	0.94303	34.6778	0.29819	-12.437	0.072916	-0.24344
5	0.43521	66.1287	0.57509	-16.018	0.289308	-0.24627
6	0.35007	47.8384	0.06706	-12.412	0.338554	-0.35491

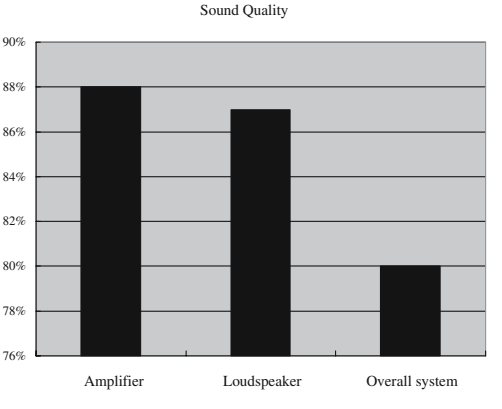


Fig. 7. Sound quality of each component

5 Conclusions

In this paper, an intelligent diagnostics for sound reproduction system is presented. This method evaluates how much each component of the system affects the final quality of audio. And artificial neural networks with single-layer perceptron are used to consider the human preferences for various genres of music. In addition, we can save both time and cost with this method, because we do not need any subjective assessment by human.

References

1. Harley, R: The Complete Guide to High-End Audio, 3rd Ed., Acapella Publishing (2004)
2. Eisler, H.: Measurement of Perceived Acoustic Quality of Sound Reproducing Systems by means of Factor Analysis. J. Acoust. Soc. Am. 39 (1966) 484-492
3. ITU-R Rec. BS. 1116-1: Method for the Subjective Assessment of Small Impairments in Audio Systems including Multichannel Sound Systems. International Telecommunications Union, Geneva, Switzerland (1994)
4. ITU-R Rec. BS. 1534-1: Method for the Subjective Assessment of Intermediate Quality Level of Coding Systems, International Telecommunications Union, Geneva, Switzerland (2001)
5. ITU-R Rec. BS. 1387: Methods for Objective Measurements of Perceived Audio Quality, International Telecommunications Union, Geneva, Switzerland (1998)
6. Thiede, T., et al.: PEAQ - The ITU Standard for Objective Measurement of Perceived Audio Quality. J. Audio Eng. Soc. 48(1/2) (2000) 3-29

Active Learning of Support Vector Machine for Fault Diagnosis of Bearings

Zhousuo Zhang, Wenzhi Lv, and Minghui Shen

School of Mechanical Engineering,
State Key Laboratory for Manufacturing Systems Engineering,
Xi'an Jiaotong University, Xi'an 710049, China
zzs@mail.xjtu.edu.cn, lvwenzhi@stu.xjtu.edu.cn,
smhui@stu.xjtu.edu.cn

Abstract. Based on traditional Active Support Vector Machine (ASVM), the learning method of Probabilistic Active SVM (ProASVM) is introduced to detect fault of bearings. Compared with the general SVM, the active learning methods can effectively reduce the number of samples on the condition of keeping the classification accuracy. ASVM actively selects data points closest to the current separation hyperplane, while ProASVM selects the points according to the probability of the sample point as a support vector. The two methods are applied to classify the practical vibration signal of bearings and the results show that ProASVM is a better algorithm of classification than ASVM.

1 Introduction

In recent years, support vector machine (SVM) [1] has demonstrated excellent performance in a variety of fields, such as handwriting recognition, face detection, gene classifications and image classification.

SVM use a convex quadratic optimization (QP) algorithm to find unique globally optimal separation hyperplane. The separation hyperplane can be represented by a small subset of training data lying on the margin. These samples, known as support vectors, contain the relevant information of the classification. The other samples can be removed from the training samples, and the same results can be got through recalculating the separation hyperplane. This makes active learning of SVM become in practice. As we know, obtaining classification labels is difficult and expensive in fault diagnosis of bearings, while large quantities of unlabeled samples are readily available. Because the quantity of labeled samples is relatively small, they cannot describe the whole samples distribution well. Active learning of SVM is a good solution for this condition, because active learning of SVM not only takes advantage of the labeled samples but also the unlabeled samples.

In active learning, instead of learning from "random samples", the learner has the ability to select training samples [2]. Active learning is an iterative process in form. First, the learner selects some samples according to prior knowledge or randomly does, and labels them as the initial training set. The training set must contain different

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 390 – 395, 2006.

classes. Second, a learner trains a classifier based on current labeled samples. Then based on the current state of the classifier, the learner selects some of the "most informative" samples from the unlabeled ones and labels them, so that a new classifier can be trained based on the selected samples and the former samples. Obviously, this greatly accelerates the learning compared to random sampling. Finally, the learner selects some of the "most informative" samples again, repeats the above processes until all samples have been labeled or the classifier has already satisfied a certain target.

This paper is organized as fellows: in section 2 we briefly review the theory of SVM, which provides a fundamental for this paper. In section 3 we describe two active learning methods of SVM: Active SVM and Probabilistic Active SVM. The two methods are applied to fault diagnosis of bearings and the results are provided in section 4. We finally end this paper by conclusions in section 5.

2 Support Vector Machines

In this section, we briefly review the SVMs in binary classification problems. Let the training set $D = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$, with each input $x_i \in R^m$, and the output label $y_i \in \{-1, +1\}$. The SVM first maps x to $z = \phi(x)$ from the input space to the feature space. Usually, the dimensionality of the feature space is very high. Consider the case that the data is linearly separable in feature space, there exists a vector w and a scalar b such that

$$y_i(\langle w, \phi(x_i)\rangle + b) \geq 1 \tag{1}$$

for all the training set, where $\langle w, \phi(x)\rangle$ represents the inner product of w and $\phi(x)$ .Then the SVM constructs a hyperplane $(\langle w, \phi(x)\rangle + b)$, so that the separation between the positive and negative data is maximized. The hyperplane defined above can be interpreted as a decision boundary, therefore the sign of

$$f(x) = \langle w, \phi(x)\rangle + b \tag{2}$$

gives the predicted label of input x .

In many practical situations, a linear separating hyperplane does not exist. To allow for possibilities of violating Eq. (1), slack variables $\xi_i \geq 0$ are introduced to get

$$y_i(\langle w, \phi(x_i)\rangle + b) \geq 1 - \xi_i, i = 1, \ldots, m \tag{3}$$

The support vector approach for minimizing the generalization error consists of the following:

$$\text{Minimize } \phi(w, \xi) = \langle w, w\rangle + C_p \sum_{i=1}^{m} \xi_i \text{ , subject to constraints } \xi_i \geq 0 \text{ and (3)} \tag{4}$$

Here, C_p is a regularization parameter controlling the tradeoff between model complexities and training error, in order to ensure good generalization performance.

As the dimension of feature space are very high, it is very difficult to obtain w or $\phi(x)$ explicitly. For this approach being practical, the kernel function, as a key characteristic of the SVM, has been introduced. This provides a way of dealing with the curse of dimensionality. An inner product in feature space has an equivalent kernel in input space, $K(x, x') = \langle \phi(x), \phi(x') \rangle$, provided certain conditions satisfied.

3 Active Learning of SVM

The key of active learning is how to select the "most informative" samples. For linear support vectors machine, when the unlabeled samples within the margin are selected, the separation hyperplane of the classifier changes most possibly. And the data outside of the margin have very little influence on the classifier. So one of the typical active learning methods, called ASVM, picks up the unlabeled samples lying closest to the current separation hyperplane [3] and does not consider the underlying distribution of unlabeled data. However, the current separation hyperplane itself may be far from the optimal hyperplane. As a result, the selected samples are less likely to be the actual support vectors. So a better method has been put forward, which is called as ProASVM. It selects the points according to the probability of sample x as a support vector [2].

3.1 Probabilistic Active SVM (ProASVM)

In this algorithm [2], the "most informative" samples are selected according to a probability. The probability is defined as fellows:

$$p_r = \begin{cases} c & y_i(\langle w, \phi(x_i) \rangle + b) \le 1 \\ 1-c & \text{otherwise.} \end{cases} \tag{5}$$

Here c is a confidence factor, which denotes how close the current separating hyperplane $\langle w, b \rangle$ is to the optimal one. y is the label of x.

The meaning of P_r is as follows: when c is high, the current hyperplane is close to the optimal one, data points with margin value less than 1 are highly likely to be the actual SVs. When the value c is low, the probability of selecting a point lying within the margin decreases, and a high probability value $(1-c)$ is then assigned to a point having high margin value. The method to estimate the confidence factor c is following.

$S_t = \{s_1, s_2, \ldots, s_n\}$ is the support vector set obtained after tth iteration, and $B = \{x_1, x_2, \ldots, x_{m2}\}$ denotes the samples whose class labels are unknown. For every $s_i \in S_t$, k points in B nearest to s_i are computed. Among the k nearest points, let k_i^+ points and k_i^- points respectively have labels "+1" and labels "-1". The confidence factor c is then defined as

$$c = \frac{2}{nk} \sum_{i=1}^{n} \min(k_i^+, k_i^-). \tag{6}$$

The maximum value of the confidence factor c is 1 when $k_i^+ = k_i^-$, $\forall$ i=1,… ,n., and the minimum value is zero when $\min(k_i^+, k_i^-) = 0$ $\forall$ i=1,…,n .The first case denotes that all the support vectors lie near the class boundaries and the set $S_t = \{s_1, s_2, \ldots, s_n\}$ is close to the actual support vector set. And the second case implies that the set of S_t consists only of interior points and is far from the actual support vector set. Thus, the confidence factor c measures the degree of S_t close to the actual support vector set.

The basic work of ProASVM is finished, and the algorithm is following. Let $X = \{x_1, x_2, \ldots, x_m\}$ denote the entire training set for active SVM. $SV(E)$ denotes the set of support vectors of the set E obtained using SVM. $S_t = \{s_1, s_2, \ldots, s_n\}$ is the support vector set obtained after ***t***th iteration, and $\langle w_t, b_t \rangle$ is the corresponding separating hyperplane. D_t is a sample set actively selected in ***t***th iteration and consists of ***m*** points. c is the confidence factor.

Initial step: Randomly select an initial training set D_0 from the training set X. $B = \{x_1, x_2, \ldots, x_{m2}\}$ denote the unselected data for which class labels are unknown. Set $t = 0$ and $S_0 = SV(D_0)$.

```
While stopping criterion is no satisfied:
  D_t =0.
  While cardinality (D_t)≤m:
    Randomly select a sample x∈B, let y be the label of x.
    If y(⟨w_t, x⟩ + b) ≤ 1:
      Select x with probability c. Set D_t= D_t∪x.
    Else
      Select x with probability1- c. Set D_t= D_t∪x.
    End if
  End while
  S_t=SV(S_t∪D_t); t=t+1.
End while
```

The set S_T, where T is the iteration at which the algorithm terminates, contains the final SV set.

Stopping criterions: Among the m points actively queried at each step ***t***, let $\boldsymbol{m_t}$ points have margin greater than unity ($y(\langle w_t, x \rangle + b) > 1$). Learning is stopped if $\boldsymbol{(c^* m_t/m)}$ exceeds a threshold $\mathbf{h_t}$. The threshold, $0<\mathbf{h_t}<1$, depends on the performance of classification.

4 Applications of Active Learning in Bearings Fault Diagnosis

The two methods mentioned above have been successfully applied to the fault diagnosis of bearings. Figure 1 shows the vibration signals of bearings [4], and operating frequency of the bearings is about 29.2Hz. Signal sampling frequency is 12KHz; the data sample includes 1024 points. The normal and fault samples are treated as training and testing samples for active learning.

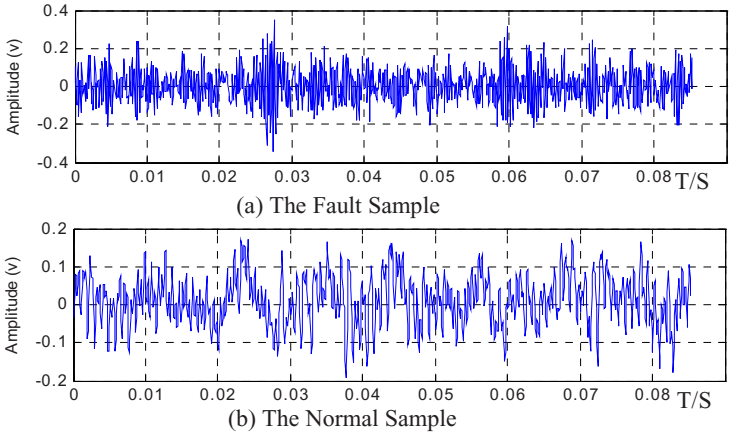


Fig. 1. The waveform of data samples in time domain

There are 210 samples in the training set X. When we classify the X by means of the general SVM, we must spend much time to label all the samples as training set. The execution time of the general SVM is 81.2 seconds, and classification accuracy is 96%. All the programs run on a computer with a 600 MHz Intel Pentium Ⅲ Processor.

Table 1. comparison of the two active learning algorithms, ker='rbf', C_p=100

Initial data	Algorithm	Classification Accuracy (%)	Execution time (Sec)
RandomX1	ASVM	83	184.4
	ProASVM	92	138.7
RandomX2	ASVM	70	173.9
	ProASVM	91	135.8
RandomX3	ASVM	85	177.2
	ProASVM	94	115.8
RandomX4	ASVM	90	174
	ProASVM	92	110.9
Average	ASVM	82	177.4
	ProASVM	92.25	122.2

In order to compare the performances of the two active learning, we carry out four times of experiments, and the results are presented in table 1. For every experiment, we randomly select 30 samples from X as initial training set and label them. Class labels of the remaining points are unknown.

Tab.1 shows the comparison of the two active learning algorithms in terms of classification accuracy and execution time. The classification accuracy of ProASVM ranges between 91% and 94%, while that of ASVM changes greatly from 70% and 90%. On average, the former 92.25% is higher than the latter 82%. And the execution time of ASVM varies acutely from 173.9 to 184.4 seconds and that of ProASVM is from 110.9 to 138.7 seconds. Tab.1 also shows that the algorithm of ProASVM has less dependence on initial training set than ASVM.

5 Conclusions

Generally, the performance of Probabilistic Active SVM (ProASVM) is better than that of traditional Active SVM(ASVM). Initial training set influences the result of ASVM greatly. Therefore, in practical application, if the initial training set is not proper, classification accuracy will be lower and execution time will be longer; On the other hand, ProASVM not only selects the points close to the separation hyperplane, but also some other points far from the separation hyperplane. Thus, even if the initial separation hyperplane is improper, the separation hyperplane can be corrected by the later points. So, in practical applications such as bearings fault diagnosis, the ProASVM outperforms than ASVM.

Acknowledgements

This work was supported by the project of National Basic Research Program of China (No. 2005CB724100), National Natural Science Foundation of China (No.50335030, No.50175087).

References

1. Vapnik, V.: Statistical Learning Theory. New York: Wiley (1998)
2. Mitra, P., Murthy, C. A., Pal, S. K.: A Probabilistic Active Support Vector Learning Algorithm, IEEE Transactions on Pattern Analysis and Machine Intelligence 26(3), (2004) 413-418
3. Mitra, P., Murthy, C. A., Pal, S.K.: Data Condensation in Large Databases by Incremental Learning with Support Vector Machines. In: Proceedings 15th International Conference on Pattern Recognition. (ICPR2000), Barcelona, Spain (2000) 712-715
4. Li, L., Zhang, Z.: Research of Mechanical System Fault Diagnosis Based on Support Vector Data Description, Journal of Xi'an Jiaotong Unversity 37(9) (2003)

Growing Structure Multiple Model System Based Anomaly Detection for Crankshaft Monitoring

Jianbo Liu[1], Pu Sun[2], Dragan Djurdjanovic[1], Kenneth Marko[2], and Jun Ni[1]

[1] University of Michigan, Ann Arbor, MI 48109, USA
{jianbol, ddjurdja, junni}@umich.edu
[2] ETAS Inc., Ann Arbor, MI 48103, USA
{Pu.Sun, Ken.Marko}@etas.us

Abstract. While conventional approaches to diagnostics focus on detecting and identifying situations or behaviors which have previously been known to occur or can be anticipated, anomaly detection focuses on detecting and quantifying deviations away from learned "normal" behavior. A new anomaly detection scheme based on Growing Structure Multiple Model System(GSMMS) is utilized in this paper to detect and quantify the effects of slowly evolving anomalies on the crankshaft dynamics in a internal combustion engine. The Voronoi sets defined by the reference vectors of the growing Self-Organizing Networks(SONs), on which the GSMMS is based, naturally form a partition of the system operation space. Regionalization of system operation space using SONs makes it possible to model the system dynamics locally using simple models. In addition, the residual errors can be analyzed locally to accommodate unequally distributed residual errors in different regions.

1 Introduction

Anomaly detection can be described as identification for abnormal system behavior that has not been observed before. The anomaly detection is analogous to one of the important functionalities of the immune system in human body. Before taking any actions to eliminate the foreign invaders, the immune system first has to correctly identify anything harmful that is not "self", even though the virus or bacteria have never been encountered by the immune system. This capability of being able to identify non-self is vital for the survival of living organisms since the types of infections are very diverse and unpredictable.

While conventional approaches to diagnostics focus on detecting and identifying situations or behaviors which have previously been known to occur or can be anticipated, anomaly detection focuses on detecting and quantifying deviations away from learned "normal" behavior. Identifying "non-self" and quantifying the dissimilarity require an accurate definition of normal behavior. For anomaly detection, constructing a profile representing normal behavior is essential for the success of the anomaly detection system. Once an accurate model is obtained,

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 396–405, 2006.

the discrepancies between the model and the real system then can be utilized to sense and quantify the deviations.

This report is organized as follows: In Sect. 2, some background about the GSMMS will be provided; Detailed information of the modeling and simulation of crankshaft dynamics based on first principle will be given in Sect. 3; A model is constructed in Sect. 4 using GSMMS for the purpose of anomaly detection and fault diagnosis; In Sect. 5, a region dependent decision making scheme is utilized to detect the friction changes based on the model obtained in Sect. 4. Section 6 gives the conclusions and future work.

2 Background

Instead of using global modeling approaches, such as neural networks, an alternative for modeling nonlinear dynamic system is the "divide-and-conquer" approach, which divides the whole system operation space into sub-regions and model the local dynamics individually within a small sub-region. The heuristics is that the modeling task for a small sub-region of the system behavior becomes much easier to deal with than modeling the system as a whole. By dividing the operational space into small regions, the multiple local model system can provide, although limited in general, transparency about the physical processes.

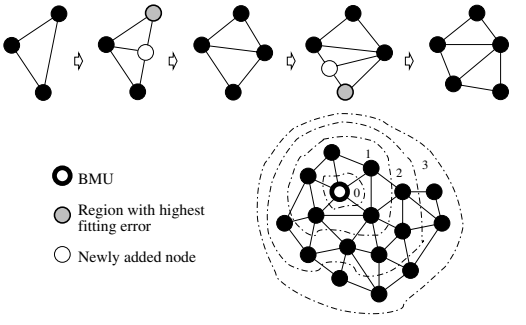


Fig. 1. Overview of the training procedure for growing structure multiple model system

GSMMS combines the advantages of the growing SONs [1][2] with efficient linear model parameter estimation algorithms [3] into an integrated framework for modeling and identification of general nonlinear dynamic systems. GSMMS can be implemented on-line to learn the structural parameters and the local model parameters simultaneously. Figure 1 shows the overview of the training procedure for GSMMS. We start from relatively small number of nodes (or equivalently local models) and then gradually add nodes to the ordered network where the fitting error is high. The training procedure will stop once the stopping criteria are satisfied. In this way, the number of nodes can be determined in a automatic fashion.

The updating procedure for the local model parameters consists of following steps. First we have to identify the Best Matching Unit(BMU) and update the local model associated with it. Then the local models associated with the neighbors of the BMU will also be updated and the updating rates are determined by their memberships with respect to the BMU. In general, the values of the memberships are inversely proportional to the shortest distance between the neighboring nodes and the BMU on the networks. The further away the node from the BMU, the smaller the updating rate. Figure 1 also shows schematically how to determine the membership of a node with respect to the BMU. The shortest distance between the neighboring nodes and the BMU can be calculated efficiently using Breath-first algorithm from the adjacency matrix [4].

Moreover, a nonlinearity measure is proposed in [5], which is based on the fitting errors from the local models, to adjust the location of each region by modifying the learning rate in the standard weight vector updating formula [6] of the SONs. Each Voronoi set defined by the weight vectors of the resulting SONs essentially forms a system operational region. Introduction of the nonlinearity measure has the effect of moving weight vectors in the network towards the region where the system nonlinearity is high. For more detailed information about the GSMMS, please refer to [5] and references therein.

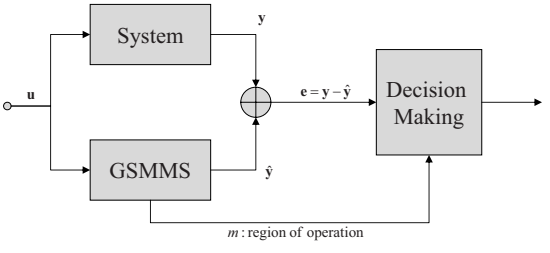


Fig. 2. Overview of region dependent decision making scheme

The resulting model is able to provide additional information about the system expected operational region. By expected operational region, we mean the region where the system is supposed to operate when the system is operating normally. The model is operating in a way that is similar to the operation of a hybrid system, in the sense that the system switches between discrete states (regions) and within each discrete state a local dynamic model generates continuous dynamics. The discrete state switchings underlying the apparently continuous dynamics are completely driven by system input history. This additional information about the discrete state operation thus leads to the natural solution to the problem of anomaly detection: region dependent decision making scheme. Figure 2 depicts the paradigm of the proposed region dependent decision making scheme. Instead of analyzing the residuals from all different regions, this region dependent decision making scheme compares the residual errors locally within each operational region defined by the multiple model system. The advantages of this newly proposed decision making scheme will be further discussed in detail in Sect. 5.

3 Simulation of Crankshaft Dynamics Based on First Principle Model

A first principle model of the crankshaft dynamics in a internal combustion engine is constructed in this section and serves as a *real* system for later testing the performance the proposed anomaly detection algorithms. We will formulate the first principle model in crankshaft angle domain instead of time domain as in [7].

Figure 3 shows the two-mass model for oscillating and rotating masses of piston and crankshaft motion at one cylinder. Following the conventions in [8], the mass of the connecting rod $m_{\rm rod}$ is separated into two portions: an oscillating portion $m_{\rm rod}\frac{l_{\rm osc}}{l}$ and a rotational portion $m_{\rm rod}\frac{l_{\rm rot}}{l}$, where $l_{\rm osc}$ and $l_{\rm rot}$ are defined by the location of the center of gravity of the connecting rod as shown in Fig. 3. Then the equivalent oscillating mass for the crankshaft-piston combination at each cylinder is

$$m_{\rm osc} = m_{\rm piston} + m_{\rm rod}\frac{l_{\rm osc}}{l} \quad . \tag{1}$$

where $m_{\rm piston}$ is the mass of the piston assembly including the mass of piston, wrist pin, piston rings etc. For a N cylinder engine, the total equivalent rotational mass for the crankshaft-piston combination is

$$m_{\rm rot} = \frac{J_{\rm cs}}{r^2} + N m_{\rm rod}\frac{l_{\rm rot}}{l} \quad . \tag{2}$$

under assumption that the crankshaft is infinitely stiff. $J_{\rm cs}$ is the moment of inertia of the crankshaft only.

The *torque balance equation* that relates the torques applied and the angular acceleration of the crankshaft in the crankshaft angle domain is given as [7]:

$$J\ddot{\alpha} = T_{\rm i}(\alpha) + T_{\rm r}(\alpha) + T_{\rm l}(\alpha) + T_{\rm fp}(\alpha) \quad . \tag{3}$$

where $\alpha = \alpha(t)$ is the crankshaft angle. $J = m_{\rm rot}r^2$ is the equivalent moment of inertia which takes into account the effects of all the rotating parts of crankshaft

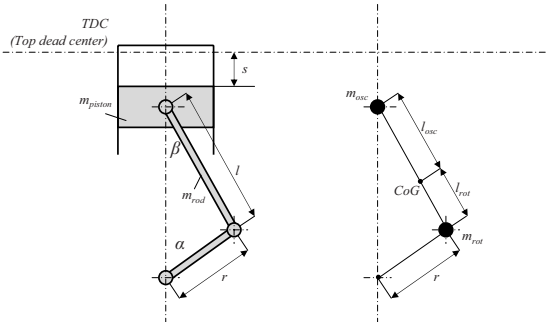


Fig. 3. Two-mass model for oscillating and rotating masses of piston and crankshaft motion at one cylinder

and the rods connecting to the crankshaft, where r is the crankshaft radius. $T_{\rm i}(\alpha)$ is the indicated torque generated by the combustion air pressures, $T_{\rm r}(\alpha)$ is the reciprocating torque, $T_{\rm fp}(\alpha)$ represents the energy losses due to friction and pumping and $T_{\rm l}(\alpha)$ is the load torque consists of wind resistance etc.

The piston stroke $s = s(\alpha)$, as shown in Fig. 3, is a function of crankshaft angle α and is completely determined by engine geometry. The individual torques in (3) can then be modeled as functions of piston stroke s and its derivatives with respect to crankshaft angle α as in [8] [7]. We list them here merely for the sake of completeness. For a N-cylinder engine, the indicated torque is the total torque generated by the combustion air pressures from all cylinders

$$T_{\rm i}(\alpha) = \sum_{j=1}^{N} \left(p_j(\alpha) - p_0\right) A_{\rm p} \frac{ds_j(\alpha)}{d\alpha} \quad . \tag{4}$$

where $p_j(\alpha)$ is the the air pressure in the combustion chamber of j^{th} cylinder, p_0 is the ambient pressure, $A_{\rm p}$ is the area of the piston head and $s_j(\alpha) = s(\alpha - (j-1)\frac{4\pi}{N})$ is the piston stroke of j^{th} cylinder.

The reciprocating inertia torque $T_{\rm r}(\alpha)$ describes the effects of oscillating mass $m_{\rm osc}$ from all cylinders of the engine

$$T_{\rm r}(\alpha) = -m_{\rm osc} \left[\sum_{j=1}^{N} \left(\frac{ds_j(\alpha)}{d\alpha} \right)^2 \ddot{\alpha} + \sum_{j=1}^{N} \frac{ds_j(\alpha)}{d\alpha} \frac{d^2 s_j(\alpha)}{d\alpha^2} \dot{\alpha}^2 \right] \quad . \tag{5}$$

The load torque $T_{\rm l}$ is viewed as an external input to the system in our simulation, which, in general, is proportional to the square of the vehicle speed plus some fluctuations.

$T_{\rm fp}(\alpha)$ represents the energy loss due to cylinder friction, bearing friction, pumping etc. The friction torque due to cylinder friction $T_{\rm cf}$ is given by the Coulomb law as in [8]

$$T_{\rm cf} = c_{\rm f} \sum_{j=1}^{N} \left(\frac{ds_j(\alpha)}{d\alpha} \right)^2 \dot{\alpha} \quad . \tag{6}$$

where the $c_{\rm f}$ is the viscous damping coefficient. The rest of $T_{\rm fp}$ due to bearing friction and pumping are lumped into a single term $T_{\rm bfp}$, which is modeled as a quadratic term with respect to angular velocity $\dot{\alpha}$ and external load torque $T_{\rm l}$.

We are interested in detecting and quantifying anomalies not only during the steady-state operations but also during engine routine operations. In order to simulate the transient behavior of engine, the firing pressure $p_j(\alpha)$ are *partially* modulated with a firing strength waveform which is considered to be proportional to the air mass inhaled into the cylinder during the intake stroke. Fig. 3 shows the in-cylinder pressure and the corresponding engine speed for a 8-cylinder engine under constant ignition angle. Three cycles are simulated.

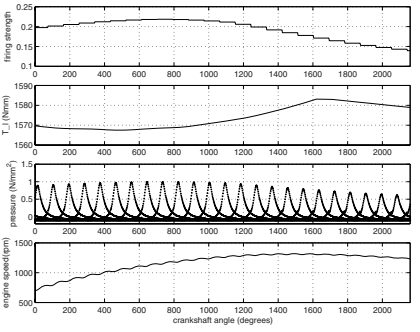


Fig. 4. Simulation results for a 8-cylinder engine. Total three cycles are simulated. The firing strength and the load torque can be viewed as inputs and the engine speed is the resulting output. The ignition angle is kept constant.

4 Modeling Crankshaft Dynamics Using GSMMS

The goal in this paper is to detect and quantifying the slowly evolving anomalies caused by parameter degradations. In particular, we are interested in assessing the system performance degradations caused by increases in the value of viscous damping coefficient c_{f} in (6). The changes in c_{f} could be the result of engine oil degradations. The first principle model created in Sect. 3 is used as a *real* system and an input-output model will be created using GSMMS to mimic its behavior. Then the residual errors between the first principle model and the GSMMS model are utilized to detect any performance deviations of the first principle model, which is depicted in Fig. 2.

A set of training data is first collected from the simulation of the first principle model. From each combustion cycle, which is corresponding to two revolutions of the crankshaft, eight samples are collected. This is equivalent to the sampling period of $\frac{4\pi}{N}$ in the crankshaft angle domain. The vector $\boldsymbol{x}$, based on which the system operation space is partitioned, has the following form

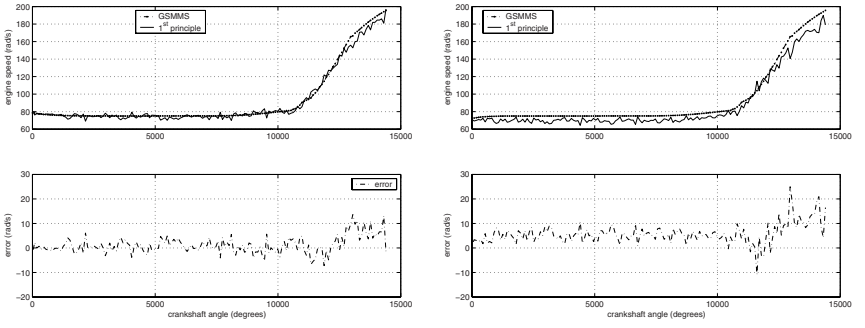
$$\begin{aligned}\boldsymbol{x} = [&A(\alpha), A(\alpha - \frac{4\pi}{N}), \ldots, A(\alpha - N\frac{4\pi}{N}), T_{\mathrm{l}}(\alpha), T_{\mathrm{l}}(\alpha - \frac{4\pi}{N}), \ldots, \\ &T_{\mathrm{l}}(\alpha - \frac{N}{2}\frac{4\pi}{N}), \dot{\alpha}(\alpha), \dot{\alpha}(\alpha - \frac{4\pi}{N}), \dot{\alpha}^2(\alpha), \dot{\alpha}^2(\alpha - \frac{4\pi}{N})]^T \ . \end{aligned} \tag{7}$$

where $A(\alpha)$ is the firing strength and ^{T} denotes the matrix transpose. Then within each region, a local model with the following form

$$\hat{\dot{\alpha}}(\alpha + \frac{4\pi}{N}) = \boldsymbol{\theta}_m^T \boldsymbol{x} + b_m \ . \tag{8}$$

is used to describe the evolutions of the system dynamics, where $\boldsymbol{\theta}_m$ is the coefficient vector of the local model for the m^{th} region and b_m is the corresponding bias term. Note that considerable multiplicative noises have been added to the

output. A multiple model system with twenty five regions is then constructed using the normal training data based on a batch training algorithm similar with the sequential training algorithm proposed in [5]. Note that the elements in the vector $\boldsymbol{x}$ have been properly normalized before training.



(a) $c_{\mathrm{f}} = 0.0175$ Ns/mm (b) $c_{\mathrm{f}} = 0.021$ Ns/mm

Fig. 5. Comparison of the outputs of the first principle model and the trained GSMMS model

Figure 5(a) compares the outputs of the first principle model and the trained GSMMS model when the viscous damping coefficient c_{f} is set to its nominal value. The GSMMS model can roughly capture the transient behavior of the crankshaft dynamics despite the existence of multiplicative noises. Figure 5(b) also shows the resulting residual errors when c_{f} is increased by twenty percent. Comparing with Fig. 5(a), the increases in magnitudes of the residual errors are clearly visible at the low engine speed. Once the GSMMS model is obtained from normal training data, the residual errors can then be utilized to detect and quantifying performance deviations or anomalies as illustrated in Fig. 2.

5 Assessment of Friction Induced Degradation

Interpretation of the residual errors between the outputs of the model and real system is not a trivial task. It is quite common in practice that the learned model is not perfect and there are always regions that the model can not approximate well compared with other regions. This is either due to the incomplete training data set or the inherent limitations of the learning algorithms. Consequently, the magnitudes of the modeling errors may vary at different operational regions. We can observe this from Fig. 5(a), where at low engine speed the magnitudes of the residual errors are substantially smaller than those at high engine speed. As a result, surges in residual errors could merely be the results of shifting from one operational region to another and not necessarily indicate anomalies.

A region dependent decision making scheme is proposed in [5] to overcome this problem and to improve the detection accuracy. The schematic overview

of the newly proposed approach is also shown in Fig. 2, where the residual errors are analyzed region by region. Figure 6 shows the error distributions in different regions under different values of viscous damping coefficient c_{f}. We can observe that as c_{f} shifts away from its nominal value $c_{\mathrm{f}} = 0.0175$ Ns/mm the error distributions also shift away from the its corresponding distributions that represent normal behavior.

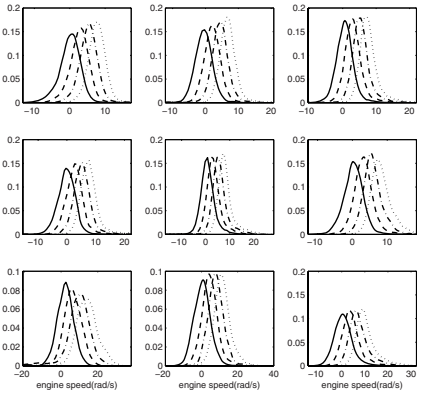


Fig. 6. Error distributions in selected nine adjacent regions. Solid line: $c_{\mathrm{f}} = 0.0175$ Ns/mm, which is the nominal value; Dashed line: $c_{\mathrm{f}} = 0.01925$ Ns/mm; Dash-dotted line: $c_{\mathrm{f}} = 0.021$ Ns/mm; Dotted line: $c_{\mathrm{f}} = 0.02275$ Ns/mm.

If only detection of anomalies is of interest, Exponentially Weighted Moving Average (EWMA) control charts [10] can be employed to perform this task by detecting off limit residual errors. Due to the fact that the distributions of the residual errors are region dependent, which can be observed from Fig. 6, it is advantageous to construct the EWMA control chart individually for each of the regions. Since the goal is to detect any anomalous system behaviors, the statistics of the residual errors during normal operations are used to construct the EWMA control charts within each region. By varying the exponential forgetting factor, we can also adjust the memory length and investigate the performance of the anomaly detection systems at given mean time to detection[1]. The mean time to detection can be varied by changing the values of the forgetting factor used in constructing EWMA control chart [10].

After constructing the EWMA control charts, we can then use them to detect anomalies. Essentially, the EWMA control chart in each region forms an anomaly detector, which is responsible for detecting anomalous behaviors once the system visits that region. Receiver Operating Characteristic(ROC) curves [9] are utilized in this paper to evaluate the performance of anomaly detection. Figure 7 shows the ROC curves for detecting 10% increase in the viscous damping coefficients

[1] Since the crankshaft dynamics are expressed in crankshaft angle domain, the mean time to detection here is referred as number of rotations.

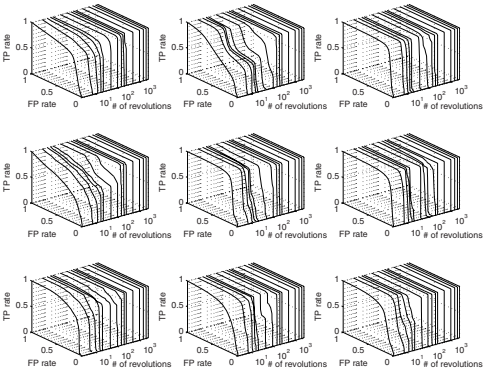


Fig. 7. Receiver operating characteristic curves from selected nine adjacent regions, which are the same with those shown in Fig. 6. The ROC curves are plotted against the amount of data used (N samples per revolution).

c_{f} at different given mean time to detection. The nine regions selected in Fig. 7 are the same with those selected in Fig. 6. We can observe that the performance the anomaly detectors in all nine regions keeps improving as more samples are utilized. Almost perfect detection can be achieved from all different regions when data length reaches about 1000 revolutions. Similarly observations have also been made for detecting 20% and 30% increases in c_{f} and the mean time to detection is substantially much shorter for achieving similar detection accuracy.

We have observed that as c_{f} shifts away from its nominal value the error distributions also shift away from the its corresponding distributions that represent normal behavior. As a result, the overlaps become smaller as the severity of the anomaly increases. Therefore, one way to quantify the severity of the anomaly is to evaluate the overlaps between the current error distributions with error distributions representing the normal operation as what has been done in [5].

6 Conclusions and Future Work

Growing Structure Multiple Model System(GSMMS) coupled with region dependent decision making scheme are utilized in this paper to detect and quantify the effect of friction changes on the dynamics of crankshaft in internal combustion engines. GSMMS combines the advantage of growing SONs with efficient linear parameter estimation algorithms into a integrated framework for modeling general nonlinear dynamic systems. The Voronoi sets defined by SONs naturally partition the full system operation space into small regions where system dynamics can be modeled locally using substantially simpler local models. The growing mechanism employed by GSMMS, which is based on growing SONs, is able to determine the number of regions automatically.

Based on the information about system operational regions provided by the resulting multiple model system, one can analyze the residual errors locally within

each region. As a result, the anomaly detection system is able to tolerate unequally distributed residual errors in different operational regions, which is common in practice for most of the nonlinear modeling techniques.

The preliminary results are encouraging. Several potential practical applications can be extended from this paper, such as oil quality assessment, bearing fault detection and vehicle maintenance quality evaluation. The application of the proposed anomaly detection approach on more complex dynamic system is currently under investigation.

References

1. Fritzke, B.: A Growing Neural Gas Network Learns Topologies. In: Advances in Neural Information Processing Systems, Vol. 7. MIT Press, Cambridge (1995) 625-632
2. Fritzke, B.: Growing Cell Structures - A Self-Organizing Network for Unsupervised and Supervised Learning. Neural Networks 7 (1994) 1441-1460
3. Ikonen, E., Najim, K.: Advanced Process Identification and Control. Marcel Dekker, New York (2002)
4. Sedgewick, R.: Algorithms in C++. Part 5, Graph Algorithms. 3rd edn. Addison-Wesley, London (2002)
5. Liu, J., Djurdjanovic, D., Marko, K., Ni, J.: Growing Structure Multiple Model System for Anomaly Detection and Fault Diagnosis. Technical report (2005)
6. Kohonen, T.: Self-Organizing Maps. Springer-Verlag, Berlin New York (1995)
7. Connolly, F.T.: Modeling and Identification of the Combustion Pressure Process in Internal Combustion Engines. Ph.D. dissertation, Dept. Elect. Eng., Univ. of Michigan, Ann Arbor (1992)
8. Kiencke, U., Nielsen, L.: Automotive Control Systems : for Engine, Driveline, and Vehicle. Springer, Berlin (2005)
9. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. 2nd edn. Wiley, New York (2001)
10. Montgomery, D.C.: Introduction to Statistical Quality Control. 4th edn. Wiley, New York (2001)

Fault Diagnosis for Induction Machines Using Kernel Principal Component Analysis

Jang-Hwan Park[1], Dae-Jong Lee[2], and Myung-Geun Chun[2]

[1] Information & Control Engineering, Chungju National University, Chungju, Korea
park@chungju.ac.kr
[2] Dept. of Electrical and Computer Engineering, Chungbuk National University, Korea
djmidori@empal.com, mgchun@chungbuk.ac.kr

Abstract. For the fault diagnosis of three-phase induction motors, we set up an experimental unit and then develop a diagnosis algorithm based on pattern recognition. The experimental unit consists of induction motor drive and data acquisition module to obtain the fault signals. As the first step for diagnosis procedure, preprocessing is performed to make the acquired current simplified and normalized. To simplify the input data, three-phase currents are transformed into the magnitude of Concordia vector. As the next step, feature extraction is performed by kernel PCA. Finally, we used the linear classifier based on two types of distance measures. To show the effectiveness, the proposed fault diagnostic system has been intensively tested with the various data acquired under the different electrical and mechanical faults with varying load.

1 Introduction

Induction motors play a very important role in the safe and efficient operation of industrial plants and processes. So, it is essential ingredient to prevent the abrupt break-down of them. The unexpected fault of motor causes many troubles such as the pause of overall process machinery as well as motors. Therefore, the advance technique, for example, fault monitoring or fault detection has become essential to effectively operate the process. In recent years, the fault detection and diagnosis of induction motors have been gaining more interests in the filed of highly reliable systems. For reliable fault diagnosis, it is extremely required in detecting and classifying the fault elements. There are some detection methods to identify the motor faults. Among the detection methods, the mainly used approaches are vibration monitoring and motor current signature analysis (MCSA). The vibration method is based on detecting vibration signal when motors happen to fault. This method, however, has some problems such as selection of reliable sensors and position attached on induction motors.

For fault detection method using MCSA, the stator current is detected by current transducers and sampled by the acquisition board. Here, sensing unit is easily constructed by using the current transformer comparing with vibration approach. Many results have clearly demonstrated that MCSA is a powerful technique for monitoring the health of three-phase induction motors [1-4]. Mark Fenger [1] discussed the effectiveness of MCSA to analyze different faulty modes of induction motors. From

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 406 – 413, 2006.

this, we know that many diagnosis techniques are based on spectrum analyzer. This method, however, can't effectively detect the various faults of motor when the acquired signal has the harmonic frequency produced by electrical device. On the other hand, Hamid Nejjari [2] and Fatiha Zidani [3] only use the Concordia's vector to perform the diagnosis process based on artificial neural networks and fuzzy logic, respectively. Recently, LDA based diagnosis system was presented [5].

To perform rapid diagnostic procedure, we extract patterns from Concordia's vectors rather than spectrums, and propose a diagnosis algorithm based on statistical pattern recognition, such as the kernel PCA and LDA. To show the effectiveness, the proposed fault diagnosis system has been intensively tested with the various data acquired under the different electrical and mechanical faults with varying load.

2 Analysis of Current Signals

In three-phase induction motors, current signals are measured between line to line rather than line to neutral point. Therefore, the measured current has no homo-polar component. A two-dimensional representation can then be used to describe three-phase induction motor phenomena. A suitable 2-D representation is based on the current Concordia vector [3].

The current Concordia vector components (i_a, i_b, i_c) are a function of mains phase variables (i_α, i_β) as

$$\begin{aligned} i_\alpha &= \sqrt{\frac{2}{3}} i_a - \frac{1}{\sqrt{6}} i_b - \frac{1}{\sqrt{6}} i_c \\ i_\beta &= \frac{1}{\sqrt{2}} i_b - \frac{1}{\sqrt{2}} i_c \end{aligned} \tag{1}$$

In ideal conditions, three-phase currents lead to a Concordia vector with the following components

$$\begin{aligned} i_\alpha &= \frac{\sqrt{6}}{2} I \sin \omega_s t \\ i_\beta &= \frac{\sqrt{6}}{2} I \sin\left(\omega_s t - \frac{\pi}{2} \right) \end{aligned} \tag{2}$$

Where I and ω_s denote the maximum current value and angular frequency, respectively. The trajectories of the Concordia vectors are circular pattern centered at the origin of the coordinates as shown in Fig. 1(b).

To progress diagnostic works, it is reasonable to use the one-dimensional data than two-dimensional data in sense of numerical simplicity. In this paper, therefore, the two-dimensional Concordia vector is converted into its one-dimensional signal in disregarded of phase property as follows.

$$i_{mag} = i_\alpha^2 + i_\beta^2 \tag{3}$$

Fig. 1(c) shows the current waveform transformed by Eq (3). As seen in this figure, the current signal is displayed as line pattern with dc value under ideal conditions. By using this simple current waveform, we make it possible to detect abnormal condition by monitoring the variations of acquired patterns.

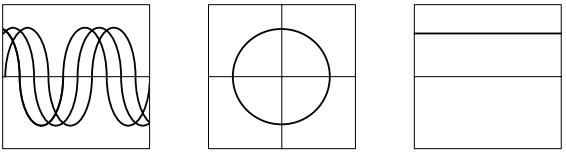


Fig. 1. Current Pattern for ideal conditions. (a) original 3-phase current, (b) Concordia pattern (2-dim.), (c) magnitude pattern (1-dim.).

3 Diagnosis Method

The proposed fault diagnosis system for induction motors is illustrated in Fig. 2. It is shown that the system contains three-modules as preprocessing, feature extraction and classification parts.

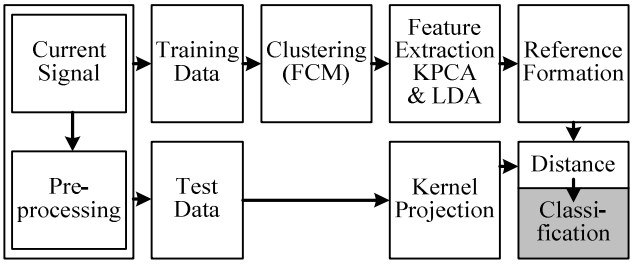


Fig. 2. The proposed fault diagnosis system

To make reasonable data for feature extractions, the preprocessing part performs the following steps. First, one periodic signal represented in three-phase currents by zero-crossing points is extracted. And then, the split frame signals are transformed into the magnitude of Concordia currents described by Eq. (2) and (3). Finally, we perform the normalization of the magnitude pattern and select the representative sample from many training data by using the clustering technique. Here, the number of representative sample is proportional to the dimension of projection matrix produced through the stage of feature extraction.

In this paper, we use the clustering technique of fuzzy c-means (FCM). FCM is a data clustering technique herein each data point belongs to a cluster to some degree that is specified by a membership grade. This technique was originally introduced by Jim Bezdek [6] as an improvement on earlier clustering methods. It provides a method that shows how to group data points that populate some multidimensional space into a specific number of different clusters.

Feature extraction is calculated by Kernel PCA. The basic idea of feature extraction is used on kernel PCA and LDA to map the input data into a feature space. Kernel PCA can be derived using the known fact that PCA can be carried out on the dot

product matrix instead of the covariance matrix [7]. Let $\{x_i \in R^M\}_{i=1}^N$ denotes a set of data. Kernel PCA first maps the data into a feature space F by a function $\Phi : R^M \to F$, and then performs standard PCA on the mapped data. Defining the data matrix X by $X = [\Phi(x_1)\ \Phi(x_2) \cdots \Phi(x_N)]$, the covariance matrix C in F becomes

$$C = \frac{1}{N}\sum_{1}^{N}\Phi(x_i)\Phi(x_i)^T = \frac{1}{N}XX^T \tag{4}$$

We assume that the mapped data are centered as $1/N \cdot \Sigma_1^N \Phi(x_i) = 0$. We can find the eigenvalues and eigenvectors of C via solving the eigenvalues problem

$$\lambda u = Ku \tag{5}$$

The $N \times N$ matrix K is the dot product matrix defined by $K = 1/N \cdot X^T X$ where

$$K_{ij} = \frac{1}{N}\Phi(x_i) \bullet \Phi(x_i) = \frac{1}{N}k(x_i, x_j) \tag{6}$$

Let $\lambda \geq \cdots \geq \lambda_p$ be the nonzero eigenvalues of K ($P \leq N$, $P \leq M$) and $u^1, \cdots, u^P$ the corresponding eigen-vectors. Then C has the same eigenvalues and there is a one-to-one correspondence between the nonzero eigen-vectors $\{u^h\}$ of K and the nonzero eigenvectors $\{v^h\}$ of C: $v^h = \alpha^h X u^h$, where α^h is a constant for normalization. If both of the eigenvectors have unit length, $\alpha^h = 1/\sqrt{\lambda_h N}$. We assume $\|v^h\| = 1/\sqrt{\lambda_h N}$ so that $\alpha^h = 1$.

For test data x, its h[th] principal component y_h can be computed using kernel function as

$$y_h = v^h \bullet \Phi(x) = \sum_{i=1}^{N} u_i^k k(x_i, x) \tag{7}$$

Then the Φ image of x can be reconstructed from its projections onto the first H ($\leq P$) principal components in F by using a projection operator P_H

$$P_H \Phi(x) = \sum_{h=1}^{H} y_h v^h \tag{8}$$

Commonly used kernels include polynomial, Gaussian and sigmoid kernel as

$$k(x, y) = (x, y)^d \text{ (polynomial)} \tag{9}$$

$$k(x, y) = \exp\left(\frac{\|x - y\|}{2\sigma^2}\right) \text{ (Gaussian)}$$

$$k(x, y) = \tanh\left(\kappa(x, y) + \Theta\right) \text{ (sigmoid)}$$

The kernel PCA allows us to obtain the features with high order correlation between the input data samples. In nature, the kernel projection of data sample onto the

kernel principal component might undermine the nonlinear spatial structure of input data. Namely, the inherent nonlinear structure inside input data is reflected with most merit in the principal component subspace.

To extract feature, we uses LDA as well as a kernel PCA so as to examine the discriminative ability of kernel principal components. In brief, LDA calculate the optimal projection matrix to maximize the ratio of the determinant of the between-class matrix of the projected samples to the determinant of the within-class scatter matrix of the projected samples. The detail descriptions are given in [8].

After obtaining the feature vectors, the classification or diagnosis is achieved by finding the minimum distance between the coefficients of test patterns and training patterns. Here, the distance is calculated for two cases; 1-norm distance and 2-norm (Euclidean) distance.

4 Experimental Result

The experiment equipment is composed of induction motor drive system, data acquisition board and PC. The current signal is saved on the MATLAB by a data acquisition board. Fig. 3 shows a block diagram of our fault diagnosis system and Fig. 4 shows experimental setup consisted of a three-phase induction motor connected the system with varying load and a PC-based data acquisition system. To obtain the various fault data, we construct the brake equipment in motor drive system. Table 1 presents the brief specification of each structural element.

To demonstrate the effectiveness of the proposed kernel PCA-based diagnosis technology, we acquired the fault signal under the unbalance condition among the 3-phase current. In order to simulate the electrical faults in the induction motor, potential transformer equipped through input sources. The signal is extracted under different load conditions from 0 to 0.12 kg·m in case of three simulated electrical faults as shown in the top of Table 2.

We considered the fault conditions of mechanical defects as well as electrical problems. The mechanical faults are simulated by the mechanical looseness of a

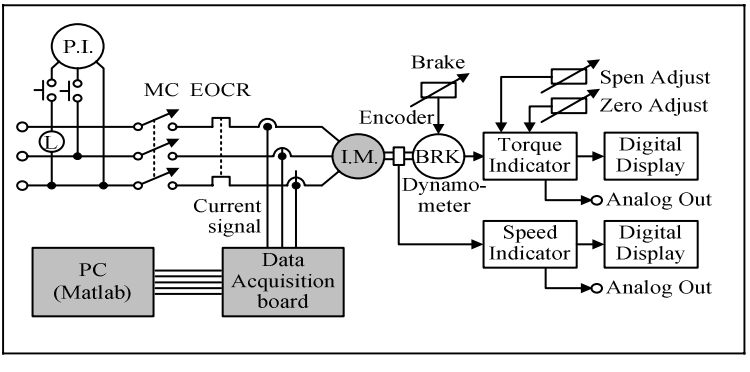


Fig. 3. Block diagram of the fault diagnosis system

screw used for fixation in the machine. When the screw is firmly fixed and the three phase currents are symmetrical, the induction machine is effectively operated in the normal condition. On the other hand, when the screw is loosened, abnormal current with harmonic frequency is produced because of abnormal vibration [9]. This current is measured through current sensors in experimental unit. Three types of looseness are considered: two left, two right and all of four screws to fix a motor. The bottom in Table 2 also shows different conditions related mechanical faults.

Table 1. Specification of the system

Comp.	Spec.
Induction motor	0.4 kW(1/2 HP), 220 V, 60 Hz 4p, 1700 rpm, Efficient: 65%
Current sensor	Rating current: ±10 A, Output: ±4 V
Data acquisition	500 Ks/s, 16 bit, 4 channel
Load equipment	Brake system of eddy current type
PC	Pentium 4 2.0 GHz

Table 2. Different conditions for data acquisition

Fault	Condition	Load
Health	No faulty condition	1. No load 2. 0.04 kg·m 3. 0.08 kg·m 4. 0.12 kg·m
Electrical Faults	1% unbalance for a phase	
	2% unbalance for a phase	
	3% unbalance for a phase	
Mechanical Faults	Looseness of two left screws	
	Looseness of two right screws	
	Looseness of all screws	

All diagnosis processes are executed on the offline with the MATLAB. The fault signal is obtained under the various fault condition as seen in Table 2. Here, the number of motor condition is 7 including health, 3-electrical faults and 3-mechanical faults. Also, we obtained the fault signal by varying the three types of load condition for each motor condition. With the time interval, we acquired the data three times per each condition. So, we obtained 84 current signal [7(fault conditions) X 4(load conditions) X 3(experiments)]. The each signal detected and selected per each periodic unit. And then feature extraction is performed by using the selected periodic signal. Among the many periodic signals, we select the sample by choosing the first 5-periodic signal for each acquired data. Finally, we obtained 420 samples (84X5).

After acquiring the sample data, features are obtained by PCA, kernel PCA and LDA. Fault diagnosis is performed by comparing the test features with training features. To be accurate diagnosis, it is very important to determine the training set. In

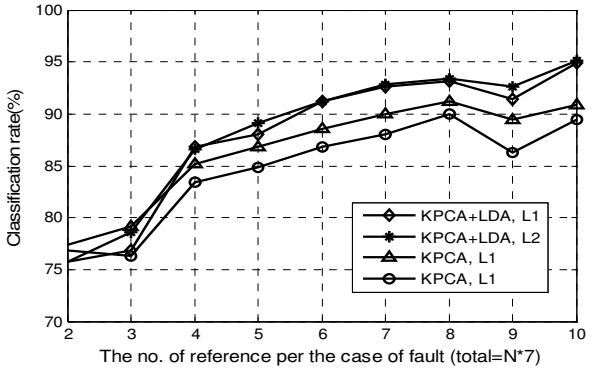


Fig. 4. Classification rate according to the number of cluster for test data (the dimension of features is 45)

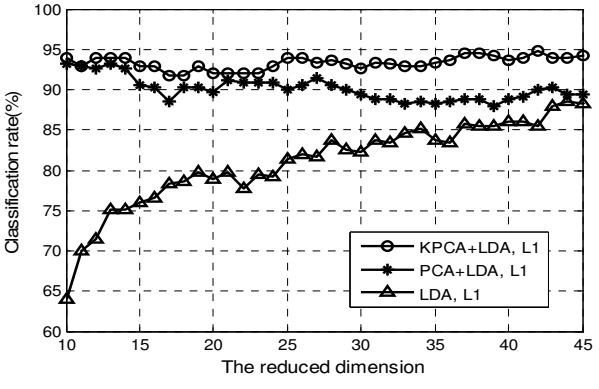


Fig. 5. Testing performance for kernel PCA+LDA, use of polynomial kernel, $(x,y)^3$

Table 3. Testing performance for each fault [%]

Fault	H	E1	E2	E3	M1	M2	M3
KPCA+LDA	90.1	95.9	100	96.3	90.1	95.6	94.0
PCA+LDA	78.4	94.1	95.5	95.5	80.2	84.1	84.2
LDA	64.3	96.1	100	97.6	74.2	86.2	86.3

this paper, we construct the training features by FCM clustering algorithm. Fig. 4 shows the classification rate according to the number of cluster. As seen in Fig. 4, classification shows best performance when the number of cluster is equal to 10.

Fig. 5 shows classification results according to feature reduction methods as LDA, PCA + LDA and kernel PCA + LDA. As seen in Fig. 6, the kernel PCA+LDA shows best performance. More specifically, total classification rates are equal to 81.0%, 90.1% and 94.0% for LDA, PCA+LDA and kernel PCA+LDA. Table 3 presents classification results for each fault in detail. As seen in Table 3, classification shows

the better performance by using the electrical faults (E1-E3) rather than mechanical faults (M1-M3).

5 Conclusion Remarks

In this paper, we proposed an advanced fault diagnosis system based on kernel PCA and LDA. To be more accurate diagnosis system, the proposed system consisted of preprocessing, formation of training and test features, and classifier modules. In preprocessing module, the acquired 3-phase fault current was transformed into Concordia vector space with 2-dimension. And training and test features were obtained by feature extraction methods. In particular, we used the FCM clustering technique to get the proper training set from the samples. Finally, classifier was designed by two types of distance measure. To show the effectiveness, we acquired the various fault condition under load variation for a 0.5 Hp squirrel-cage induction motor. From experimental test, we obtained classification rate about 95%, which is satisfactory and promising for an industrial application of the statistically pattern recognition based on kernel PCA and LDA.

Acknowledgement

This work has been supported by EESRI (R-2003-B-285), which is funded by MOCIE (Ministry of Commerce Industry and Energy).

References

1. Thomson, W. T., Fenger, M.: Current Signature Analysis to Detect Induction Motor Faults, IEEE Ind. Applications Magazine (2001) 26-34
2. Nejjari, H., Benbouzid, M. H.: Monitoring and Diagnosis of Induction Motors Electrical Faults Using a Current Park's Vector Pattern Learning Approach, IEEE Trans. Ind. Application 36 (2000) 730-735
3. Zidani, F.: Induction Motor Stator Faults Diagnosis by a Current Concordia Pattern-Based Fuzzy Decision System, IEEE Trans. Energy Conversion 18 (2004) 469-475
4. Peter,V.: Artificial Intelligence Based Electrical Machines and Drives. Oxford Sci. Pub (1999)
5. Lee, D.J., Park, J., Kim, D.H., Chun, M.G.: Fault Diagnosis of Induction Motor Using Linear Discriminant Analysis. Knowledge-Based Intelligent Information and Engineering Systems. KES 2005, Proceedings, Part IV (2005) 860-865
6. Bezdec, J.: Pattern Recognition with Function Algorithms. Plenum Press, New York (1981)
7. Scholkopf, B., Smola, A.: Nonlinear Component Analysis as a Kernel Eigenvalue Problem. Neural Computation 10 (1998) 1299-1319
8. Richard, O. D., et al. : Pattern Classification, John Wiley & Sons Inc. (2002)
9. Wu, S., Chow, T.: Induction Machine Fault Detection using SOM-based RBF Neural Network, IEEE Trans. Ind. Electron.51 (2004) 183-194

Application of RBF and SOFM Neural Networks on Vibration Fault Diagnosis for Aero-engines

Kai Li, Dongxiang Jiang, Kai Xiong, and Yongshan Ding

Department of Thermal Engineering, Tsinghua University, Beijing 100084, China
likai03@mails.tsinghua.edu.cn, jiangdx@tsinghua.edu.cn

Abstract. This paper applies two ANN methods—RBF and SOFM on fault diagnosis for two-shaft aero-engines. Two-shaft aero-engines are complex rotating machines which have many components and high rotating speed. First we presented both the principles and advantages of RBF and SOFM neural networks. Second we described the fundamentals of two-shaft aero-engines vibration fault diagnosis, and then obtained the standard fault samples (training samples) and simulation samples (testing samples). Third we applied the two ANN methods to perform diagnosing. The accurate diagnostic results have proved the effectiveness of the RBF and SOFM methods for vibration fault diagnosis of two-shaft aero-engines. Finally, the relative advantages and disadvantages of the two ANN methods are contrasted, and suggestions can be obtained on when one might use one of the two methods.

1 Introduction

Artificial Neural Networks (ANNs) have proven effective in solving problems in a wide variety of areas, such as image processing, speech recognition and automation control [1]. We describe here the application of ANNs to the fault diagnosis of two-shaft aero-engines. The two-shaft aero-engine is one kind of rotating machinery that has many components and is highly complex. Although the nonlinear model-based method is popularly used in the industry, it is hard to build a reasonably accurate model. Another potential disadvantage is that model-based method can only identify faults that have been modeled already and might misclassify a novel abnormal situation [2]. ANNs appear to be reasonable alternatives to traditional method because of their ability to learn complex nonlinear functions. By means of continuously learning from the training samples (history data), a neural network renews its weights and, therefore, memorizes the correct responses. So ANN does not require the detailed knowledge about the system for establishing its model and is able to overcome the shortcomings of model-based techniques [3].

This paper presents two ANN methods, Radial Basis Function (RBF) and Self-Organizing Feature Map (SOFM) neural networks, on two-shaft aero-engines fault diagnosis. Both of them have powerful advantages compared with the most popularly used Back-Propagation neural network (BPNN). RBF network has the advantages of fast learning, high accuracy and strong self-adapting ability. While the SOFM network has the most distinct feature that training is an unsupervised process. It also

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 414 – 419, 2006.

has some advantages such as simple structural algorithm, fast learning and lateral association, etc. We applied the two ANN methods on fault data sets and the experiment results verified their nice prospect of applying to complex rotating machinery. At the end of this paper, we compared them in different ways.

2 Principles of RBF and SOFM Neural Networks

2.1 RBF Neural Network

The Radial Basis Function network has a multiplayer feed forward structure. The hidden layer is made up of a radial basis function. The character is that its response decreases monotonically with distance from center point [3]. The structure of the radial basis function network is illustrated in Fig.1.

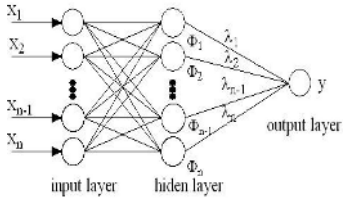


Fig. 1. Structure of RBF Neural Network

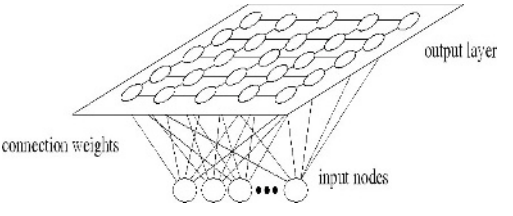


Fig. 2. SOFM Neural Network

X is the input, y is the output. The hidden layer is connected with output layer by weights λ . RBF neural networks are modeled by the following relation,

$$y = \sum_{i=1}^{n} \lambda_i \phi_i (P\ x - c_i\ P) + \lambda_0 \cdot \tag{1}$$

Where $\phi_i(x)$ denotes the RBF,

$$\phi_i (\| x - x_i \|) = \exp\left(\frac{\| x - x_i \|}{2\sigma^2} \right) \cdot \tag{2}$$

x_i is the center of the Gaussian Function and σ is the Standard Deviation. x_i , σ and λ are the parameters of the model. The RBF network has a universal approximation capability and theoretically, it can approximate any nonlinear function with sufficient neurons in the hidden layer [4].

2.2 SOFM Neural Network

Compared with the most popularly used model BPNN, SOFM has a structure more similar to humanity biology. The most distinct feature is that the training is an unsupervised process [5]. The structure of the model is shown in Fig. 2.

Such model is made up of two layers—input and output: Every input neuron connects with the output ones by connection weighting vectors. The number of the input nodes is determined according to the dimensions of the input vectors and the input nodes receive input values. The output layer is a plane, which is made up of

neurons arrayed in a certain way (square or hexagon, etc). The learning of the Self-organizing Feature Map neural network is different from that of RBF, because of self-organizing. That is to say, it is not necessary to give any prospective output before the learning process [5]. The training procedures (or algorithms) are available in reference [6].

3 Fundamental of Two-Shaft Aero-engines Fault Diagnosis

The two-shaft aero-engine's structure is highly complex. It has two compressors–high and low pressure compressors, and similarly has two turbines. The high pressure rotor drives high-pressure compressor and turbine, while the low pressure rotor drives the low-pressure ones. We described high-pressure rotor as HPR, low- pressure rotor as LPR, high-pressure compressor as HPC, and low-pressure compressor as LPC. According to field expert's experience and theoretical research outcomes, Table 1 shows the vibration faults classification of two-shaft aero-engines. The table includes typical nineteen faults numbered F0 to F18. Unbalance is the most common cause of aero-engines vibration as the result of non-uniform mass distribution.

Table 1. Faults classification of two-shaft aero-engines

Faults No.	Faults Name	Faults No.	Faults Name
F0	normal	F10	HPR axial rubbing
F1	LPC unbalance	F11	LPR clearance vibration
F2	HPC unbalance	F12	HPR clearance vibration
F3	LPR misalignment	F13	LPR bearing rigidities (vertical and horizontal) differ greatly
F4	HPR misalignment	F14	HPR bearing rigidities (vertical and horizontal) differ greatly
F5	LPR bent shaft	F15	LPR crack
F6	HPR bent shaft	F16	HPR crack
F7	LPR shaft-seal rubbing	F17	LPC surging
F8	HPR shaft-seal rubbing	F18	HPC surging
F9	LPR axial rubbing		

Nowadays spectrum analysis is well proven as a practical and powerful tool for fault diagnosis of rotating machinery because it results from a great deal of engineering experience. However, it is a difficult work to find the best relationship of faults and spectrum data because state of rotating machinery is complex, and influenced by numerous of process parameters [7]. Table 2 shows the spectrum symptoms and process parameters description.

Now we get the standard fault samples for a two-shaft aero-engine which is described by a symptom-fault relationship matrix. The matrix includes typical nineteen faults with seventeen spectrum symptoms and three process parameters. The data in each category are the percentages of possibilities based on experience. For example, with an initial unbalance of LPC, F1, there is a 90% probability it will appear at running frequency (s6) and 5% possibility it will appear at twice running frequency (s10) and other higher multiples.

Table 2. Spectrum and procecess description

Spectrum	Description	Spectrum	Description
s1	0~0.5X1	s11	2X1~2X2
s2	0.5X1	s12	2X2
s3	0.5X1~0.5X2	s13	2X2~3X1
s4	0.5X2	s14	3X1
s5	0.5X2~X1	s15	3X1~3X2
s6	X1	s16	3X2
s7	X1~X2	s17	>3X2
s8	X2	US	unsteady vibration
s9	X2~2X1	VER	vertical vibration
s10	2X1	LEV	level vibration

4 Applications of Two ANN Methods on Fault Diagnosis

In section 3 we get standard fault samples—nineteen standard fault samples (F0-F18). In order to prove the practicability of this method, we provide seven testing samples (simulation fault samples) T0—T6, which are simulated from F0, F1, F6, F11, F16, F18 and F9. Now we have both the training samples and testing samples to continue our research.

4.1 Application of RBF

We construct the RBF network according to the symptom-fault relationship matrix and the principle of RBF. Because the matrix has nineteen standard faults, the number of input nodes is nineteen. Accordingly the hidden and output layers have nineteen nodes, too. Other than input the standard training samples, we also need to input the prospective output. The prospective output is a 19×19 unit matrix "A", and "A_{ii}=1" means the "i"th fault has strictly occurred. Now we input the training samples and prospective output, construct the network, and diagnose the testing samples using the RBF neural network. Table 3 showed the part of the diagnostic output results.

Table 3. Part of the diagnosis output by RBF network

Test samples	Output nodes F0	F1	F2	F3	F4	F5	F6	F7	F8	F9
T0	0.94	0	0	0	0.06	0	0	0	0	0
T1	0	0.94	0	0	0.05	0	0	0	0	0
T2	0	0	0	0	0.05	0	0.95	0	0	0
T3	0	0	0	0	0.06	0	0	0	0	0

The actual output is also a matrix. We defined if the value of A_{ij} is larger than 0.8, then the "i"th test sample is diagnosed as the "j"th fault. For example, A_{11}=0.94>0.8, then T0 is diagnosed as F0 (normal). Similarly we can get the other conclusions. T1—F1 (LPC unbalance); T2—F6 (HPR bent shaft); T3—F11 (LPR clearance vibration); T4—F16 (shaft-seal rubbing); T5—F18 (HPC surging); T6—F9 (LPR

axial rubbing). This accurate classification results has verified the practicability of the RBF network method.

4.2 Application of SOFM

The number of output nodes will affect the diagnostic result of SOFM directly [5]. In order to get the best result by least time consuming, we select the number of output nodes as 289 (17×17). We input the training samples and diagnose the testing samples by SOFM neural network. The training result is shown in Fig. 3. As is shown in Fig.3, the training samples have been fully distinguished (classified). The output input seven simulation samples mapping is shown in Fig. 4.

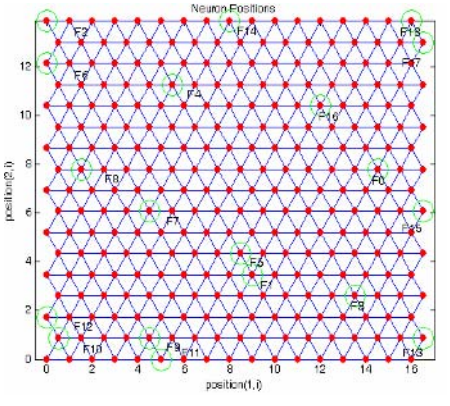


Fig. 3. Training result by SOFM

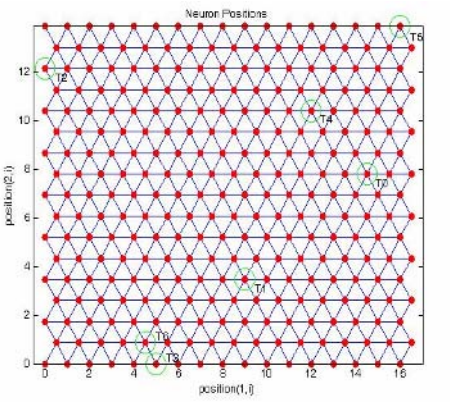


Fig. 4. Diagnostic result by SOFM

Comparing Fig.3 and Fig.4, we find T0 is in the same position on the network as F0, and then T0 is diagnosed as F0. We may also get the other accurate results.

4.3 Comparison of RBF and SOFM

Although both of the two ANN methods performed satisfactorily for the chosen case study, they have their own advantages and disadvantages. In our experiment, the time consuming of RBF for case study is 0.593s, while SOFM is18.875s. In order to

Table 4. Comparison of RBF and SOFM neural networks

Criterion	RBF Neural Network	SOFM Neural Network
Structure	Multiple layers with hidden layer	Two layers without hidden layer
Algorithm	Prospective output needed	Prospective output no needed
Convergence Speed	faster than BPNN and SOFM neural networks	faster than BPNN but slower than RBF neural networks
Robustness to noise	Better than BPNN but less than SOFM networks	Better than both BPNN and RBF networks
Rang of application	Good	Good

compare the robustness of the two methods, we add noise to the simulation fault samples and performed diagnosing again. We also concern on the comparisons of the structures, algorithms and application ranges of the two ANN methods. A summary of the comparison is provided in Table 4.

5 Conclusions

In this paper, two artificial neural network methods—Radial Basis Function (RBF) and Self Organizing Feature Mapping (SOFM)—have been proposed to diagnose vibration faults of two-shaft aero-engines. First we presented both the principles and advantages of RBF and SOFM neural networks. These two ANN methods are able to overcome the shortcomings of model-based method. Second we described the fundamentals of vibration fault diagnosis. The standard fault samples are symptom-fault matrix including nineteen faults, each is characterized by twenty symptoms. Third we applied RBF and SOFM methods to perform diagnosing. We take the symptom-fault matrix as training samples and provide seven simulation fault samples to be diagnosed. The accurate diagnostic results have fully verified the practicability of the two ANN methods for two-shaft aero-engines. Finally, we compared RBF and SOFM networks in structure, algorithm, convergence speed, robustness and so on. The suggestions can be obtained on when one might use one of the two methods.

References

1. Chen, C.Z., Mo, C.T.: A Method for Intelligent Fault Diagnosis of Rotating Machinery. Digital Signal Processing 14 (2004) 203-217
2. Rengaswamy, R., Mylaraswamy, D., Arzen, K.E., Venkatasubramanian, V.: A Comparison of Model-based and Neural Network-based Diagnostic Methods. Engineering Applications of Artificial Intelligence 14 (2001) 805–818
3. Chang, Y.J.: Fault Detection for Plasma Etching Process Using RBF Neural Networks. Lecture Notes in Computer Science, 3498 (2005) 538–543
4. Wen, G.R., Qu, L.S., Zhang, X.N.: A Rapid Response Intelligent Diagnosis Network Using Radial Basis Function Network. Lecture Notes in Computer Science, 3498 (2005) 508–513
5. Jiang, D.X., Li, K., Zhao, G., Diao, J.H..: Application of Fuzzy SOFM Neural Network and Rough Set Theory on Fault Diagnosis for Rotating Machinery. Lecture Notes in Computer Science, 3498 (2005) 561–566
6. Kohonen, T.: The Self-organizing Map. Proceedings of IEEE **78** (1990) 1464–1480
7. Zhao, G., Jiang, D.X., Li, K., Diao, J.H.: Data Ming for Fault Diagnosis and Machine Learning for Rotating Machinery. Key Engineering Materials 293-294 (2005) 175–182

Predictive Fault Detection and Diagnosis of Nuclear Power Plant Using the Two-Step Neural Network Models

Hyeon Bae, Seung-Pyo Chun, and Sungshin Kim

School of Electrical and Computer Engineering, Pusan National University,
30 Jangjeon-dong, Geumjeong-gu, 609-735 Busan, Korea
{baehyeon, buzz74, sskim}@pusan.ac.kr
http://icsl.ee.pusan.ac.kr

Abstract. Operating the nuclear power generations safely is not easy way because nuclear power generations are very complicated systems. In the main control room of the nuclear power generations, about 4000 numbers of alarms and monitoring devices are equipped to handle the signals corresponding to operating equipments. Thus, operators have to deal with massive information and to analyze the situation immediately. In this paper, the fault diagnosis system is designed using 2-steps neural networks. This diagnosis method is based on the pattern of the principal variables which could represent the type and severity of faults.

1 Introduction

The nuclear power generation was introduced in the middle of 1950's and has been continuously expanded until now. Recently, 440 nuclear power generations are charging 16% of total electric power production in the world [1]. However, the nuclear power generation is a large-scaled and complex system that firstly requires stability of the plant. Therefore, it is very difficult and complicated to control and manage the power generation. To control the system, there are about 4,000 sensors and instruments in the main control room. If the power generation plant becomes transient, stop, or emergence status caused by malfunction or abnormal situations of the power generation, over 500 alarms are early raised and many measurement signals are sharply changed. Therefore, operators can feel difficult to identify and determine the status of the plant [2-4].

In this research, an automatic diagnostic algorithm is proposed to avoid the improper management of the plant failure caused by unskilled operators. When the failures of the plant happen, the fault detection and diagnosis is achieved by the neural network model that uses the failure patterns for model inputs. However, the system complexity can be increased because the failure type and severity have to be considered together for correct analysis of the cause and trend. Therefore, the multi-steps neural network model is proposed and designed in this research. The first model detects the failure type and the other model diagnoses the severity. The proposed system shows the good performance in the case that the feature of the failure is not significant.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 420 – 425, 2006.

2 Pressurized Water Reactor (PWR)

This research dealt with the pressurized water reactor (PWR) that is a global model for power generation. This is the most common type, with over 230 in use for power generation and a further several hundred in naval propulsion [5]. It uses ordinary water as both coolant and moderator. The design is distinguished by having a primary cooling circuit which flows through the core of the reactor under very high pressure, and a secondary circuit in which steam is generated to drive the turbine. The steam drives the turbine to produce electricity, and is then condensed and returned to the heat exchangers in contact with the primary circuit.

In this research, the applied failure data are gathered from the simulator that mimics the physical power generation (Fig. 1). The data are acquired under the same condition with the actual situation when the failures actually happen. Therefore, the data collected from the simulator can guarantee reliability for the application [6]. Using the simulator, the several data sets can be gathered under the similar status with actual situations; therefore, the data can be available for the failure diagnosis with avoiding the drawback.

Fig. 1. The physical simulator that was used in this research

3 Experimental Results

3.1 Fault Data of Nuclear Power Generation

In this research, the automatic failure diagnostic system was proposed for rapid detection and countermeasures. Figure 2 shows the system structure that is related to this research [7]. Character *P* as shown in Table 1, indicates the measured parameters. For the failure detection and diagnosis, the parameters are measured at the points. As shown in the table, the variation of the parameter values actually represents the features of the failures that are treated importantly in operation of the nuclear power generation. Character *A* as shown in Table 2, shows the failure points, that is, the failure types. Because the failures have possibility to be expanded to a large-scale accident, in this research, 11 parameters of the character *P* were used to detect and diagnose the 6 failures of the character *A*.

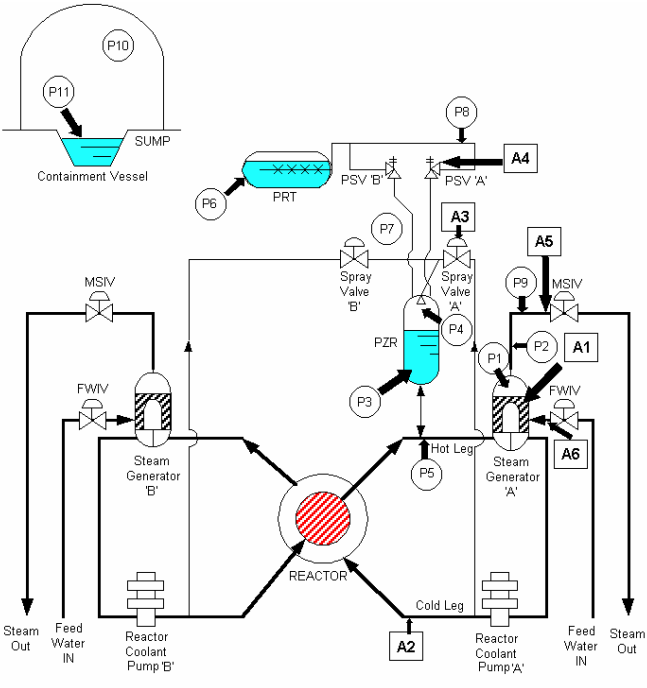


Fig. 2. The system scheme of the nuclear power plant

Table 1. Principal measured variables for fault diagnosis

No.	Variables	Abbreviation
P1	**S**team **G**enerator **1 L**evel	SG1L
P2	**S**team **G**enerator **1 S**team **P**ressure	SG1SP
P3	**P**ressuri**z**e**r L**evel	PZRL
P4	**P**ressuri**z**e**r P**ressure	PZRP
P5	**L**oop **1 W**ide range **T**emperature	LP1WT
P6	**P**ressurizer **R**elief **T**ank **L**evel	PRTL
P7	**S**pray **L**ine **D**ischarge **T**emperature	SLDT
P8	**P**ressurizer **S**afety valve **L**ine **T**emperature	PSLT
P9	**S**team **G**enerator **B**low-down **R**adiation	SGBR
P10	**C**ontainment **V**essel **H**igh range **R**adiation	CVHR
P11	**C**ontainment **V**essel **S**ump **L**evel	CVSL

Table 2. Fault types of nuclear power plants

No.	Failure name	Abbreviation
1	**S**team **G**enerator **T**ube **R**upture	SGTR
2	**L**oss **o**f **C**oolant **A**ccident	LOCA
3	**P**ressuri**z**e**r S**pray valve stuck open	PZR SPRAY
4	**P**ressuri**z**e**r P**ressure **S**afety **V**alve stuck open	PZR PSV
5	**M**ain **S**team **L**ine **B**reak	MSLB
6	**M**ain **F**eed **L**ine **B**reak	MFLB

3.2 Structure of Network and Data

The goal of the research is to classify the failure types and severities. The first model is used to classify only the failure types; therefore the model is trained with measured data of the failure types. Firstly the failure type is classified by the first model and then the severity is detected by the second model sequentially. As shown in the block diagram of Fig. 3, both networks are not linked structurally.

The data structure of the first neural network model is shown in the upper of Fig. 4. Each variable has 31 sampling data and one input set includes 11 variables. In this research, the numbers of the failures are six, so the input and output data set become a 6 by 31 matrix, respectively. However, the effect of the severity is reflected by using different target values, that is, 0.6, 0.7, 0.8, 0.9, and 1.0 are used for from 40% severity to 80% severity by 10% degree. In the second network model, the severity is detected for a classified failure using the same input data. The severity consists of 5

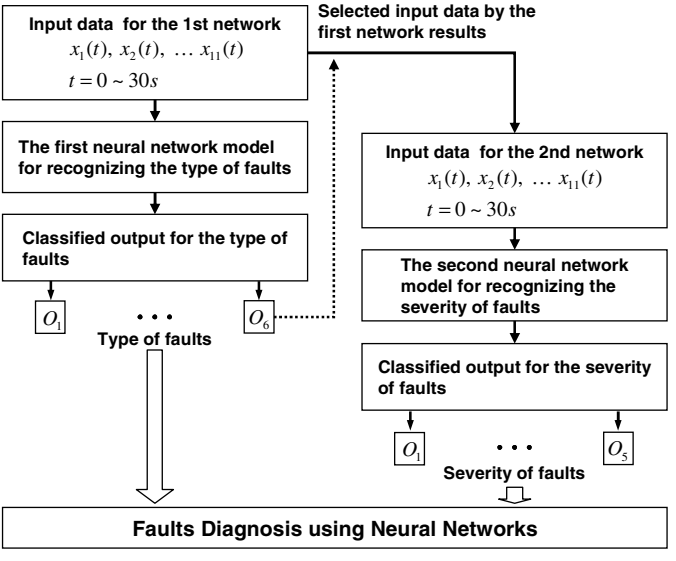


Fig. 3. The structure of the proposed two-steps neural network model

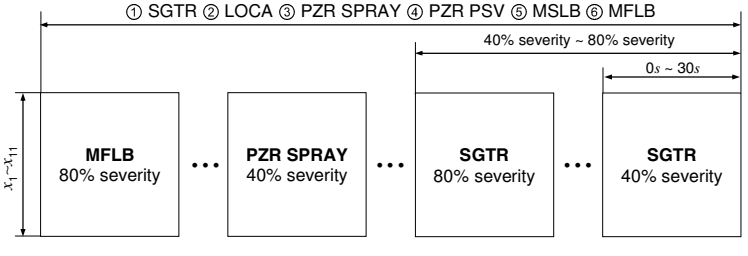


Fig. 4. Structure of the input variables for the primary network

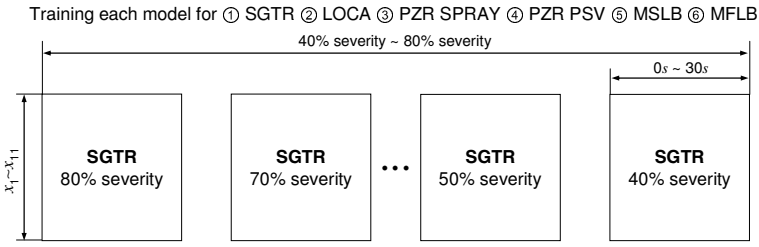


Fig. 5. Structure of the input variables for the secondary network

degrees from 40% to 80% by 10% and 31 sampling values are inputted to the model with an order. Figure 5 shows the data structure of the network that is applied to detect the failure severity.

3.3 Results of Failure Detection and Diagnosis

To evaluate the simulation results of the failure diagnosis, firstly the accuracy of the first model is presented in Table 3. This table shows the classification results for the six types of the failure in the first model. As shown in the result, the failure types are classified well but the error of the result can be high when the parameter pattern of the failure is similar each other or the variation of the parameter values is insignificant. The second network model is used to detect the failure severity of the classified failure by the first network model. The target values are set by 1.0 (one) in model training because the second model is just used for severity detection. On the other hand, the first model was used for six failures and five severities with from 0.6 to 1.0

Table 3. The results of the fault diagnosis corresponding to fault types

Fault types	Severity	Target	Diagnosis results (Average for 30sec)						Final result (After 30sec)
			y_1	y_2	y_3	y_4	y_5	y_6	
SGTR	40%	0.6	**0.565**	0.001	0.125	0.042	0.009	0.029	**0.607**
	80%	1.0	**0.847**	0.001	0.025	0.066	0.052	0.233	**0.902**
PZR SPRAY	40%	0.6	0.019	0.001	**0.773**	0.002	0.081	0.090	**0.673**
	80%	1.0	0.094	0.000	**0.737**	0.005	0.004	0.086	**0.639**
MSLB	40%	0.6	0.065	0.010	0.036	0.002	**0.567**	0.043	**0.649**
	80%	1.0	0.024	0.102	0.016	0.064	**0.283**	0.058	**0.815**

Table 4. The results of the fault detection corresponding to fault severities

Fault types	Severity	Target	Diagnosis results (Average for 30sec)					Final result (After 30sec)
			y_1	y_2	y_3	y_4	y_5	
SGTR	40%	1.0	**0.863**	0.151	0.017	0.000	0.008	**0.931**
	80%	1.0	0.031	0.000	0.000	0.006	**0.979**	**0.995**
PZR SPRAY	40%	1.0	**0.816**	0.267	0.028	0.008	0.020	**0.934**
	80%	1.0	0.073	0.074	0.065	0.065	**0.719**	**0.974**
MSLB	40%	1.0	**0.593**	0.270	0.211	0.090	0.002	**0.817**
	80%	1.0	0.056	0.091	0.128	0.313	**0.661**	**0.952**

value in training. The untrained data were evaluated by the trained model and the results are shown in Table 4.

4 Conclusions

The detection and diagnosis model was constructed by two-step networks. One is to classify the failure type and the other is to detect the failure severity. After training the model, the evaluated results showed the reliable to apply in the failure detection. But when the pattern of the parameters is similar, the results become deteriorated. Therefore, it is necessary to adopt other supplementary methods to increase the accuracy of the network models.

Acknowledgement

This work was supported by "Research Center for Logistics Information Technology (LIT)" hosted by the Ministry of Education & Human Resources Development in Korea.

References

1. Chang, S. H. and Baek, W. P.: Nuclear Safety. Cheong Moon Gak Publishers, Seoul (1998) 1-37
2. Uhrig, R. E.: Potential Application of Neural Networks to the Operation of Nuclear Power Plant. Nuclear Safety **32** (1) (1991) 68-78
3. Kang, K. S., Kim, H. G., Chang, S. H., Jeong, H. K., and Park, S. D.: Development Strategies on an Integrated Operator Decision Aid Support System for Nuclear Power Plants. IEEE Transactions on Nuclear Science **40** (4) (1993) 1295-1304
4. PWR Accident Analysis and Mitigating Core Damage. Westinghouse Training and Operational Services (1991)
5. NEI World Nuclear Industry Handbook (2005)
6. Lee, Y. K. and Jo, B. H.: Development of Training Simulator for Power Plant. Korea Electric Power Research Institute (1998)
7. Moon, B. H.: Process Diagram for Kori Unit # 2. Kori Nuclear Power Site, Korea Hydro and Nuclear Power Co. Ltd. (1999)

Kernel PCA Based Faults Diagnosis for Wastewater Treatment System

Byong-Hee Jun[1], Jang-Hwan Park[2], Sang-Ill Lee[3], and Myung-Geun Chun[4]

[1] School of Fire & Disaster Prevention, Kangwon National University, Samcheok, Korea
[2] School of Electrical and Electronic Engineering, Chungju National University, Korea
[3] Dept. of Environmental Engineering, Chungbuk National University, Cheongju, Korea
[4] Dept. of Electrical and Computer Engineering, Chungbuk National University, Korea
mgchun@chungbuk.ac.kr

Abstract. A Kernel PCA based fault diagnosis system for biological reaction in full-scale wastewater treatment plant was proposed using only common bio-chemical sensors such as ORP (Oxidation-Reduction Potential) and DO (Dissolved Oxygen). SBR (Sequencing Batch Reactor) is one of the most general sewage/wastewater treatment processes and, particularly, has an advantage in high concentration wastewater treatment like sewage wastewater. During the SBR operation, the operation status could be divided into normal status and abnormal status such as controller malfunction, influent disturbance and instrumental trouble. For the classification and diagnosis of these statuses, a series of preprocessing, dimension reduction using PCA, LDA, K-PCA and feature reduction was performed. Also, raw data obtained from SBR were transformed to synthetic data or fusion data and the performance were compared with each other. As the results, the fault recognition rate using fusion data showed the better result than that of raw data of [ORP] or [DO] and the combination method of K-PCA with LDA was superior to other methods such as PCA and LDA.

1 Introduction

The SBR has an advantage to carry out biological nitrogen removal in a single reactor by maintaining aerobic and anoxic stages sequentially. The critical control target in SBR operation is the determination of switching time of air supply/stop (mixing) and influent feeding. In many cases, SBR control strategies depend on ORP or DO values [1]. The instrument fault diagnosis using current signature, vibration, acoustic noise and temperature have been previously studied in several fields [2]. However these fault detection techniques need additional sensors and instruments. Related to modeling a non-linear system like wastewater treatment, we already showed that the knowledge-based system was useful for the classification and diagnosis [3]. If a reliable operation diagnosis using DO and ORP is developed, this technique will contribute to small-scale wastewater treatment plant operation without additional instrument.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 426 – 431, 2006.

2 SBR (Sequencing Batch Reactor) Process and Faults

2.1 SBR Process

The SBR (Sequencing Batch Reactor) is a water treatment system operating a fill-and-draw (batch) basis as shown in Fig. 1.

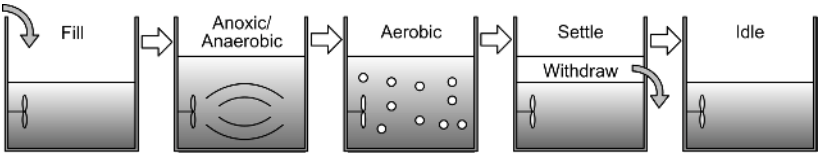


Fig. 1. The cyclic operation of the sequencing batch reactors

Although the basic principle of SBR, recently SBR system has been designed to adapt nutrient removal in wastewater. Jun *et al.* [4] showed the effect of fill ratio and aeration mode on nitrogen removal in swine wastewater treatment and suggested intermittent feeding of influent with the fill ratio of 1/12 at the anoxic step in sub-cycle operation (Fig. 2). This operation mode was especially useful for high nitrogen wastewater such as swine wastewater may cause toxic effect on biological reaction. As shown in Fig. 2, a whole-cycle of SBR consisted of 4 sub-cycles with 1hr anoxic and 3hr aerobic period. Because the most operation time is occupied with aeration time, the optimization and control of aeration are important. The full-scale SBR with effective volume of 20m^3 and digester of 30 m^3 for swine wastewater treatment were installed in Kimhae City. The profiles of DO and ORP were obtained from full-scale SBR. In previous study, the process control using ORP or DO was performed with threshold method including set point of dORP/dt or dDO/dt. However, since these set points are affected by reactor and influent conditions, a periodical fine tuning of set point was required for stable control [5]. This study includes the evaluation and diagnosis of set points in threshold type controller.

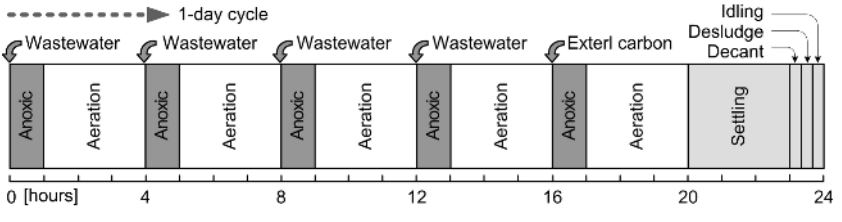


Fig. 2. Schematic diagram of overall SBR operating cycle

2.2 Fault Selection

For the purpose of this study, fault types in three categories-controller malfunction, influent disturbance and instrument trouble were selected. Table 1 lists the detailed definition of the selected malfunctions.

Table 1. Selected malfunctions

Location	Malfunctions		Fault no.
Set value in controller (dORP/dt)	high		F2
	low		F3
Influent	loading rate	extremely high	F4
		high	F5
		no feeding	F6
	quality	scraper type	F1(normal)
		slurry type	F7
Instruments	chemical pump trouble		F8

Aeration time in sub-cycle was controlled with threshold method using dORP/dt value as control parameter. If the set point of control parameter is too high, the mode change from aerobic phase to anoxic phase should be performed more rapidly and cause imperfect ammonium oxidation. In adverse, the under-evaluated set value results in over-aeration in aerobic phase. These problems are included into controller malfunction and a feed-back response against fault diagnosis is recommended. Fig. 3 show typical profiles of ORP and DO and fault cases during SBR operation.

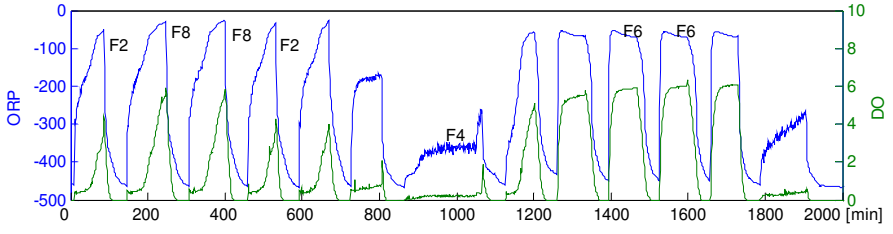


Fig. 3. ORP and DO profiles

3 Faults Diagnosis Algorithm

3.1 Configuration of Proposed Diagnosis Algorithm

Fig. 4 showed the diagram of fault diagnosis. ORP and DO values acquired from full-scale SBR were preprocessed by resampling, low-pass filtering and normalization. After FCM (fuzzy C-means) clustering, the normalized data were reduced by K-PCA (kernel principal component analysis) [8] and LDA (linear discriminant analysis) and then the feature vectors were produced. Using the Hamming distance measure, the test data were compared with the feature vectors and classified to each class from F1 to F8. Table 2 shows the data number for training and test, respectively. Here, the center vector for training was obtained by the clustering of several samples.

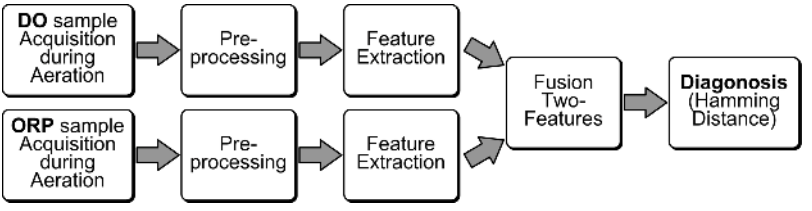


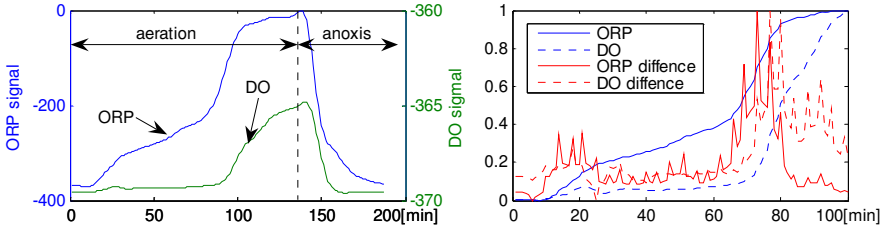
Fig. 4. Diagram of fault diagnosis

Table 2. The number of data

Object	F1	F2	F3	F4	F5	F6	F7	F8
Training (centers of cluster)	3	3	3	3	3	3	3	3
Testing	18	11	9	6	17	10	18	12

3.2 Preprocessing

The typical profiles of ORP and DO, obtained in a sub-cycle consisting of aeration and anoxic phase, were shown in Fig. 5 (a). As described above, the aeration phase plays the most important role in whole cycle and contains much information about the status of biological reaction. In this study, ORP and DO values in the aeration phase were applied to operation diagnosis.



(a) Original data of a sub-cycle (b) Preprocessed data

Fig. 5. Results of preprocessing

Because the aeration time in real plant is not fixed but varied with influent loading rate and operational condition, the dimensions of data had different range of time and needed to be normalized. The ORP and DO values in aeration phase applied to low-pass filter with 20% cutoff frequency and normalized to have the range of 0-1 value. Fig. 5(b) shows an example of the preprocessing results of ORP and DO values. For more enhanced performance, the differential data, obtained by following equation (1) using of ORP and DO, were applied to feature extraction process and compared with the results from raw data.

$$\Delta x = x(t) - x(t-1) \tag{1}$$

3.3 Feature Extraction

While the ORP values reflect the status of biological activity very well, they are too sensitive to system noise. On the other hands, since the DO values are affected by only oxygen concentration as the results of biological reaction, stable process control is possible, but only limited information as the diagnosis parameter are extracted. The basic idea of feature extraction is used on kernel PCA and LDA to map the input data into a feature space. The kernel PCA allows us to obtain the features with high order correlation between the input data samples. In nature, the kernel projection of data sample onto the kernel principal component might undermine the nonlinear spatial structure of input data. Namely, the inherent nonlinear structure inside input data is reflected with most merit in the principal component subspace [6]. Here, the applied kernel function is a Gaussian form as

$$k(x, y) = \exp\left(\|x - y\| / 2\sigma^2\right) \tag{2}$$

To extract feature, we uses LDA as well as a kernel PCA so as to examine the discriminative ability of kernel principal components. In brief, LDA calculate the optimal projection matrix to maximize the ratio of the determinant of the between-class matrix of the projected samples to the determinant of the within-class scatter matrix of the projected samples [6].

In this study, four data set of [ORP], [DO] and synthetic data of [ORP DO] and fusion data were investigated to apply for diagnosis and compared with results from their differential data, respectively. Here, the fusion data were obtained from the summation of feature vectors of ORP and DO.

3.4 Classification and Diagnosis

A hamming distance of the feature vectors between observed and training data was calculated and the observed data were classified to three classes having minimum distance.

$$\|x - y\| = \sum_{i=1}^{n} |x_i - y_i| \tag{3}$$

where, x and y are the feature vector of training data and test data, respectively.

Real scale SBR was operated during about one year and ORP and DO data were acquired every 1 minute through PC interface. During the operation, reactor status can be divided to 8 cases of F1-F8 as shown in Table 1. The F1-F8 statuses have 18, 11, 9, 6, 17, 10 ,18 and 12 data and the total number of training data were 101. Each status data were preprocessed and normalized to 100×1 vector as described earlier.

Table 2 shows the summarized diagnosis results for [ORP], [DO], [ORP DO] and fusion data when used raw data or differential data. As results, the diagnosis result using differential data was superior to that of raw data, and the fusion data show better results than other data. Also, the results of combination of K-PCA and LDA were better than those of LDA or (PCA+LDA). Here the clustering number of training data was 3 and the dimensions of projection of PCA, K-PCA and LDA were 14, 14 and 7, respectively. Finally, the fault recognition rate in case of using only ORP or DO was around maximum 97.03% and the fusion method showed better result of maximum 98.02%.

Table 2. Diagnosis results

	Feature Ext. method	Performance ORP	DO	[ORP DO]	Fusion
Data	LDA	96.04	97.03	93.07	94.06
	PCA+LDA	96.04	96.04	94.06	96.04
	K-PCA+LDA	96.04	97.03	94.06	96.04
Difference of data	LDA	94.06	95.05	97.03	97.03
	PCA+LDA	95.05	96.04	97.03	98.02
	K-PCA+LDA	95.05	97.03	97.03	98.02

4 Concluding Remarks

In this study, the fault diagnosis system for wastewater treatment plant was proposed using only common sensors such as ORP and DO. For the classification and diagnosis of these abnormal statuses, dimension reduction using PCA, LDA, K-PCA and feature reduction was performed. Also, raw data obtained from SBR were transformed to synthetic data or fusion data and the performance were compared with each other. As the results, the fault recognition rate using fusion data showed the better result than that of raw data of [ORP] or [DO] and the combination method of K-PCA with LDA was superior to other methods such as PCA and LDA. Moreover, the rapid fault detection by using only aeration phase signal made it possible to response against each fault case. The improved fault recognition rate in fusion data was thought to creative results in this study.

Acknowledgements

This work was supported by Ministry of Environment (071-041-060).

References

1. Norcross, K.L.: Sequencing Batch Reactors-An Overview. Wat. Sci. Tech. 26 (1992) 2523-2526
2. Grant, H.M., Charles, F.L.: A Novel Method to Detect Broken Rotor Bars in Squirrel Cage Induction When Interbar Current Are Present", IEEE Tran. Energy Con. 18(1) (2003) 71-79
3. Kwak, K.C., Ryu, J.W., Chun, M. G.: FCM-based Adaptive Fuzzy Inference System for Coagulant Dosing Process in a Water Purification Plant", Intl. J. Know-based Intel. Engin. Sys. 4(4) (2000) 230-236
4. Jun, B.H., Poo K.M., Choi E.H., Lee, H.I., Kim, C. W.: High-performance SBR Operation by Optimized Feeding Method of External Carbon Source for Piggery Wastewater Treatment. J. of Kor. Soc. of Environ. Eng. 24(11) (2002) 1957-1964
5. Jun, B.H., Kim D.H., Choi, E.H., Bae, H., Kim, S. S., Kim, C. W.: Control of SBR Operation for Piggery Wastewater Treatment with DO and ORP. J. of Kor. Soc. on Water Quality. 18 (2002) 545-551
6. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley Interscience. (2001)

On the Symbolic Analysis of Market Indicators with the Dynamic Programming Approach

Lukáš Pichl[1], Takuya Yamano[2], and Taisei Kaizoji[2]

[1] Division of Natural Sciences, International Christian University
lukas@icu.ac.jp
http://cpu.icu.ac.jp/∼lukas/
[2] Division of Social Sciences, International Christian University,
Osawa 3-10-2, Mitaka, Tokyo, 181-8585, Japan

Abstract. Symbolic analysis of time series of economic indicators offers an advantage of transferring quantitative values into qualitative concepts by indexing a subset of intervals with a set of symbols. In a similar way, computer codes routinely process continuous problems in a discrete manner. This work explains an appealing analogy between the DNA code of life and the symbol series derived from financial markets. In particular, it is shown that similarity scoring schemes and the alignment gap concept known in bioinformatics have even more natural and deeper analogies in the economic systems. The symbolic analysis does not solely mean a loss of information; in also allows us to quantify a similarity degree between various financial time series (and their subsequences) in a rigorous way, which is a novel concept of practical importance in economic applications. Our symbolic analysis concept is illustrated by two types of market indicator series, namely the analysis of Dow Jones vs. NIKKEI 225 indices on one side, and the CZK/EUR exchange rate vs. Prague money market rates on the other side. The present framework may also yield a significantly reduced computational complexity as compared to the neural networks in the class of similarity-comparison algorithms.

1 Introduction

It has been demonstrated in various research papers, and established in the finance literature over the last decade, that by using the time series of indicators on input, neural networks can be trained to predict both market trends and short-time price movements patterns [1]. Nevertheless, if we predict numerical values of the indicators rather than the direction of their change, the degree of accuracy is substantially lower [2]. This is, in part, due to incomplete modeling but also partial efficiency of the markets; bootstrapping of the time series allows us to distinguish between these two effects [3].

From another point of view, the fact that full numerical values of indicators (real numbers) are needed to predict partial changes (sign, trend) represents a deep imbalance between the information content of the data required on input, and the one obtained from data on output, which is well established in most quantitative forecast methods in various subfields of economics [4].

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 432–441, 2006.

In this work broadly inspired by computer science algorithms from the area of bioinformatics [5, 6], we develop a quantitative similarity degree between general time series of indicators, and present a sequence-matching algorithms for the symbolized sequences derived from the time series data. The bioinformatic concept of gap insertion (sequence comparison) and generalized scoring functions (symbol comparison) is elaborated upon. For the sake of clarity, we restrict ourselves to a 5-letter alphabet α in this paper, {F, D, N, I, G}, which represents the extreme fall, decrease, neutral trend, increase and extreme growth of each indicator on a session-to-session basis. The limits of $|\alpha| \to 1$ (identity) and $|\alpha| \to \infty$ (original series in a discretized numerical representation) clearly confine our symbolic analysis, and will be analyzed in a subsequent study. Let us note that the strings of symbols representing data series have one substantial advantage over the original numerical values: each particular letter in the alphabet represents a set of values (market states), which allows for a natural aggregation controlled by the size of the alphabet $|\alpha|$. The boundaries of particular interval bins can be easily fuzzyfied by applying continuous membership functions across interval delimiters, if further loosening of symbol membership relation proves advantageous [7].

The paper is organized as follows. Section 2 outlines two ways to define symbols: apriori definitions of data intervals vs. ex-post discretization schemes based on data histograms. Scoring schemes for symbol matching are tailored specifically for the indicator time series; we also explain the role and concept of gap alignment (time delay) in sequence comparison. Section 3 defines the similarity of symbol sequences and explains their actual alignment obtained with the dynamic programming algorithm. Applications to various time series of indicators from financial markets are analyzed in Section 4. Concluding remarks in Section 5 close the paper.

2 Symbolization and Scoring

This section explains the relation of our approach to standard algorithms in bioinformatics. First, the ex-ante and ex-post approaches to series symbolization are explained and then the scoring scheme for symbol comparison is developed.

Computer science in recent years has extremely elaborated on the problem of sequence comparison in the field of bioinformatics [5, 6]. Unlike from the usual text analysis, which is based on matching the same characters (e.g. the search functions implemented in wordprocessors), the comparison of two biological sequences (e.g. DNA) admits a character mismatch or insertion of a gap (blank) in either sequence over the four-letter alphabet {A,C,G,T}. The rationale of these concepts in biology are the errors in experimental methods (about 1-5 base readout errors per 100 nucleotides) and the occasional loss of some fragments (gaps in either sequence originating from reading intermissions or the presence of impurities in the gel in DNA reader devices).

Notably, there exist even better economic rationales regarding gaps and mismatches in pattern matching of indicator time series. A symbol mismatch is

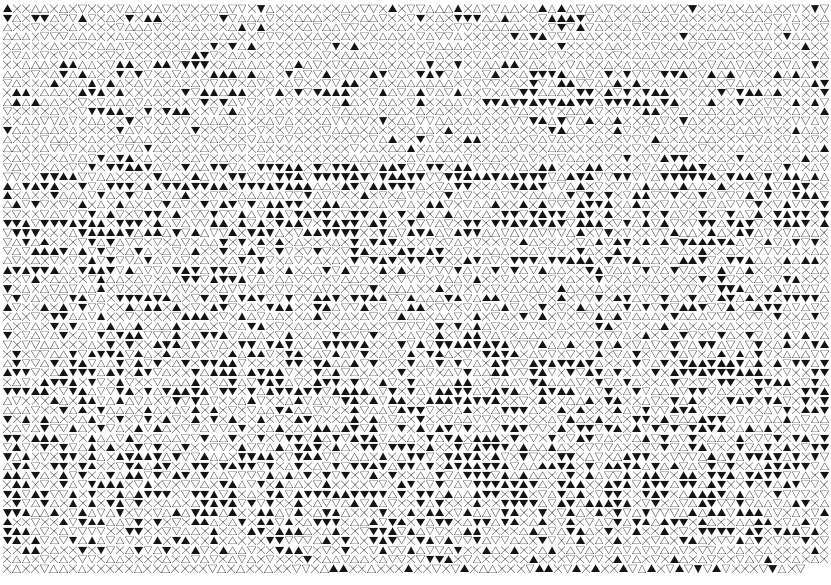


Fig. 1. NIKKEI 225 symbol series of normalized returns: the symbols label values of $\log(I_t/I_{t+1})$. The alphabet is denoted as D=$\bigtriangledown$, N=$\times$, I=$\bigtriangleup$, and full triangles for F and G, respectively (cf. Fig. 2 for the numerical data series).

defined as

$$s_1[i] \neq s_2[j], \quad i \in 1..T_1, \quad j \in 1..T_2, \tag{1}$$

while a gap means an insertion of a blank,

$$\begin{vmatrix} s_1[i] \\ - \end{vmatrix} \text{or} \begin{vmatrix} - \\ s_2[j] \end{vmatrix} . \tag{2}$$

In economics terms, a time delay (or response interval) of certain length between various indicators corresponds to the finite response time. This is embodied in the notion of the sequence gap, "−". A symbol mismatch between two aligned sequences may account for a presence of stronger market factor in one of the time series, which acts in a different direction than the analogous factor in the other time series (e.g. due to an incomplete theory). Such a mismatch then represents a situation in which the effect of this factor appears irregular with respect to the pattern searched. Unlike from bioinformatics where all mismatches $s_1[i] \neq s_2[j]$ are equally penalized, an economic mismatch has a more subtle point: for instance, a mismatch between N (neutral) and I (increase) symbols is certainly less serious than a mismatch between F (bearish) and R (bullish) symbols. Be a and b two letters in the alphabet α; the scoring scheme is denoted as $p(a, b)$. A standard scoring scheme in bioinformatics is [5]

$$p(a,b) = \begin{cases} 1 & \text{for } a = b \\ -1 & \text{for } a \neq b \end{cases} . \tag{3}$$

With respect to the economic quantities, a more appropriate scheme would be (mapping the alphabet α on the set of integers {-2,-1,0,1,2})

$$p(a,b) = \begin{cases} 0 & \text{if } i_a \text{ or } i_b \text{ equals } 0 \\ |i_a| & \text{for a match } i_a = i_b \\ -|i_a - i_b| & \text{for a mismatch } i_a \neq i_b \end{cases} \quad (4)$$

The above scoring scheme accounts for time delays by filtering the neutral trend (a =N or b =N) out (the first line in Eq. (4)). The penalty for gap insertion (cf. Eq. (2) is denoted by the letter g, $g < 0$. Reasonable scoring schemes require the matches to be preferred over mismatches, and any mismatch of the type $a \neq b$ to be preferred over an alignment of a with " $-$ ", and "$-$" with b (i.e., any mismatch score $p(\cdot,\cdot)$ should be larger than $2g$ in such a case). The maximum value of the scoring function is denoted by p_m in what follows.

Be $\{I_t\}_{t=1}^{T}$ a series of indicators, and $\alpha = \{L_i\}_{i=1}^{B}$ the alphabet consisting of $|\alpha| = B$ symbols. An indicator I_t is assigned a certain symbol L_i, iff

$$C_{i-1} < I_t \leq C_i, \quad (5)$$

where C_{i-1} and C_i are the delimiters of interval i, B is the total number of intervals (bins), and $C_0 = -\infty$ and $C_B = \infty$ formally. Alphabet symbols are assigned ex-ante, in case the delimiters C_t are defined without apriori knowledge of the data; if the symbols are assigned based on percentiles from data histograms, the time series are symbolized ex post. The applications in Section 4 are based on the latter method. Figure 1 shows the symbol time series of NIKKEI 225 index in the period of 4-Jan-84 to 24-Oct-05 (5367 values). Symbol N is used in 30% cases, D and I in 20% cases each, and F and G in 15% cases each.

3 Similarity and Sequence Alignment

This section briefly summarizes the two-step dynamic programming procedure [8], namely the recursive fill of a scoring matrix for substring alignments, and the reverse traceback procedure for generating the optimal alignment.

The alignment of two symbol series is defined as two sequences $S_1[t']$ and $S_2[t']$ ($t' = 1..T' \leq T_1 + T_2$), where each sequence $S_i[t']$ contains elements $s_i[t]$ in the same ordering as the initial sequence of symbols, except for possible gap insertions. The similarity measure of the two sequences is defines as the aggregate of similarity scores,

$$\text{sim}(s_1, s_2) = \sum_{t'=1}^{T'} p(S_1(t'), S_2(t')). \quad (6)$$

In the above equation, the penalty score of g is applied in case $S_1[t']$ or $S_2[t']$ is a gap (which is an extension to Eqs. 3-4).

The algorithm to find an optimal alignment of the two sequences in the above setting is well known in computer science. First, a scoring matrix of size $T_1 + 1$

by $T_2 + 1$ (rows 0 to T_1, columns 0 to T_2) is created and the 0-th row and 0-th column initialized with 0 scores. Then the scoring matrix $M[\cdot, \cdot]$ is recursively filled by

$$M[i,j] = \max(M[i-1,j] + g, M[i-1,j-1] \\ +p(s_1[i], s_2[j]), M[i,j-1] + g). \quad (7)$$

Depending whether the maximum in Eq. (7) was reached at the first, second, or the third argument, a pointer to $[i-1, j]$, $[i-1, j-1]$ or $[i, j-1]$ is stored in a tracking matrix T at position $T[i, j]$. The actual alignment is obtained at the same time by aligning $s_1[i]$ with $s_2[j]$ for a pointer to $[i-1, j-1]$, or by aligning $s_1[i]$ and gap, or $s_2[j]$ and gap in the other two cases. The traceback procedure starts at the largest element of the last row of the scoring matrix. Overlapping alignments are excluded by using a labeling vector. The standard algorithm from Ref. [5] has been extended here in order to list all possible alignments with a score higher than a desired threshold defined by $x((T_1)p_m + (T_1 - T_2)g)$ for $T_1 \geq T_2$. The above described procedure is quadratic in terms of memory complexity $(T_1 \times T_2)$; for sequences over the length of 10,000 we have also implemented a slower version of the dynamic programming (DP) algorithm, which is linear in memory space [9, 10].

4 Applications

In this section, the formalism developed above is briefly applied to two prototypes of market comparison problems. First, two stock exchange indices in the US and Japan are compared; second, empirical correlations are searched in the Czech foreign exchange and interbank money markets by using the DP method.

4.1 Stock Exchange Indices: NIKKEI 225 and Dow Jones

Figure 2 (a) displays the NIKKEI 225 stock-exchange time series from 4 January 1984 to 24 October 2005, the longest Japanese stock exchange stock market data series publicly available online from www.yahoo.com. Note the bubble burst in 1990, and the associated index volatility (or normalized returns on day-to-day basis) in the upper part of the figure. Historical data from the US Dow Jones Index (1 October 1928 to 24 October 2005) have been retrieved from the same source for comparison. The Dow Jones time series (with a window inset for the last 20 years) and the index volatility are plotted in Fig. 2 (b). The data are symbolized by using a 5-letter alphabet based on the procedure described at the end of Section 2.

In order to illustrate the power of our approach, we have selected an 8-day segment of NIKKEI 225 series from the bubble formation period (data starting from 8th of June 1987), s_1=GINDNIDF, and searched for the analogous pattern in the Dow Jones Index historical data. Table 1 shows the alignments obtained for the first 27 hits with the highest score, and their location in the Dow Jones Index sequence (symbol #). A scoring scheme with +1 for match, -1 for mismatch,

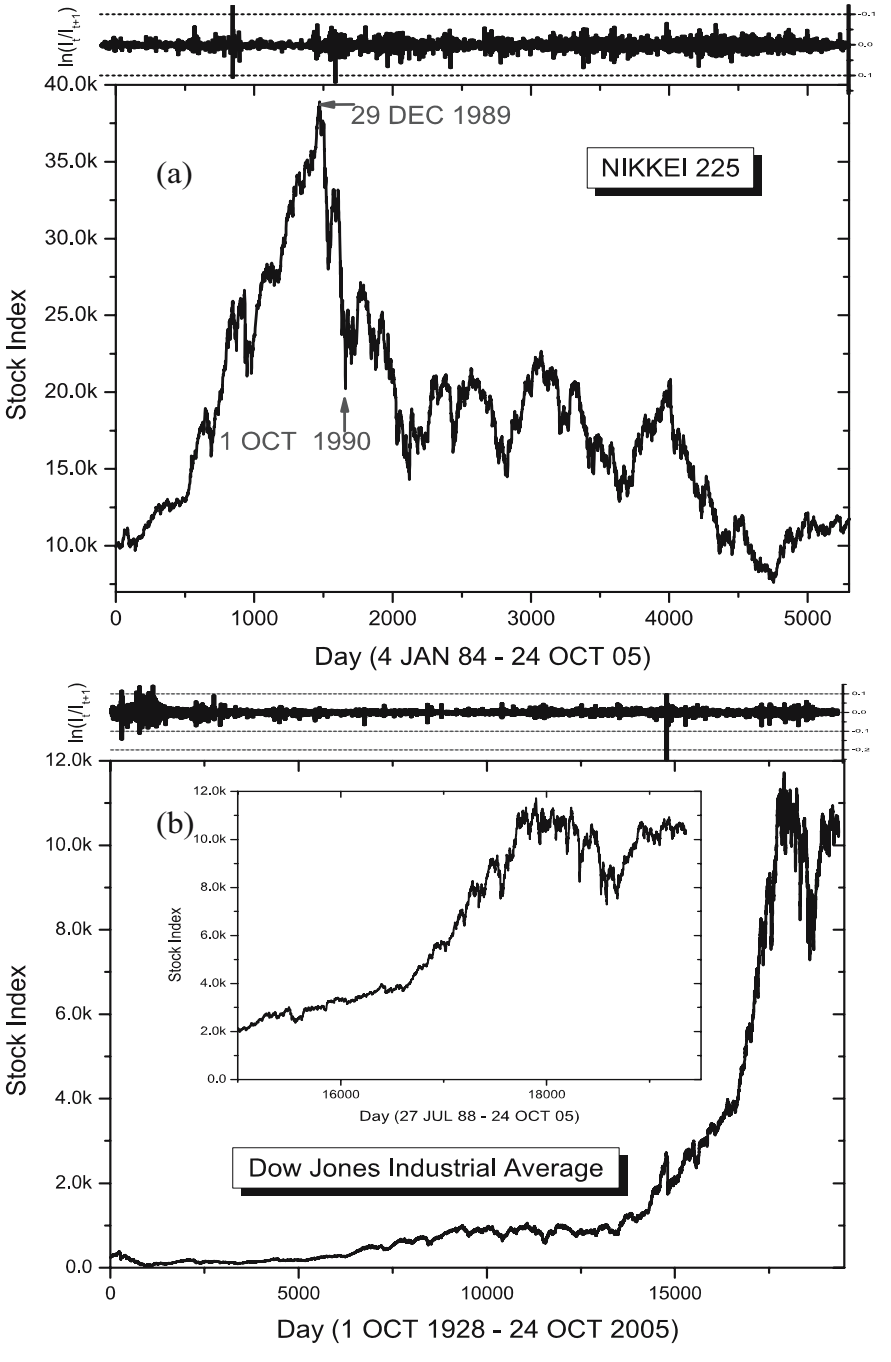


Fig. 2. Index time series: (a) NIKKEI 225 - 5,367 entries, (b) Dow Jones - 19,349 entries

Table 1. The alignment of 8-day NIKKEI segment starting from 8 June 1987 against the symbolized Dow Jones Index series between 1 October 1928 and 24 October 2005

Align	Hit	Align	Hit	Align	Hit
GINDNIFF	16145	GNI-DNIDF	2738	GINFDN-DF	10479
GINDNIDF	# 6.0	G-INDNIDF	# 4.0	GIN-DNIDF	# 4.0
GI-DNIDF	9546	DINDNGDF	4587	IINDNNDF	10984
GINDNIDF	# 5.5	GINDNIDF	# 4.0	GINDNIDF	# 4.0
GINDNGIDG	4654	NNNDNIDF	6331	GDNINIDF	11977
GINDN-IDF	# 4.5	GINDNIDF	# 4.0	GINDNIDF	# 4.0
DINDDNIDF	7062	GINDNDDN	6981	NIIDNIDF	12193
GIN-DNIDF	# 4.5	GINDNIDF	# 4.0	GINDNIDF	# 4.0
GINGDNFDF	9564	GGNDNNDF	7237	GINDIIDI	12476
GIN-DNIDF	# 4.5	GINDNIDF	# 4.0	GINDNIDF	# 4.0
GIDNGNIDF	18283	GINDNINN	7570	NINDNIDN	12822
GI-NDNIDF	# 4.5	GINDNIDF	# 4.0	GINDNIDF	# 4.0
GINDFIDI	2071	IINDNDDF	7780	GINDNNDN	13848
GINDNIDF	# 4.0	GINDNIDF	# 4.0	GINDNIDF	# 4.0
GI-DNIIDF	2173	GIFDIIDF	10099	DINDNINF	14236
GINDN-IDF	# 4.0	GINDNIDF	# 4.0	GINDNIDF	# 4.0
GIFFNIDF	2314	GIIN-NIDF	10212	GINDDIIF	14935
GINDNIDF	# 4.0	G-INDNIDF	# 4.0	GINDNIDF	# 4.0

and -1.5 for the gap penalty has been used in this case. Note that the first match, i.e. the alignment of s_1 with s_2's segments located at 16,145 and 18,283 correspond to a similar economic situation: sharp market rise before eventually strong corrections start to take place or the market bubble bursts.

4.2 Exchange Rate and Money Market Rate Correlations

The example in previous subsection illustrated the possibility of aligning symbol patterns for markets of the same type. Although NIKKEI 225 and Dow Jones indices were quite different, the symbolic analysis could unveil interesting similarities in the series patterns. Another application of practical interest is to study the time delay and propagation of policy effects among various markets. Here we opt to use the CZK/EUR foreign exchange market, and PRIBOR interbank money market as another example. Figure 3 shows the series of the two indicators over the same period of 154 months (January 1993 to October 2005). Let us note the dramatic spike in interest rates in May 1997 associated with a speculative attack and depreciation of the Czech crown (CZK). Both markets were highly correlated in this period as the Czech National Bank attempted to withdraw liquidity from the domestic financial sector and to block the currency attack. On the other hand, towards the end of the data series there is little correlation between both markets. This corresponds to the abolition of parity system for the Czech currency as a result of the speculative attack; also the currency started to continuously appreciate as the Czech economy integrates into the EU

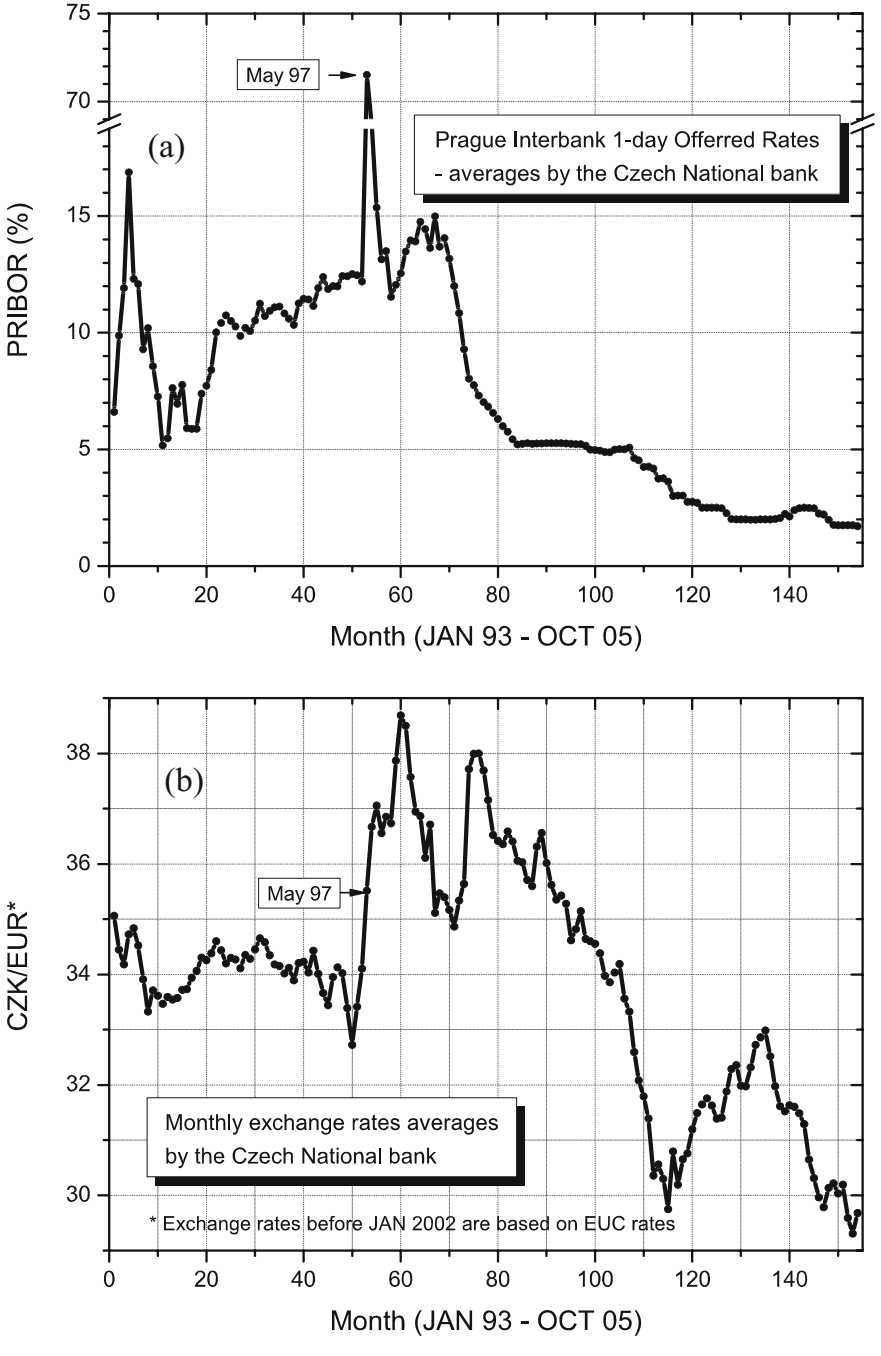


Fig. 3. Monthly averages of (a) 1-Day Money Market Rate and (b) CZK-EUR exchange rate (period of 12 years and 10 months)

structures. Within this heterogenous type of the two data series, our symbolic analysis can help us to study the time lags, with which the policy effects spread from one market to another.

The time series of indicators from Fig. 3 were symbolized using the technique described at the end of Section 2. The corresponding symbol sequences are shown in Table 2. It is interesting to note that a randomly selected pattern from the

Table 2. Symbolized CZK/EUR and PRIBOR time series

Exchange Rate Sequence:
FGDGNDFFINDNNNNNININNIDDNNDINIINDDNNNDI
NDIFDDGINFFGGGGIFINGGDFFNFGFINDDGIGINDF
FNNIDDNDNGIFDDNNFIIFNNDFNIIFDFFDFFIDFGF
GNGIINNDNGGNDNIGNNFFDNNNNDFDDDINDIFDG
Pribor Sequence:
FGGGFDFGFFFIGFGFNNGIGGIIDDDIDIGDIINDDDG
INDGIDINININDGFFFIFIIGINGDDGFIFFFFFDFDD
DFNDDDDDDNNNNNNNNNNNNNDDNDDNINNIFDFNDFN
NFNDFNNNDFFDNNDNNNNNIGDGIINDFDFFNNNND

PRIBOR time series before the speculative attack (9 month period starting from March 1996 to November 1996) aligns against the foreign exchange rate series after the 1997 crisis (9 month period from August 1998 to April 1999) with a high matching score:

Market	String	Location
PRIBOR	FINDDGIGI	38
FX CZK/EUR	GIN-DGIDI	67

This result has been obtained with the alignment equation (4). It demonstrates how the present algorithm can be able to filter out irregular periods, in this particular case the period of the 1997 crisis in the Czech Republic.

5 Concluding Remarks

In this paper, we have developed a symbolic analysis approach to comparing time series of indicators. The symbolic approach allows us to align subsets of the time series and rigorously quantify the similarity degree in such an alignment. It was demonstrated that the symbolic analysis is capable of cross-market similarity pattern recognition as well as filtering of irregularities and analysis of the time delays.

It is also worthwhile to note that the present formalism opens application perspectives even wider than those common in bioinformatics. Whilst biologists typically perform search for sequences representing a certain gene in the full

genome, it is also plausible in the economic analysis to search for market self-similarities over various time periods and tick frequencies (fractal analysis), vary the histogram periods in ex-post symbolization schemes, or study the effects of the size of the alphabet on sequence alignment. In brief, the symbolic method in economics is unrestricted as compared to the case of apriori given physiological meaning of biological sequences. Further applications are deferred to the next paper.

Acknowledgments

The authors would like to acknowledge a partial support by Japan Society for the Promotion of Science (JSPS) and Academic Frontier Program of Japanese Ministry of Education, Culture, Sports, Science and Technology (MEXT).

References

1. Trippi, R., Turban, E.: Neural Networks in Finance and Investing. McGraw-Hill/Irwin Publishing, New York (1996)
2. Moody, J: Forecasting the Eonomy with Neural Nets: A Survey of Challenges and Solutions. In: Orr, G. B. and Mueller, K. R. (eds.): Neural Networks - Tricks of the Trade. Springer, Berlin (1998)
3. Franke, J., Neumann, M. H.: Bootstrapping Neural Networks. Neural Computation **12** (2000) 1929–1949
4. Kilian, L., Taylor, M. P.: Why Is It So Difficult to Beat the Random Walk Forecast of Exchange Rates? Journal of International Economics **60**(1) (2003) 85–107
5. Setubal, J., Meidanis, J.: Introduction to Computational Molecular Biology. PWS Publishing Company ITP, Boston (1997)
6. Bergeron, B.: Bioinformatics Computing (Pearson Education, Prentice Hall PTR, Indiana (2003)
7. Bufardi, A.: On the Fuzzyfication of the Classical Definition of Preference Structure. Fuzzy Sets and Systems **104**(2) (1999) 323–332
8. Bellman, R.: Dynamic Programming. University Press, New York (1957)
9. Myers, E. W., Miller, W.: Optimal Alignments in Linear Space. CABIOS **4** (1988) 11–17
10. Huang, X., Hardison, R. C., Miller, W.: A Space-Efficient Algorithm for Local Similarities. CABIOS **6** (1990) 373–381

Neural Network Method to Predict Stock Price Movement Based on Stock Information Entropy

Xun Liang[1,2]

[1] Institute of Computer Science and Technology, Peking University, Beijing 100871, China
[2] Department of Economics and Operations Research, Stanford University, Palo Alto, CA 94305, USA
liangxun@icst.pku.edu.cn

Abstract. In this paper, the neural network is employed to learn the complicated association of the stock information stream with stock price movement. In general, the positive stream of stock information is assumed to stimulate buying and increase the stock prices, and the negative stream of stock information is assumed to result in selling and decrease the stock prices. Based on the stock information entropy, the intensity for stock information stream is measured and applied to associating with the stock price movement with the aid of neural networks. Experiments illustrate the associations statistically. The results are helpful in probing the microstructure of the stock markets from a new angle.

1 Introduction

In finance, the stock price prediction based on stock information is an important financial subject that has attracted attentions of numerous scientists and financial practitioners for many years. As it is well-known, the stock market is a very complex system and there are plenty of factors influencing it. Even so, it is commonly believed that the stock information plays an outstanding role in shifting the market and acts as a hinge through which the information implements its leverage function on the market. From one aspect, it involves an assumption that the past publicly available information is predictive to future stock returns [7]. Consequently, it is never overemphasized to reiterate the functions of stock information.

Various kinds of information, that influence the physiology of investors, are disseminated to the stock market and motivate them to make decisions to trade. In general, positive (good) information is assumed to stimulate buying, increase the stock prices, and move the market up. In contrast, negative (bad) information is assumed to result in selling, decrease the stock prices, and move the market down. The samples of such information include economic variables such as interest rates and exchange rates, industry specific information such as growth rates of industrial production and consumer price, and company specific information such as dividend yields, earnings announcements, annual reports announced by companies, the consumer price index (CPI), rumors, and emergency events in the world.

For the relations between the stock information and the stock prices, most of papers focus on the stock market responses for particular stock information. Clearly, a

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 442 – 451, 2006.

more comprehensive study of the impacts by stock information should not only consider each piece of information, but also taking the total intensity of overall stock information stream into account.

Before the era of the internet, the stock information was only published on the newspapers or broadcasted on radios and televisions, and people can read or listen to a very limited amount of the stock information each day. Nowadays, the stock information permeates everywhere on the webs. On this condition, the direct measure of financial information intensity stream on a daily (or even a shorter time period) basis is beyond our ability. With the aid of the internet, the computers can process millions of pieces of the stock information extremely conveniently because the web lodges oceans of stock information and feeds the stock information to our computer terminals through the internet carrier at ease. Although the reports on how the information influences the stock market have been quite a few, the research on the impacts of stock information in the cyberworld on the stock market is still little [2][6][9].

Comparing with the European and American stock markets, the Chinese stock market is still a teenager, and [6] listed many aspects that the Chinese stock market is different from mature markets. A conclusion in [6] implies that those special features in China make her market more coupling to the stock information. This paper offers an extension to the literature [6] by applying the entropy theory to reflect the rarity or popularity of a piece of information. The theory suggests that the information with less probability to occur insinuates a larger impact on the stock market.

In the area of financial time series analysis, many results based on neural networks have been achieved [3]. This paper contributes to the field from the new aspect by studying the financial information in the internet instead of the traditional medias like newspaper, televisions. As a result, more stock information could be collected automatically by computer harvesters and processed, as opposed to the handling of a particular information.

The remainder of this paper is as follows. Section 2 states the process of understanding stock information on web based on the natural language processing technique. Section 3 exploits the associations of stock information with the stock market in experiments based on neural network. Section 4 concludes the paper.

2 Parsing Web Stock Information Based on Natural Language Processing

2.1 Harvesting Stock Information on Web

The procedure of interpreting the web stock information is divided into two phases. First, we need to harvest the web stock information in the cyberworld, and save it into the database. Second, the work of natural language processing should be done.

Our computer harvesters are coded in Java language and run on the J2EE platform. The computer harvesters collect some 200 principal Chinese financial information websites. For example, www.sic-cfi.com, www.cfi.net. cn, www.cnlist.com, www.chinabond.com.cn, www.cnfund.cn/jjcs.asp.

The following is a sample webpage of Chinese information harvested from the internet on 2004-11-26 for a Chinese IT firm named Tongfang. The information is a

piece of typical bad information [1], stating that the IT industry in China in year 2005 is predicted worse. The information is as follows.

<Time>2004-11-26</Time>
<Title>the IT industry in China in year 2005 is predicted worse [2]</Title>
<Media>Journal of Chinese Market</Media>
<Author>Xu Yaping</Author>
<Text>Six leading companies predicted that the IT industry would be worse in 2005…</Text>

In the beginning of the information, the published time, the title of the article and the author are listed in the brackets of <> </>. Then the text body follows.

2.2 Strength of Stock Information

In order to process the texts in the Chinese natural language in each piece of information, we need to have the Chinese Word-and-Phrase Dictionary. A typical part of the dictionary, produced by the Institute of Computing Technology, Chinese Academy of Sciences, is exemplified as follows.

share holder n 40
share holder meeting n 1
share n 71
system of stock shares n 142
interest on stocks n 1
...

where n denotes noun, v denotes verb, and the numbers behind n and v are the occurring frequencies q_j ($j \in J$ where J is the set of word indices) based on the statistics on principal Chinese newspapers.

Another Chinese dictionary we hold to process the texts more precisely is the *Chinese Word-Association Dictionary*, where two words are listed and those two words are frequently used in the same sentence. The word combinations and their statistical frequencies form the above dictionary. A typical part of the dictionary, also produced by *the Institute of Computing Technology, Chinese Academy of Sciences*, is exemplified as follows.

stock market@long term 2
stock market@growth 1
stock market@finance 1
stock market@appear 3
stock market@in a state 1
stock market@create 1
stock market@rebound 2
stock market@repeat 1
stock market@in succession 1
stock market@management 1
…

In addition, we use the C++ open codes, produced also by *the Institute of Computing Technology, Chinese Academy of Sciences*, to process the Chinese texts of the stock information on web.

The computer program first implements the divisions of Chinese word-phrases for the web stock information.

---

[1] Taking into account the current advances of the natural language processing techniques, it should be an adequately satisfactory result if the computer can understand the typical bad or good news.

[2] The original language on the web is Chinese. The remainder of the paper about stock news on the web and its processing are all originally in Chinese and for the Chinese language processing.

The third dictionary is the user-dictionary. In this paper, about 430 typical Chinese "stock-related" keywords, associating with the corresponding emotion grade η_j in the interval [0,1], are also selected by us in advance for examining optimism or pessimism of stock information. They are grouped into the user-dictionary.

Based on the above three dictionaries, we can program in C++ and make the computer "understand" the web stock information. The computer program examines the texts of web stock information and selected the words in our user-dictionary. The words and frequencies u_j are then found for the above information, listed as follows.

Group 1 (optimism) ($J_1 \subset J$)
predict a good market 3
increase 1
positive 1
optimism 3
...
Group 2 (pessimism) ($J_2 \subset J$)
complaint 1
fiasco 1
pessimism 7
ruin 1
trap 1
...
Group 3 (negation) ($J_3 \subset J$)
not 8
...
Group 4 (grade) ($J_4 \subset J$)
very 8
excess 1
extraordinary 18
...
Group 5 (range) ($J_5 \subset J$)
all 7
in succession 1
...
Group 6 (duration) ($J_6 \subset J$)
next year 11
this year 4
...

Clearly, words and their frequencies show that the corresponding information is very bearish information.

Intuitively, the impacting strength on the stock market of the information is not too weak according to η_j, u_j and q_j. Based on the optimistic and pessimistic words and their frequencies in Groups 1 and 2 comparing to the frequencies in the dictionaries by the NLP technology, the strengths are valued by the arithmetic calculations. For stock i, the strength is $v_i = \sum_{j \in J_1} \eta_j \frac{u_j}{q_j} - \sum_{j \in J_2} \eta_j \frac{u_j}{q_j}$. A value of -0.02 for the information strength is given for the above example.

In summary, the variable of strength of information is computed based on the frequencies of the financial word phrases and also the strength by those word phrases. The larger the absolute value of the information strength, the heavier the information impacts on the stock market intuitively.

2.3 Intensity of Stock Information Based on Entropy

However, the impacts of the information is not solely decided by the value of strength of stock information. Instead, it is dependent on a number of factors. In our paper, we additionally consider the rarity of the information. If such kind of the information rarely occurs, it normally has more impacts on the stock market. In contrast, if the similar information occurs repeatedly in the historical observation period, it normally has less impacts on the stock market. The larger the value of rarity, the less frequently the similar information occurs.

For convenience, we use the variable *popularity*, as opposed to the term *rarity*. The popularity of the information is based on the probability that the similar bullish or bearish information occurs in the historical observation period. The larger the popularity value, the more frequently the similar information occurs. For example, the popularities of the variety of information could be [very bearish, bearish, mean, bullish, very bullish] = [0.15, 0.27, 0.29, 0.20, 0.09], respectively.

Let $S = \{ s_n : n = 1, \ldots, N \}$ be a set of events, with occurring probability for event s_n as $0 \le p_n \le 1$ where $\sum_{n=1}^{N} p_n = 1$. The entropy for event s_n is defined as $I(s_n) = -\log p_n$. Clearly, the smaller p_n, the larger $I(s_n)$ is. This is consistent with our intuition. The less the event s_n occurs, or the smaller the probability p_n, the more information it carries when occurring, namely the larger $-\log p_n$ is. For example, for the small firms in the stock market, since it is normally given less attentions, there is less information for its stock. Once a piece of information for this stock arrives, the response of its stock price is more likely to be vehement, illustrating that this stock information holds more information.

The popularity is then transformed into the entropy. In the above example, the entropies for them are derived from taking the negative logarithms with the base 2, which are [2.737, 1.889, 1.786, 2.322, 3.474]. The larger the entropy value, the heavier the information impacts on the stock market. In the above example, since it is a very bearish information, the entropy is 2.737.

The term of strength of information does not duplicate the term of the popularity and entropy of information. For example, assume a piece of information is good information with many "bullish" words. As a result, the strength of this information is large. However, there is so much bullish information in that period of time. Consequently, the popularity of the information is large, too, and the entropy is small. The total result of the information may not be large.

As a result, the intensity of information θ_i is found by multiplying the strength of information and its entropy, namely θ_i = -0.02 * 2.737 = -0.055, and put into the corresponding XML tag, <InformationIntensity>-0.055</ InformationIntensity>.

2.4 Contents of Other Tags

Likewise, the duration impacted d_i can be calculated based on the enumeration of the words in Group 6, and we obtain <DurationImpacted>0.72</DurationImpacted>. The range of investors that the information impacts is scaled into [0,1]. Since the information comes from the *Journal of Chinese Market*, the newspaper about the market with a relatively wide-range of readers in China, the CirculationRange of c_i = 0.5 is assigned, <CirculationRange>0.5</CirculationRange>. There is a list for all kinds of media subjectively given in advance, and the corresponding CirculationRanges can be found in the list. In addition, the status like the bear or bull market and the status in the corresponding industry are marked with a number in [-1,1]. In 2004, the Chinese stock market in general is bearish. As a result, the StockMarketStatus is set to be a_i = -0.5, and <StockMarketStatus>-0.5</StockMarketStatus>. Also, the IT industry is bearish, and the IndustryStatus is set to be b_i = -0.3, and <IndustryStatus>-0.3 </ IndustryStatus>.

3 Experimental Study

3.1 Data Set

The above example is just one piece of information in the Tongfang's web information set. In this section, we train neural networks by a relatively larger set of examples. Totally, some 500 pieces of information were harvested in our study. The data are divided into 2 sets. The training pattern set consists of 420 pieces of information and the testing pattern set consists of some 80 pieces of information. It is admitted that the data set is not very large. The internet usage in China is not as popular as in the developed countries. However, as an exploratory subject, it is still worthy to investigate it.

3.2 Psychology-Delay

The stock returns during the corresponding period are also collected for observations. We also notice that on day of 2004-11-26 the intensity is -0.055 and the actual stock return was -0.006 (or the stock price was decreased by 0.6%). The stock prices therefore moved to the correct directions as anticipated.

Noticing that at the time when the stock information publishes, several scenarios may happen. First, the stock information publishes after the close of the stock market. In this case, the quickest response for the investors has to be delayed to the next trading day. Second, a portion of investors may not receive the information instantaneously, e.g., they may be in the meeting. Third, some investors may hesitate to trade, leading to a psychology-delay, before they make a decision. Moreover, the financial experts normally produces the anticipation for some stock information, for example, the earnings announcements, and the market already absorbed the information partially somewhat early. In that case, it may therefore be reasonable that the psychology-delay is negative. This is consistent with the hypothesis of semi-strong form in financial theory. In summary, we assume that the duration of psychology-delay is $\{-\xi, \dots, \psi\}$, where $0 < \xi, \psi$. The existence of psychology-delay yields to a new observation period as $\{-K+s, \dots, -1+s\}$, where $s \in \{-\theta, \dots, \psi\}$. Let $p_i(k)$ be the stock price for firm i on day k, $k = -K+s, \dots, -1+s, s$. Let $\Delta p_i(k) = p_i(k) - p_i(k-1)$ be the absolute value of stock price changes for firm i on day k, $k = -K+1+s, \dots, -1+s, s$.

We attempt to find the maximal absolute stock price change in the observation period. Let S be the index set for this maximal change $|\Delta p_i(s)|$, $s \in \{-\theta, \dots, \psi\}$, $S' = \{ s : \arg \max_{s \in \{-\xi, \cdots, \psi\}} \{ |p_i(s) - p_i(s-1)| \} \}$.

S' may hold more than one elements. For example, if $|p_i(s_1)-p_i(s_1-1)| = |\Delta p_i(s_2)-p_i(s_2-1)|$, $s_1 \neq s_2$, then, s_1 and s_2 both are in S'. We are only interested in the index with the smallest absolute value since this value is mostly associated with the stock price changes, hence $S = \{ s : \min_{s \in S'} \{ |s| \} \}$ is obtained. Clearly, S is non-empty. Sometimes, there are two elements in S - in this case, one must be positive and the other one must be negative, and the positive one s^* is regarded as the best one. Finally, we obtain the optimal index $s^* \in S$.

3.3 Entropy and Intensity

In our experiment, the popularity of the variety of information is expressed in probabilities [very bearish, bearish, mean, bullish, very bullish] = [0.15, 0.27, 0.29, 0.20, 0.09], respectively. As a result, the entropies are [2.737, 1.889, 1.786, 2.322, 3.474].

We have some cases for the different numbers of pieces of web stock information on one day. If there is only one piece of web stock information on some day, the stock

Table 1. Samples of information intensities

date	information (#1)			information (#2)		
	strength	entropy	intensity = strength*entropy	strength	entropy	intensity
08-02	0.5	2.322	1.161	0.005	1.889	0.009
08-10	0.01	2.322	0.023			
09-10	-0.005	1.786	-0.009			
09-20	0.005	2.322	0.012			
09-29	-0.01	2.322	-0.023	-0.02		
10-12	0.005	2.737	0.014	0.002	2.737	0.005
10-15	0.005	3.474	0.017			
11-18	0.02	1.889	0.038	0.01	1.786	0.018
11-22	-0.01	2.322	-0.023			
11-24	0.005	2.322	0.012			
11-26	-0.02	2.737	-0.055	0.002	2.322	0.005
11-30	-0.005	2.322	-0.012			

information (#3)			intensity
strength	entropy	intensity	
-0.01	2.322	-0.023	1.147
			0.023
			-0.009
			0.012
0.005	2.322	0.012	-0.012
			0.019
			0.017
0.005	3.474	0.017	0.073
			-0.023
			0.012
			-0.05
			-0.012

return on that day will be used in the experiments directly. If there is more than one piece of web stock information on the same day, the intensities of the all the information on that day are aggregated before associating them with the stock market (see Table 1 for some examples).

3.4 Training by Neural Networks

Like the architecture of neural networks in [7], we also consider the impacts of all the L days in the period and apply a sliding window. As a result, for stock i, The input vector of the neural network becomes $X_i(k) = (\theta_i(k+1), \theta_i(k), \theta_i(k\text{-}1), \ldots, \theta_i(k\text{-}L+1), d_i(k), c_i(k), a_i(k), b_i(k), r_i(k), r_i(k\text{-}1), \ldots, r_i(k\text{-}L+1))^{\tau} \in \boldsymbol{R}^{(2L+5)\times 1}$ where $(\bullet)^{\tau}$ denotes the transpose of $(\bullet)$, and the output is $r_i(k+1) \in \boldsymbol{R}$. This works better since using the older information takes into account the influence of the past information intensities.

The neural network learns the mapping without any prior knowledge. If the neural network cannot learn the patterns, a new hidden neuron is added. At the time when the mapping is learned, $H = 107$. Then we try to prune away some hidden neurons based on the orthogonal projection methods [4][5]. Finally, the neural network with $H = 42$ is obtained.

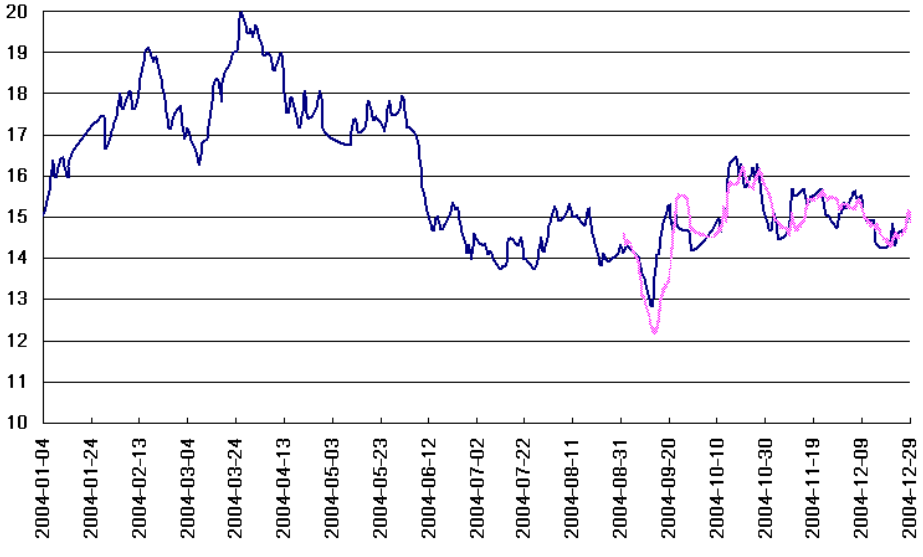


Fig. 1. The actual and predicted stock price movements for the firm Tongfang. The dark line is the actual stock price movement and the shallow line shows the price movement given by the trained neural network. From 2000-01-01 to 2000-08-31, the data are training data. From 2000-08-31 to 2000-12-29, the data are testing data. The prices are in RMB.

The parameter L in the input patterns is selected experimentally. If L is too small, it may not convey enough older information; if L is too large, it leads to unnecessarily large number of input ports and makes the network superfluous. In experiments, it appears to us that $L = 3$ (in general a piece of the information will not influence the market more than 3 days) gives the satisfactory results. After the neural network is

trained, we predict p_i'(k+1) on a day-by-day basis. For example, on the testing day 2004-11-26 (for simplicity we write k+1 = 2004-11-26), we know $X_i(k)$ and also $\theta_i(k+1)$ on the testing day. Hence p_i'(k+1) is computed based on the trained neural network. On the next trading day, let k = 2004-11-26 and k+1 = 2000-11-29. No matter the network predicts correctly or not, the actual $p_i(k+1)$, as opposed to the predicted one p_i'(k+1), is transformed and rolled into the input vector $X_i(k)$. This is because on the next testing day, the real $p_i(k+1)$ occurred in the last day is already known.

The performance of the trained neural network is shown in Fig.1.

3.5 Discussions

The dynamic input fashion is adopted and the information intensity based on entropy, as opposed to the information strength [6], is utilized to allot the appropriate impact for the stock information, For comparison, the correct predictions of return signs in this paper is 54 trading days, and this gives to an overall correct rate of 65.9%, enhanced by some 6% over the result in [6]. [6] also provides experiments for predicting the absolute amplitudes of stock price changes although no change directions are stated and comparable. The absolute predicting errors in [6] are less than 28%, while the same measure in this paper is less than 13%. The reason that prediction rate is largely improved may owe to the introduction of much more semantic information on web and also a better NLP program. In addition, the sole-volume-based amplitude estimation approach may accumulate the effects of two pieces of information with opposite emotions which should be cancel each other if correctly handled.

4 Concluding Remarks and Future Work

The stock information intensity is modeled based on stock information entropy and the NLP technology. The information intensity is related to the stock price movement and also provides the clues to the shifting of the stock market. The trained network accomplishes the mapping-learning task and establishes the association of the stock information stream with the stock price movements with certain acceptable precisions.

Our results also raise more interesting issues for the future research. Firstly, in the bear market, bad information influences the market more severely than good information, or the *asymmetry* phenomenon should be addressed separately for the bear market. As a matter of fact, the introduction of entropy makes this work pretty simple - a weighted entropy can be employed to reward the bad information a larger value and the good information a smaller value. Secondly, we need to consider the orders and distances between the financial words and phrases. Thirdly, the uncertainty for the whole stock information system shall be depicted by the information entropy and shall offer us an insightful contour for the microstructure of stock market.

Acknowledgment

The Project sponsored by NSF of China under grant number 70571003.

References

1. Antweiler, W., Frank, M.Z.: Is All That Talk Just Noise? The News Content of Internet Stock Message Boards. Journal of Finance 59 (2004) 1259-1294
2. Das, S.R., Chen, M.Y.: Opinion Extraction from Small Talk on Web. Technical Report, Santa Clara University (2001)
3. Harrald, P.G., Kamstra, M.: Evolving Artificial Neural Networks to Combine Financial Forecasts. IEEE Trans on Evolutionary Computation 1 (1997) 40-52
4. Liang, X., Xia, S.: Methods of Training And Constructing Multilayer Perceptrons with Arbitrary Pattern Sets. Int Journal of Neural Systems 6 (1995) 233-247
5. Liang, X.: Removal of Hidden Neurons by Crosswise Propagation (CP). Neural Information Processing 6 (2005) 79-86
6. Liang, X.: Impacts of Internet Stock News on Stock Markets Based on Neural Networks. In: Wang, J. et al (eds.): Advances in Neural Networks. Lecture Notes in Computer Science, Vol. 3497. Springer-Verlag, Berlin (2005) 897-903
7. Narendra, S., Parthasarathy, K.: Gradient Methods for Optimization of Dynamical Systems Containing Neural Networks. IEEE Trans on Neural Networks 2 (1991) 252-262
8. Sicilia, D.B.: The Greenspan Effect: Words That Move the World's Markets. McGraw-Hill, New York (1999)
9. Wysocki, P.D.: Cheap Talk on Web - Determinants of Postings on Stock Message Boards. Technical Report, University of Michigan (1999)

Stock Time Series Forecasting Using Support Vector Machines Employing Analyst Recommendations*

Zhi-yong Zhang[1,2], Chuan Shi[1,2], Su-lan Zhang[1,2], and Zhong-zhi Shi[1]

[1] Institute of Computing and Technology, Chinese Academy of Sciences, Beijing 100080
[2] Graduate School of the Chinese Academy of Sciences, Beijing 100039, China
{zhangzy, shic, zhangsl, shizz}@ics.ict.ac.cn

Abstract. This paper discusses the application of support vector machine (SVM) in stock price change trend forecasting. By reviewing prior research, thirteen technical indicators are defined as the input attributes of SVM. By training this model, we can forecast if the stock price would rise the next day. In order to make best use of market information, analyst recommendations about upgrading stocks are employed. So we put forward an improved method to evaluate if an upgrade classification of SVM is reliable. In our method, recommendation accuracy is first calculated according to historical advice. Then the more objective relative accuracy is deduced by considering the influence of total stock market index. Moreover, improved model is examined with the real data in Shanghai stock exchange market. Finally, we discuss some interesting hints to help readers understand this model more explicitly.

1 Introduction

SVM offers remarkable generalization performance in many areas such as pattern recognition, text classification and regression estimation. Recently, some researchers deal with the application of SVM in time series forecasting. They concern about how to make precise value prediction by means of support vector regression. However this is difficult since stock time series are random walk and non-stationarity. Obviously, we can make profit if we can forecast the direction of change of stock price. In this paper, we attempt to achieve this by using support vector classification.

As Einstein told Born, you believe in a God who plays dice, I believe in complete law and order [6]. If we could know the predictive model and all parameters, the stock price could be exactly predicted. However that is impossible in reality. News articles are adopted to make up the lack of stock market information [3]. But text-mining technique is still not mature to extract valuable and concise information. Moreover, authors are not experts, even cannot correctly using terms and expressing information. Third, analyst recommendations are more authoritative since analysts understand the stock market better. Also, the data are often structurally expressed.

The rest of the paper is organized as follows: In Section 2, we discuss some key factors, input attributes of *C*-SVC and the methods of using analyst recommendations. The experimental results are shown in Section 3 and concluded in Section 4.

* This work is supported by the National Science Foundation of China (60435010), National Great Basic Research Priorities Programme (973 Program: Grant No.2003CB317004) and the Nature Science Foundation of Beijing (4052025).

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 452 – 457, 2006.

2 Improved Trend Prediction Model

2.1 Some Key Factors About *C*-SVC

One main factor when using SVM is how to select kernel function. Linear kernel, polynomial kernel, RBF and sigmoid kernel are usually adopted. In this work, we use general RBF kernel function with two parameters: C and γ [1]. The accuracy of SVM depends on selecting appropriate parameters in training procedure. The cross-validation is preferable for this procedure [5]. In v-fold cross-validation, we divide the training set into v subsets of equal size. Sequentially one subset is tested using the classifier trained on the remaining v-1 subsets. Thus, each instance of the whole training set is predicted once so the cross-validation accuracy is the percentage of data that are correctly classified. This can also effectively avoid the over-fitting problem.

In order to approximate the cross-validation rate, we also use grid-search, an automatic grid and pattern search approach, iterating through ranges of parameters to find the optimal parameter values [5]. Though it is time-consuming, this approach does an exhaustive parameter search. And the computational time is not much more than other advanced methods since there are only two parameters. Furthermore, the grid-search can be easily paralleled because each (C, γ) is independent.

2.2 Input Attributes of *C*-SVC

By reviewing prior research, thirteen technical indicators, presented in Table1, can be used as the input attributes of *C*-SVC [2]. Where C_t, L_t and H_t are the closing price, the low price and the high price at time t, respectively. MA_t is the moving average of closing price during the last t days and is defined as $\sum_{i=1}^{n} C_{t-i+1} / t$.

2.3 Employing Analyst Recommendations

By training *C*-SVC, we can forecast if the price of a stock will rise in next day. Then how can we estimate the reliability about an upgrade prediction? It is available that an analyst familiar with a stock advises its price will upgrade. In this subsection, we attempt to distinguish which predictions are reliable in recommendations periods --- the special terms in time series. There are many types of recommendations, such as stock target price and upgrading stocks advice [7]. In this paper, they are upgrading stock advice in a short or middle term. Professionally [8], we define the short terms as from next day to 45th natural day, and the middle terms from the 30th to the 120th.

Obviously, analyst is familiar with the forecasted stock is very crucial. If he has accurately advised the rise of the stock price, his recommendations and the stock price can be considered as positive correlation, or else being inverse correlation. Therefore, recommendation accuracy is first evaluated according to the stock price in his former advice term. In addition, total stock market index may produce disastrous influence on stock price. For example, most of stocks will downgrade when total stock market index drops. Hence we can consider it as accurate with rise or even unchanged price vs. downgrading total market index. Thus we will calculate relative recommendation accuracy and eliminate market influence.

Table 1. The following 13 technical indicators are calculated with daily price features. By training *C*-SVC with these 13 input atributes, the trend of stock price will be forecasted.

Attribute	Formula	Description
DateID	t	The $t-th$ day in the time series.
%K	$(C_t - LL_{t-n})/(HH_{t-n} - LL_{t-n}) \times 100$	Comparing a security's price closed relative to its price range over n days. Where LL_t and HH_t mean the lowest price and highest high price during the last t days.
%D	$\sum_{i=0}^{n-1} \% K_{t-i} / n$	Moving average of %K.
Slow %D	$\sum_{i=0}^{n-1} \% D_{t-i} / n$	Moving average of %D.
Momentum	$C_t - C_{t-4}$	Measuring the amount change of a security's price over four days.
ROC	$C_t / C_{t-n} \times 100$	Price rate-of-change. It displays the difference between the current price and the price n days ago.
Williams' %R	$(HH_{t-n} - C_t)/(HH_{t-n} - LL_{t-n}) \times 100$	A momentum indicator measuring overbought/oversold levels.
A/D Oscillator	$(H_t - C_{t-1})/(H_t - L_t)$	A momentum indicator associating changes in price.
Disparity5	$C_t / MA_5 \times 100$	The distance of current price and the moving average of 5 days.
Disparity10	$C_t / MA_{10} \times 100$	10-day disparity.
OSCP	$(MA_5 - MA_{10}) / MA_5$	The difference between two moving averages of a security's price.
CCI	$(M_t - SM_t)/(0.015D_t)$ $where \quad M_t = (H_t + L_t + C_t)/3$ $SM_t = \sum_{i=1}^{n} M_{t-i+1} / n$ $D_t = \sum_{i=1}^{n} \lvert M_{t-i+1} - SM_t \rvert / n$	Commodity channel index. It measures the variation of a security's price from its statistical mean.
RSI	$100 - 100/\left(1 + (\sum_{i=0}^{n-1} Up_{t-i}/n)/(\sum_{i=0}^{n-1} Dw_{t-i}/n)\right)$	Relative strength index. It is a price following an oscillator that ranges from 0 to 100. Where Up_t and Dw_t mean upward and downward price change at time t, respectively.

Let's explain our improved algorithm in detail with an example as follow. Assure analyst advises the price of a stock (e.g. S_{600801}) will rise in a short period since time t_s. Hence this period ($Rperiod$) would be from t_s+1 (t_1) to t_s+45 (t_{45}). Total stock market index is named as S_{total}. Let $Rperiod_1,\cdots,Rperiod_m$ denote all recommendation periods preceding to $Rperiod$, jt_0 is the time when analyst give the $j-th$ recommendation. Firstly, the relative upgrade ratio in $t-th$ trading day of $Rperiod_j$ (RUR_{jt}) can be got as formula (1).

$$IUR_{jt}=IP_{jt}/IP_{jt_0}-1,\quad TUR_{jt}=TP_{jt}/TP_{jt_0}-1,\quad RUR_{jt}=IUR_{jt}-TUR_{jt} \tag{1}$$

Where IP_{jt} and TP_{jt} are the closing price of S_{600801} and S_{total} in $t-th$ trading day of $Rperiod_j$, respectively. IP_{jt_0} and TP_{jt_0} are the closing price at time jt_0. Hence IUR_{jt} and TUR_{jt} are the upgrade ratio of them. As depicted in Fig.1, relative upgrade ratio (RUR_{jt}) is increased and decreased at time t_8 and t_{11} respectively, since total stock market index drops and rise.

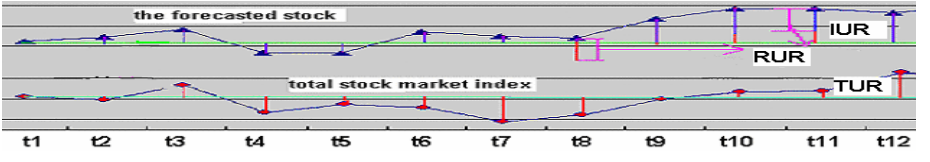


Fig. 1. Note that each value is the upgrading ratio of closing price in *i-th* day (IP_{jt}/TP_{jt}) to that in which analyst give the recommendation (IP_{jt_0}/TP_{jt_0}). Relative upgrade ratio RUR_{jt} enhances in *8-th* day since total stock market index drops, and contrary in *11-th* day.

Then $RACR_{t_k}$, the average RUR_t for S_{600801} in $Rperiod_1,\cdots,Rperiod_m$, should be evaluated according to equation (2). Note that for each t_k in $Rperiod$, all former recommended days before t_k should be employed.

$$RACR_{t_k}=\left(\sum_{j=1}^{m}\sum_{q=1}^{td_j}RUR_{jq}+\sum_{i=1}^{k}RUR_{t_i}\right)\Bigg/\left(\sum_{j=1}^{m}(td_j)+k\right) \tag{2}$$

Where td_j is the number of trading days in $Rperiod_j$. Finally we can evaluate the reliability of upgrade forecast at t_k by the follow two formulas.

$$ACC_{t_k}=\mathrm{sgn}(RACR_{t_k}-RUR_{t_k})=\begin{cases}1, & RACR_{t_k}-RUR_{t_k}>0\\ 0, & RACR_{t_k}-RUR_{t_k}\le 0\end{cases} \tag{3}$$

$$Con_{t_k} = ACC_{t_k} \wedge UFC_{t_k}, \qquad ACC_{t_k}, UFC_{t_k} \in \{0,1\} \tag{4}$$

Where sgn is a denote function, UFC_{t_k} is the upgrade forecast at t_k by training *C*-SVC, Con_{t_k} is the final forecast result through logical AND rule. If Con_{t_k} equate to 1, the upgrade forecast is reliable, otherwise being not reliable.

3 Research Data and Experimental Results

3.1 Data Set

In this study, we use the daily stock price index in Shanghai stock exchange market. The period of training data is from 01/01/2003 to 01/31/2005 and the period from 02/01/2005 to 09/30/2005 is for testing data. Based on the formulas in Table1, thirteen input attributes are first calculated. Each attributes is normalized and scaled into the range of [0, 1]. The changing trend of stock price in next trading day is categorized as "0" or "1". If the closing price in next day is 0.5% higher than the current, the class label of the current day is "1", else "0". In addition, more than 399 analyst recommendations about 218 stocks were employed [7]. Where about 221 recommendations belong to short term and the 178 belong to middle term.

3.2 Experimental Results

The popular LIBSVM is used to perform our experiment [4]. After scaling the training data, we train it with five-fold cross-validation and grid-search. In view of the pages restriction on the paper, we only depict the training result about S_{600801}. The best probability estimates about (C, γ, correct rate) is (0.94278, 3.86375, 63.843) and 95 of total 158 samples in testing data set are classified.

Table 2. Three stock upgrade are forecasted with *C*-SVC (UFC=1). By employing analyst recomemdations, only two are conformed (Con=1). Hence the uncorrect upgrade forecast in 9/21/2005 is excluded by using *C*-SVC and analyst recomemdations meanwhile.

Date	ACR	RUR	UFC	ACC	Con	Class Label
2005-8-22	0.031154	0.02642	0	1	0	1
……						
2005-9-21	0.044926	0.1114	1	0	0	0
……						
2005-9-26	0.045076	0.03441	0	1	0	0
2005-9-27	0.044741	0.01289	1	1	1	1
2005-9-28	0.044477	0.01911	1	1	1	1

As listed in Table2, for S_{600801}, there are three upgrade predictions (UFC=1) in testing data but only two are conformed (Con=1). So in this case, an error upgrade forecast of *C*-SVC is excluded by employing improved forecast model. In order to verify our improved model roundly, more stocks are tested and most get better results.

4 Conclusions

Due to random walk and non-stationarity, we attempt to analyze the forthcoming trend rather than predict regressive values. In order to make best use of market information, we put forward an improved model by employing analyst recommendations. We explain relative recommendation accuracy and logical AND rule correlative to total stock market index. Based on this forecasting model, we can distinguish if an upgrade forecast of *C*-SVC is reliable. Experiments are conducted to evaluate the performance of the model based on actual data during recent three years. The results indicate that recommendations really contribute to stock trend forecasting.

References

1. Haykin, S.: Neural Networks: A Comprehensive Foundation. 2nd edn, Englewood Cliffs, NJ: Prentice Hall, (1998)
2. Kim, K.J.: Financial Time Series Forecasting Using Support Vector Machines. Neurocomputing 55 (2003) 307-319
3. Fung, G.P.C., Yu, J.X., Lam, W.: Stock Prediction: Integrating Text Mining Approach using Real-Time News. In: Computational Intelligence for Financial Engineering, (2003) 395-402
4. Chang, C.C., Lin, C.J.: LIBSVM: A Library for Support Vector Machines. Technical Report, Department of Computer Science, National Taiwan University, (2001)
5. Hsu, C.W., Chang, C.C.: A Practical Guide to Support Vector Classification. Department of Computer Science and Information Engineering, National Taiwan University, (2003)
6. http://www.spaceandmotion.com/quantum-theory-albert-einstein-quotes.htm
7. http://www.skyone.com.cn/share/main.jsp?pcat=tytyzx&unicat=tyggtj&acat=tyyj
8. http://www.hzhao.com/bbs/read.php?tid=15619

Stock Index Prediction Based on the Analytical Center of Version Space

Fanzi Zeng and Yonghua Zhang

Institute of Computer and Communication, Hunan University,
410082 Changsha, China
zeng_f@126.com

Abstract. This paper presents a novel method for predicting the stock index based on the multiclass classification. The strategy firstly discretizes the stock index to different interval and assigns a class label to each interval, which yields a multiclass classification problem. After briefly reviewing multiclass classification algorithm, a multiclass classifier based on analytical center of version space is proposed and applied to stock index prediction. Experiments on shanghai stock index demonstrates that the strategy of stock index prediction proposed is validated and of practical value.

1 Introduction

Because of stock index series with highly noise, non-linearity and non-stationarity intrinsically, classical statistical methods [1][2] based on the assumption that series is stationary, remainders normal and independent, is always ineffective. Neural networks for their high non-linearity, self-adaptation, and function approximation power, has broadly been popular in prediction fields [3][4], but their intrinsic flaws such as local minima severely confine their practical application. In stock market, investors are mainly concerned about the direction and amplitude of stock index fluctuation, in addition, accurately predicting the stock index is very difficult for its highly non-linearity and non-stationarity. Therefore, a multiclass classification strategy is proposed to predicting the interval of stock index. The strategy firstly discretizes the stock index to different intervals, assigns a label to each interval so that stock index prediction is embedded into multi-classification framework.

Nowadays, the prevalent approach to multiclass classification is the "one versus all" (OvA) approach, which makes direct use of "standard" binary classifiers to train the output labels. Using the OvA approach, k classifiers must be constructed by repeating separating a single class from all the others for k-class problem, which leads to daunting computation and low efficiency of classification. Refs. [5][6] transform k class problem into m<k one by encode, which relieves inefficiency of OvA strategy to some extent. Ref. [7] proposes multi-class classifier based on SVM, which corresponds to a simple quadratic optimization and needn't repeating reconstructing procedure as OvA strategy does. However, because SVM corresponds to the center of the largest inscribed hypersphere in version space [8][9], thus when the version space, i.e. the space of hypotheses consistent with the training data, is elongated or asymmetric, SVMs are not very effective. Therefore, a multi-class classifier based on the

J. Wang et al. (Eds.): ISNN 2006, LNCS 3971, pp. 458 – 463, 2006.

analytical center of version space (MACM) is proposed to address the above problems and applied to stock index prediction.

2 Pre-processing of Stock Index Series

Given the stock index series $\{x_t\}_{t=1}^{N}$, calculate the relative difference series of stock index series $\{r_t\}_{t=1}^{N-1}$: $r_t = \frac{x_t - x_{t-1}}{x_{t-1}} \times 100$, then by moving window with width d transform the time structure of stock index series into spatial structure $\{X_i\}_{i=1}^{N-1-d}$, $X_i = \{r_t\}_{t=i}^{i+d-1}$. According the decision of investors, define the label corresponding to the X_i as

$$y_i = \begin{cases} 1 & \text{if } r_{i+d} \geq upper \\ 0 \text{ if } lower \leq r_{i+d} < upper \\ -1 & \text{if } r_{i+d} < lower \end{cases},$$

where *upper* and *lower* represent threshold determined by investors. If $y_i = 1$ means the chance to buy, $y_i = -1$ the chance to sold, $y_i = 0$ standby. The meaning of stock index pre-processing lies in that pre-processing suppresses the noise, mitigates the non-stationarity and caters for the need of investors. The above pre-processing yields the multi-classification problem with training input $\{X_i\}_{i=1}^{N-1-d}$, and the corresponding output $\{y_i\}_{i=1}^{N-1-d} \in \{-1,0,+1\}^{N-1-d}$. Next section formulates the multi-classifier based on analytical center.

3 Multiclass-Classifier Based on Analytical Center

To facilitate the formulation of multiclass-classification based on analytical center of version space(MACM), some definitions are introduced as follows.

Definition 1 (chunk). A vector $\boldsymbol{v} = (v_1, \cdots, v_{kd}) \in \Re^{kd} = \Re^d \times \cdots \times \Re^d$, is broken into k chunks $(\boldsymbol{v}_1, \cdots, \boldsymbol{v}_k)$ where the i-th chunk, $\mathbf{v}_i = (v_{(i\text{-}1)*d+1}, \cdots, v_{i*d})$.

Definition 2 (Expansion). Let Vec(x,i) be a vector $x \in \Re^d$ embedded in kd dimensions, by writing the coordinates of x in the i-th chunk of a vector in $\Re^{kd}$. Denote by $\mathbf{0}^{\ell}$ the zero vector of length ℓ. Then Vec(x,i) can be written formally as the concatenation of three vectors, Vec(x,i)= $(\mathbf{0}^{(i\text{-}1)*d}, x, \mathbf{0}^{(k-i)*d}) \in \Re^{kd}$. Finally, Vec($x,i,j$) = Vec($x,i$)- Vec($x,j$), is the embedding of x in the i-th chunk and $-x$ in the j-th chunk of a vector in $\Re^{kd}$.

Definition 3 (Piecewise linear separability). The point sets $A^i, i = 1, \cdots, k,$ represented by the metrics $A^i \in \Re^{d \times m_i}, i = 1, \cdots, k$ (k represents the class number), are piecewise linear separable if there exists $w^i \in \Re^d$ and $\gamma^i \in \Re, i = 1, \cdots, k,$ d represents the dimension of point, such that

$$\begin{aligned} &A_\ell^i w^i - \gamma^i > A_\ell^i w^j - \gamma^j, \\ &i, j = 1, \cdots, k, \ i \neq j, \ \ell = 1, \cdots, m_i \end{aligned} . \tag{1}$$

Definition 4 (piecewise linear classifier). Let $\tilde{w} = (\tilde{w}^1, \cdots, \tilde{w}^k)$ be a set of k vectors, where $(\tilde{w}^1, \cdots, \tilde{w}^k) \in \Re^{k*(d+1)} = \Re^{d+1} \times \cdots \times \Re^{d+1}$. Given a new point $x \in \Re^{d+1}$, the piecewise linear classifier is a function $f : \Re^{d+1} \to \{1, \cdots, k\}$

$$f(x) = \arg\max_{i=1,\cdots,k} \tilde{w}^i x' . \tag{2}$$

where argmax returns the class label corresponding to the maximum value.

To simplify the notation for the formulation of multiclass classifier based on analytical center of version space(MACM), we consider an augmented weight space as follows.

Let $\tilde{A}_\ell^i = \begin{bmatrix} A_\ell^i \\ 1 \end{bmatrix} \in \Re^{d+1}$, $\tilde{w}^i = \begin{bmatrix} w^i \\ \gamma \end{bmatrix} \in \Re^{d+1}$, $i = 1, \cdots, k, \quad \ell = 1, \cdots, m_i$, then the inequality (1) can be rewrote as

$$\begin{aligned} &\tilde{A}_\ell^i \tilde{w}^i - \tilde{A}_\ell^i \tilde{w}^j > 0, \\ &i, j = 1, \cdots, k, i \neq j, \ \ell = 1, \cdots, m_i \end{aligned} . \tag{3}$$

Let $\tilde{w} = (\tilde{w}^1, \cdots, \tilde{w}^k) \in \Re^{k*(d+1)} = \Re^{d+1} \times \cdots \times \Re^{d+1}$. According the definition 2, embedding $\tilde{A}_\ell^i$ into $\Re^{k*(d+1)}$ space, the inequality (3) becomes as follows:

$$\begin{aligned} &(\mathrm{Vec}_\ell(\tilde{A}_\ell^i, i) - \mathrm{Vec}_\ell(\tilde{A}_\ell^i, j))\tilde{w} > 0, \\ &i, j = 1, \cdots, k, i \neq j, \ \ell = 1, \cdots, m_i \end{aligned} . \tag{4}$$

Considering that

$$\begin{aligned} &\mathrm{Vec}_\ell(\tilde{A}_\ell^i, i, j) = \mathrm{Vec}_\ell(\tilde{A}_\ell^i, i) - \mathrm{Vec}_\ell(\tilde{A}_\ell^i, j), \\ &i, j = 1, \cdots, k, \quad i \neq j, \ \ell = 1, \cdots, m_i \end{aligned} . \tag{5}$$

Thus, the inequality (4) can be rewrote as follows:

$$\begin{aligned} &\mathrm{Vec}_\ell(\tilde{A}_\ell^i, i, j)\tilde{w} > 0, \ i, j = 1, \cdots, k, \\ &i \neq j, \ \ell = 1, \cdots, m_i \end{aligned} . \tag{6}$$

The inequality (6) represents the version space of $\tilde{w}$. Define the slack variable

$$\begin{aligned} &S_{i,j,\ell} = \mathrm{Vec}_\ell(\tilde{A}_\ell^i, i, j)\tilde{w}, \\ &i, j = 1, \cdots, k, i \neq j, \ \ell = 1, \cdots, m_i \end{aligned} . \tag{7}$$

We have the following minimization problem, which solver corresponds to the analytical center of $\tilde{w}$,

$$\begin{aligned} &\min \quad \Phi(\tilde{w}) = -\sum_{\ell=1}^{m_i} \sum_{i \neq j, \, i, j=1}^{k} \ln S_{i,j,\ell} . \\ &s.t. \quad h(\tilde{w}) = \tfrac{1}{2} \tilde{w} \tilde{w}' - 1 = 0 \end{aligned} \tag{8}$$

We introduce some notations as follows,

$M = k*(k-1)*\sum_{i=1}^{k} m_i$, $B = \{\mathrm{Vec}_\ell(\tilde{A}_\ell^i, i, j) | i, j = 1, \cdots, k,\ i \neq j,\ \ell = 1, \cdots, m_i\} \in \Re^{M \times k*(d+1)}$ Let B_i represents the *i*-th row vector of *B*. Then the minimization problem (8) becomes,

$$\begin{aligned} &\min \quad \Phi(\tilde{w}) = -\sum_{i=1}^{M} \ln S_i \quad \text{where } S_i = B_i \tilde{w}. \\ &s.t. \quad h(\tilde{w}) = \tfrac{1}{2} \tilde{w} \tilde{w}' - 1 = 0 \end{aligned} \tag{9}$$

In the following we solve the optimization problem (9). Let us start analysis by introducing the Lagrangian of (9). With λ being the Lagrangian multiplier we have:

$$L(\tilde{w}, \lambda) = \Phi(\tilde{w}) + \lambda \cdot h(\tilde{w}). \tag{10}$$

The KKT optimality conditions require that at the optimality the gradient of the Lagrangian vanishes and the primary feasibility corresponding the hyper-sphere constraint is met. The following equations represent the optimality conditions accounting for the primary and dual feasibility:

$$\begin{aligned} &\nabla_{\tilde{w}} L(\tilde{w}) = \nabla_{\tilde{w}} \Phi(\tilde{w}) + \lambda \cdot \nabla_{\tilde{w}} h(\tilde{w}) \\ &h(\tilde{w}) = \frac{1}{2} \tilde{w} \tilde{w}' - 1 = 0 \end{aligned}. \tag{11}$$

Next, we define a new variable $Z(\tilde{w}, \lambda)$ and let us rewrite the optimality conditions in matrix form. Thus

$$Z(\tilde{w}, \lambda) = \begin{bmatrix} \nabla_{\tilde{w}} L(\tilde{w}) \\ h(\tilde{w}) \end{bmatrix} = \begin{bmatrix} \nabla_{\tilde{w}} \Phi(\tilde{w}) + \lambda \cdot \nabla_{\tilde{w}} h(\tilde{w}) \\ h(\tilde{w}) = \frac{1}{2} \tilde{w} \tilde{w}' - 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{12}$$

In a more compact form, the optimality conditions requires $Z(\tilde{w}, \lambda)$=0. Employing the Newton-Raphson method to calculate $\tilde{w}^*$ and λ^*, then the first-order approximation linear system

$$Z(\tilde{w}_k, \lambda_k) + \nabla Z(\tilde{w}_k, \lambda_k) \begin{bmatrix} \tilde{w} - \tilde{w}_k \\ \lambda - \lambda_k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{13}$$

must be solved, in order to calculate the next iteration $(\tilde{w}, \lambda) = (\tilde{w}_{k+1}, \lambda_{k+1})$. Equation (13) allows us to iteratively approach $(\tilde{w}^*, \lambda^*)$. Here, ∇Z denotes the Jacobian of Z and $(\tilde{w}_k, \lambda_k)$ represents the current iteration. We find the Jacobian of Z

$$\nabla Z(\tilde{w}_k, \lambda_k) = \begin{bmatrix} \nabla^2 L(\tilde{w}_k, \lambda_k) & \nabla h(\tilde{w}_k, \lambda_k) \\ \nabla h(\tilde{w}_k, \lambda_k)' & 0 \end{bmatrix}. \tag{14}$$

We introduce a metrix $\mathbf{S} \in \Re^{M \times M}$ that carries in its diagonal the slack variable $S_i = B_i \tilde{w}_k$, where $\tilde{w}_k$ represents the current iterate.

Let $\mathbf{e} = (1, \cdots, 1)$ denotes a vector of ones in an appropriate dimension, here, $\mathbf{e} \in \Re^M$. Then the gradient of $\Phi(\tilde{w})$ becomes $\nabla\Phi(\tilde{w}) = -\frac{B_1}{S_1} - \frac{B_2}{S_2} - \cdots - \frac{B_M}{S_M} = -B\mathbf{S}^{-1}\mathbf{e}$ and its Hessian can

be computed similarly, resulting in $\nabla^2\Phi(\tilde{w}) = \frac{B_1B_1'}{S_1^2} + \frac{B_2B_2'}{S_2^2} + \cdots + \frac{B_MB_M'}{S_M^2} = B\mathbf{S}^{-2}B'$. Considering the gradient and Hessian of *h*, we find for the gradient $\nabla h(\tilde{w}) = \tilde{w}$ and for the Hessian $\nabla^2 h(\tilde{w}) = \mathbf{I}$, with $\mathbf{I} \in \Re^{M\times M}$ representing the unit metrix. Now it is easy to compute the Z and ∇Z.

After calculating the optimal weight $\tilde{w}^*$, the multiclass classifier $f : \Re^{d+1} \to \{1,\cdots,k\}$ is computed in the following way,

$$f(x) = \arg\max_{i=1,\cdots,k} \tilde{w}_i^* x' \cdot \tag{15}$$

where argmax returns the class label corresponding to the maximum value.

We have discussed the piecewise-separable case, if the piecewise inseparable case is encountered, the kernel strategy is used to map the training datasets into feature space, in which training datasets is piecewise separable.

4 Experimental Results

Experiments are conducted with Shanghai stock index (from 2000-9-29 to 2002-10-30) and the experimental results with the multiclass classifier proposed in this paper(MACM) are compared with those with the most popular method: M-SVM[7]. Shanghai stock index and corresponding relative difference series is shown in Fig.1. According section 2, we discretize the stock index to different interval and assign class label to each interval, which produces the multiclass classification sample. In the experiments, the ratio of training sample and test sample is 9:1, MACM with kernel substitution is used and kernel is radial basis kernel. Given test sample $\{X_i, y_i\}_{i=1}^m$, in order to evaluate the performance of MACM and effectiveness of the proposed strategy for stock index prediction, two measures is defined as follows:

1. $err = \sum_{i=1}^{m} a_i \Big/ m$, where $a_i = 1$ *if* $h(X_i) \neq y_i$, *else* $a_i = 0$,
2. $err_{-1,1} = \sum_{i=1}^{m} b_i \Big/ m$, where $b_i = 1$ *if* $h(X_i) = 1$, and $y_i = -1$, *else* $b_i = 0$.

$err_{-1,1}$ represents the ratio that class –1 is misclassified as +1, which measures the risk caused by when investors should sell but buy the stock. Sometime this kind of risk is

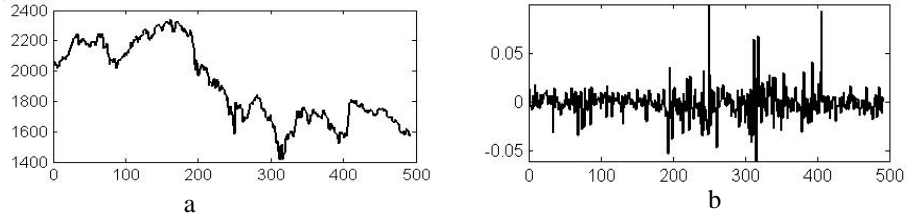


Fig. 1. (a) Stock index series. (b) Relative difference series of stock index series.

catastrophic, so controlling this kind of risk is paramount to investors. The experimental results are show in Table 1.

Table 1 witnesses that MACM outperforms M-SVM in this scenario, and misclassification error for MACM and M-SVM reduces at the beginning, and then increases as window width d increases. This is because short window contain the insufficient information, but long window probably introduces noise and irrelevant information.

Table 1. misclassification error(100%) in test set

	d	5	10	15	20	25	35
M-ACM	*err*	50.15	46.73	32.96	33.52	37.26	40.47
	$err_{-1,1}$	7.51	6.80	6.44	6.16	6.37	6.92
M-SVM	*err*	51.40	48.36	37.28	36.83	38.57	46.39
	$err_{-1,1}$	8.54	6.72	6.54	6.31	6.48	6.81

5 Conclusions

This paper presents a novel strategy for predicting the stock index based on the multiclass classification. The strategy firstly discretizes the stock index to different interval and assigns a class label to each interval, which yields a multiclass classification problem. After briefly reviewing multiclass classification algorithm, a multiclass classifier based on analytical center of version space is proposed and applied to stock index prediction. Experiments on shanghai stock index demonstrates that the strategy of stock index prediction proposed is validated and of practical value.

References

1. Brockwell, P.J.: Time Series: Theory and Methods. Springer-Verlag (1987)
2. An, H., Non-linear Time Series Analysis (in Chinese). Shanghai Science Press (1998)
3. Zhang, B.-L.: Multiresolution Forecasting for Future Trading Using Wavelet Decompositions. IEEE Trans. on Neural Networks 12(4) (2001) 765-775
4. Mkhalaf, A.A.: Time Series Prediction Using a Hybrid Model of Neural Network and FIR Filter. IEEE (1998) 1975-1980
5. Dekel, O., Singer Y.: Multiclass Learning by Probabilistic Embeddings. Advances in Neural Information Processing Systems, Vol. 15 (2002)
6. Allwein, E., Schapire, R., Singer, Y.: Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers. Proc. 17th International Conf. on Machine Learning (2000)
7. Bredensteiner, E.J., Bennett, K.P.: Multicategory Classification by Support Vector Machines. http://mpa.itc.it/biblio/papers/ bredensteiner99multicategory.pdf.
8. Smola, A.: Advances in Large Margin Classifiers. The MIT Press, Cambridge (2000)
9. Trafalis, T.B., Malyscheff, A.M.: An Analytic Center Machine. Machine Learning, 46(1/3) (2002) 203–223

Comparison of Forecasting Performance of AR, STAR and ANN Models on the Chinese Stock Market Index

Qi-an Chen[1] and Chuan-Dong Li[2]

[1] College of Economics and Business Administration,
Chongqing University 400030, China
chenqian@cqu.edu.cn
[2] College of Computer Science, Chongqing University, Chongqing 400030, China
licd@cqu.edu.cn

Abstract. This paper investigates whether it is possible to exploit the nonlinear behavior of daily returns to improve forecasting on Chinese Shanghai stock market index over short and long horizons. We compare out-of-sample forecasts of daily returns for the Chinese Shanghai Stock Market Index, generated by five competing models, namely a linear AR model, the LSTAR and ESTAR smooth transition autoregressive models and two ANN models: MLP and JCN. The research results show that the nonlinear ANN models may be an appropriate way to improve forecasts. The return on the Chinese Shanghai Stock Market Index could be predicted more accurately by using ANN models, and the neural network technique could be said to represent a slight improvement in prediction of the stock index with respect to AR model and STAR models.

1 Introduction

Over the past several decades, many important changes have taken place in the environment of financial markets. On the one hand, the development of powerful communication and trading facilities has enlarged the scope of selection for investors [1]. On the other hand, the fact that a large number of emerging stock markets in many developing countries have been setting up also changes the features of traditional stock markets. Traditional capital market theory has changed and methods of financial analysis have improved [2]. How to predict stock market index better has become an important financial subject attracting many researchers' attentions for many years.

Forecasting stock return involves an important assumption that past publicly available information has some predictive relationship to future stock return [3]. However, most of the present studies attempting to capture the relationship between the available information and the stock returns rely on the linear assumption [4]. Practically, there is no evidence that the relationship between the stock returns and the financial or economic variables is perfectly linear. This evidence has provoked theoretical and practical interest in nonlinear financial time-series models. A large number of nonlinear models have been built to describe and forecast financial and economic time series, e.g., bilinear model, ARCH and its extensions, STAR, ANNs, wavelets, etc..

The Chinese stock market is a developing market and its efficiency cannot be comparative to the developed stock markets in USA and Europe obviously. Consequently, predictability of Chinese stock market index should be greater than that of

J. Wang et al. (Eds.): ISNN 2006, LNCS 3971, pp. 464 – 470, 2006.

USA and Europe, and the forecasting performance of various nonlinear financial models should be better theoretically. However, empirical research on predictability of Chinese stock market is inadequate so far. The objective of this paper is to investigate whether it is possible to exploit the existence of nonlinearities to improve forecasting on Chinese stock market index over short and long horizons.

To study the predictability of Chinese stock index, we analyze the out-of-sample forecasts from STAR and ANN models using the research methods provided in the related literature [5] for reference. We examine whether out-of-sample forecasts generated by nonlinear models are more accurate and preferable to those generated by linear AR models for stock index, employing statistical criteria such as goodness of forecast measures, proportion of times the signs of index are correctly forecasted and directional accuracy test [6-7]. The research work of this paper would become basis of studying more deeply predictability and efficiency of Chinese stock market and have very important theoretical and practical implication.

2 Forecasting Model

Because we aim to analyze the dynamic characteristic of returns for Chinese stock index, only lagged returns are considered as explanatory variables in the research of this paper. In order to realize this goal, it is necessary to explain the linear AR model and the nonlinear STAR and ANN model used in this paper respectively.

2.1 AR Model

A p-order autoregressive process (AR(p)) is expressed as follows[8]:

$$r_t = \sum_{i=1}^{p} \phi_{t-i} r_{t-i} + \varepsilon_t \tag{1}$$

Where r_{t-i} (i=0,1,2,$\cdots p$) are the returns of stock market index, ϕ_{t-i} (i=1,2,$\cdots p$)are the unknown parameters, ε_t is an i.i.d. stochastic residual error.

2.2 STAR Models

A simple first-order STAR model with two regimes could be defined as follows[9]:

$$r_t = \phi_{10} + \sum_{i=1}^{p} \phi_{1i} r_{t-i} + \left[\phi_{20} + \sum_{i=1}^{p} \phi_{2i} r_{t-i} \right] F_{t,d}(\gamma, c) + \varepsilon_t \tag{2}$$

Where r_{t-i} (i=0,1,2,$\cdots p$) are the returns of stock market index, ϕ_{ij} (i=1,2; j=0,1,2,$\cdots p$) are the unknown parameters that correspond to each of the two regimes, and ε_t is a i.i.d. stochastic residual error. $F_{t,d}(\gamma,c)$ is the transition function, assumed to be twice differentiable and bounded between 0 and 1, and γ is the transition rate or smoothness parameter, c is the threshold value which represents the change from one

regime to another, d is the number of lags of transition variable which is usually defined as r_t.

According to the different transition function $F_{t,d}(\gamma,c)$, we could obtain different STAR model. In most of the research literatures, the following two kinds of transition functions are widely used:

(i) $F_{t,d}(\gamma,c)=\dfrac{1}{1+e^{-\gamma(r_{t-d}-c)}}$, $\gamma>0$, in which case the STAR model is called Logistic STAR or LSTAR(p, d).

(ii) $F_{t,d}(\gamma,c)=1-e^{-\gamma(r_{t-d}-c)^2}$, $\gamma>0$, in which case the STAR model is called Exponential STAR or ESTAR(p, d).

2.3 ANN Models

Artificial neural networks are one of the technologies that have caused the most recent excitement in modern financial environment. They provide an interesting technique that theoretically can approximate any nonlinear continuous function on a compact domain to any designed degree of accuracy [10]. The novelty of neural networks lies in their ability to model nonlinear processes without a priori assumptions about the nature of the generating process. This is useful in security investment and other financial areas where much is assumed and little is known about the nature of the processes determining asset prices. Consequently, ANN models are considered to be a universal predictor in a wide variety of nonlinear patterns. The following figure (i.e., fig.1) describes a kind of classical neural networks.

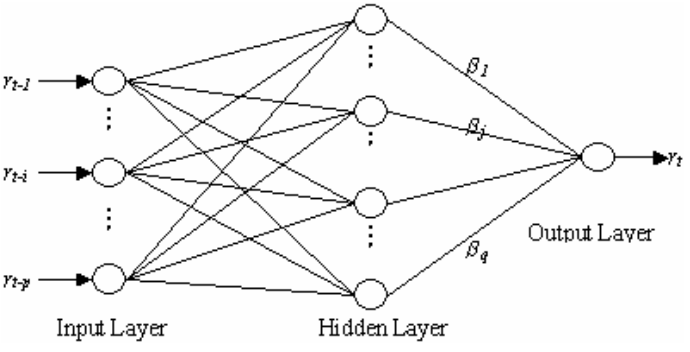


Fig. 1. A Three-layer feed-forward neural networks used for prediction of stock returns

According to Fig.1, we could obtain various different neural network models used for forecasting the stock returns. In this paper, we only make use of the Multi-layer Perception Model (simplified by MLP(p, q)) and Jump Connection Nets Model (simplified by JCN(p,q)) which are most widely used and simplest to forecast Chinese stock market return.

(i) MLP(p, q) model
MLP(p, q) model is defined as the following form[11]:

$$r_t = \beta_0 + \sum_{j=1}^{q} \beta_j g(\sum_{i=1}^{p} \phi_{ij} r_{t-i} + \phi_{0j}) + \varepsilon_t \quad (3)$$

where r_t is the stock market return in t or system output, and ε_t is a i.i.d. stochastic residual error. p is the number of the inputs or lagged stock market returns taken as explanatory variables β_j (j=1,2…,q) represent the weights from the hidden layer to the output unit and ϕ_{ij} (i=1,2, …,p; j=1,2…,q) represent the weights from the input layer to the hidden layer, and they bring together all the network weights. The function $g(\bullet)$ determines the connections between nodes of the hidden layer and is used as the hidden unit activation function to enhance the nonlinearity of the model. In this paper, for the sake of convenience, the functional form that we take is that when x>1, $g(x)$=1; when 0≤x≤1, $g(x)$=x; and when x<0, $g(x)$=0.

(ii) JCN(p, q) model
JCN(p, q) model is a network with direct connections between the inputs and output. It is defined as the following form [11]:

$$r_t = \beta_0 + \sum_{i=1}^{p} \alpha_i r_{t-i} + \sum_{j=1}^{q} \beta_j g(\sum_{i=1}^{p} \phi_{ij} r_{t-i} + \phi_{0j}) + \varepsilon_t \quad (4)$$

Where α_i (i=1,2, …,p) are direct input-output weights, and other parameters and variables and function $g(\bullet)$ in the Eq.(4) are similar to the Eq.(3). We could see from Eq.(3) and Eq.(4) that JCN(p, q) model is really mixture of the MLP(p, q) model and p-order AR(p) model.

3 Data Selection

In order to study forecasting performance of the above-mentioned several models on Chinese stock market index, we choose from the CSMAR-Trading System the daily closing prices of Shanghai Stock Market Index from 8 October 1996 to 31 December 2004, with a total of 1989 observations, to serve as our research sample data. The reason of doing so is that Up-Stop Board system and Down-Stop Board system are carried out during this period, the stock policy is not greatly changed, and the policy factor of Chinese Stock Market is not too much. The observation series is transformed into logarithms to compute continuous returns according to the following expression: r_t=$log(P_t/P_{t-1})$, t=1,2,⋯,1988, where P_t (t=0,1,2, …,1988) are the daily closing prices of Shanghai Stock Market Index, and log is the nature logarithm.

4 Empirical Results and Analyses on the Empirical Results

4.1 Statistical Measures and Tests

In order to assess the predictability of various models on the Chinese Stock Market Index, we compare the out-of-sample forecasts using two different approaches,

because it is generally impossible to specify a forecast evaluation criterion that is universally acceptable. Firstly, we examine the forecast accuracy of all the above-mentioned models by calculating the MAE, MAPE, RMSE, Theil's U-statistic and the proportion of times the sign of returns are correctly forecasted. Secondly, we employ the following test of hypotheses: the Pesaran and Timmermann (DA,1992) test [6] is used to examine the directional prediction accuracy of changes.

4.2 Empirical Results

Before giving the formal empirical results, we briefly describe the main characteristics of in-sample aforementioned model estimates:

(i) By calculating ACF of r_t, we choose p=5 in AR(p) model. The parameters ϕ_{t-I} (i=1,2,⋯, 5) in Eq.(1) are estimated by OLS method.

(ii) Making use of the modeling procedure described by Granger and Teräsvirta(1993)[12], we choose p=3 and d= 6 on the basis of SBIC in LSTAR(p, d) model and ESTAR(p, d) model. All parameters in LSTAR(p, d) and ESTAR(p, d) is estimated by quasi-maximum likelihood estimations (QMLE).

(iii) We consider that ANN models include one hidden layer and many hidden units, moreover, p, q and $g(\bullet)$ are not chosen a priori. The p-order lagged returns are calculated by sequential validation, and so we estimate ANN models with different values of p and q. The rank of the terms employed is p, q=1,2,⋯, 8. All lagged stock index returns are scaled assuming a uniform distribution within the interval [-1,1]. All ANN models are trained over a given period (i.e., training sample) using 1800 cycles and cross-validation. The resulting set is used to estimate the neural network weights. To improve the in-sample fitting performance of the ANN models, the estimated set of weights is used as a set of initial values for training. We use cross-validation strategy in training to avoid over-fitting. The decrease in the error rate in the training and "test" phases is then tested. The output is compared to the sample of original values of the output by comparing the RMSEs, which is found to decline as training progressed. When RMSE reaches a minimum and then starts increasing, this indicates that over-fitting may occur. On the basis of the estimated weights from nth training over the training period, out-of-sample forecasts are generated for subsequent "test" periods. In terms of the minimum RMSE in the out-of-sample phase, the best adjusted MLP model holds p=3 and q=5, and JCN model holds p=2 and q=3 in this paper.

Once each model has been estimated, we could construct its out-of-sample forecast. The following table gives the empirical results.

4.3 Analyses on the Empirical Results

Table1 gives the results of the goodness of forecast on Chinese Shanghai Stock Market Index for all the aforementioned models. The main conclusions are as follows:

(i) The results for forecast horizons and forecasting methods are heterogeneous in terms of the MAE, MAPE and RMSE measures. However, The ANN forecast is an improvement on the linear AR and nonlinear STAR models, because ANNs have lower forecast errors.

Table 1. Goodness of forecast for all the above-mentioned models on Chinese Shanghai Stock Market Index

Models	H=100 MAE	MAPE	RMSE	Theil-U	Signs	DA
AR	0.9925	103.3323	1.2540	0.8813	0.5032	0.9423
ESTAR	0.9836	102.8769	1.3067	0.8345	0.5245	0.8378
LSTAR	0.9764	103.6715	1.2133	0.8624	0.5512*	0.8905
MLP	0.9327*	102.2534	1.2056*	0.8126*	0.5482	0.8532
JCN	0.9433	102.1192*	1.2083	0.8267	0.5512*	0.8817
Models	H=200 MAE	MAPE	RMSE	Theil-U	Signs	DA
AR	0.9522	115.3602	1.3832	0.8717	0.5731	0.9520
ESTAR	0.9473	110.7100	1.3256	0.8645	0.5802	0.8907
LSTAR	0.9421	109.3837	1.3378	0.8913	0.5976	0.9053
MLP	0.9055*	108.5410*	1.3254	0.8575	0.6430*	0.8847
JCN	0.9283	109.1926	1.2998*	0.8430*	0.6359	0.8761
Models	H=300 MAE	MAPE	RMSE	Theil-U	Signs	DA
AR	0.9982	127.9156	1.4341	0.8991	0.5336	0.7843
ESTAR	0.9763	126.5632.	1.4035	0.8932	0.5429	0.7228
LSTAR	0.9800	123.5467*	1.5323	0.8894	0.5791	0.7564
MLP	0.9532	130.5563	1.3978	0.8787*	0.5564	0.6933
JCN	0.9445*	128.4392	1.3496*	0.8810	0.5840*	0.7396

(i) * indicates the best result between models in the same column.
(ii) H indicates the out-of-sample forecast horizons. In this paper, we take the value of H to be 100, 200 and 300-trading days.

(ii) Taking into account the number of times where ANNs are better than the linear AR and nonlinear STAR models, we think that the beat forecast horizon for our ANN technique is 200-forecast periods.

(iii) It is clear that ANN models surpass the other linear AR and nonlinear STAR models by using Theil's U-statistic at any horizon and forecasting method. Thus, the Chinese Shanghai Stock Market Index could be predicted by the nonlinear ANN models more Accurately.

From the analysis mentioned above, we can see that the ANN models give best results, with a superior out-of-sample performance to that of the other models.

5 Conclusions

We compared out-of-sample forecasts of daily returns for the Chinese Shanghai Stock Market Index, generated by five competing models. This comparison was carried out on the basis of various statistic criteria, e.g., MAE, MAPE, RMSE and Theil-U etc.

The research results showed that the nonlinear ANN models may be an appropriate way to improve forecasts. In the light of the results obtained, the statistical criteria suggest that the out-of-sample ANN forecasts are more accuracy, provide a better fit and provide improved predictions than do the AR and STAR models. Consequently, we conclude that the return on the Chinese Shanghai Stock Market Index could be predicted by using ANN models, and the neural network technique could be said to represent a slight improvement in prediction of the stock index with respect to AR model and STAR models.

Acknowledgements

This work is supported by the NNSF of China (Grant No.60574024).

References

1. Elton, E.J., Gruber, M.J.: Modern Portfolio Theory and Investment Analysis. 4th edn. John Wiley &Sons New York (1991)
2. Poddig, T., Rehkugler, H.: A World of Integrated Financial Markets Using Artificial Neural Networks. Neurocomputing 10 (1996) 251-273
3. Fama, E.F.: Efficient Capital Markets:II. Journal of Finance 46 (1991) 1575-1617
4. Keim, D., Stambaugh, R.: Predicting Returns in the Stock and Bond Markets. Journal of Financial Economics 17 (1986) 357-190
5. Pérez-Rodríguez, J.V., Torra, S., Andrada-Félix, J.: STAR and ANN Models: Forecasting Performance on the Spanish "Ibex-35" Stock Index. Journal of Empirical Finance 12 (2005) 490-509
6. Pesaran, M.H., Timmermann, P.: A Simple Nonparametric Test of Predictive Performance. Journal of Business and Economic Statistics 10 (1992) 461-465
7. Diebold, F., Mariano, R.: Comparing Predictive Accuracy. Journal of Business and Economic Statistics 13 (1995) 253-263
8. Mills, T.C.: The Econometric Modeling of Financial Time Series. 2nd edn. Cambridge University Press Cambridge (1999)
9. Teräsvirta, T.: Specification, Estimation and Evaluation of Smooth Transition Autoregressive Models. Journal of the American Statistical Association 89 (1994) 208-218
10. Cybenko, G.: Approximation by Superpositions of A Sigmoidal Function. Math. Control Signals Systems 2 (1989) 303-314
11. Kuan, C., White, H.: Artificial Neural Networks: An Econometric Perspective. Econometric Reviews 13 (1994) 1-91
12. Granger, C.W.J., Teräsvirta, T.: Modeling Nonlinear Economic Relationships. Oxford University Press Oxford (1993)

Index Prediction of KOSPI 200 Based on Data Models and Knowledge Rules for Qualitative and Quantitative Approach

Hyeon Bae[1], Sungshin Kim[1], and Joing-Il Bae[2]

[1] School of Electrical and Computer Engineering, Pusan National University,
30 Jangjeon-dong, Geumjeong-gu, 609-735 Busan, Korea
{sskim, baehyeon}@pusan.ac.kr
http://icsl.ee.pusan.ac.kr
[2] Division of Electrical and Control Instrumentation Eng., Pukyong National University,
100 Yongdang-dong, Nam-gu, 608-739 Busan, Korea
jibae@pknu.ac.kr

Abstract. In this paper, the neural network, the dynamic polynomial neural network (DPNN) and the fuzzy logic employed for the prediction of the KOSPI 200. The prediction results are compared by the root mean squared error (RMSE) and scatter plot, respectively. The results show that the performance of the fuzzy system is little bit worse than that of the DPNN but better than that of the neural network. We can develop the desired fuzzy system by optimization methods.

1 Introduction

The prediction of the stock price index is to predict the price trend of the future for preparing the dynamical changes of the markets. Practically the economist or statistics are using the charts or history data to predict the stock price index [1]. In general, the prediction of the stock price is separated into a technical analysis and fundamental analysis that is taken the internal values of the stock markets. But in the real market, it is very difficult and complicated to predict because there are the complexity and uncertainty in the market.

The artificial techniques are applied for the prediction system of the stock price and introduced as the commercial application for the prediction [2]. The neural networks are broadly employed for the prediction system that uses just history data. But the performance can be worse with respect to the market conditions. On the other hand, the fuzzy logic can use the human's knowledge, so if we can have good market information, the fuzzy model shows the better performance under the changeable market conditions [3], [4]. Also the neural network model and the dynamic polynomial neural network (DPNN) model are introduced in this paper [5], [6] [7]. The inputs of the fuzzy model are extracted based upon the moving average that is one of typical prediction approaches of the stock price index. In the experimental results, the prediction performance is presented that is produced by the neural network model, the DPNN model, and the fuzzy model using the KOSPI 200.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 471 – 476, 2006.

2 Prediction Models

2.1 Knowledge Information

2.1.1 KOSPI 200 Index

KOSPI 200 is to deal with two stock price markets. One is the futures market started from Jun 3 1996; the other is the options market opened from July 7 1997 KOSPI 200 is a capitalization-weighted index composed of 200 stocks from a broad range of industries. The component stocks are weighted according to the total market value of their outstanding shares. The impact of a component's price change is proportional to the issue's total market value, which is the share price time the number of shares outstanding. The KOSPI 200 is calculated by the following formula:

$$\text{KOSPI 200} = \frac{\text{Current aggregated market value of component stocks}}{\text{Base aggregated market value of component stocks}} \times 100 \quad (1)$$

2.1.2 Trifle Cross Method

Trifle cross method is one of the popular analysis method of the index that combines the moving average of the 4 days-9 days-18 days generally. In this paper, the moving average of 5 days-10 days-20 days is applied for the trifle cross method. From the cross pattern of the moving averages we can decide the trend of the stock price as rising, steadiness, and falling. The general knowledge is as follows.

- A: the moving average of 5 days is going through the moving average of 10 days and 20 days ⇒golden cross (rising trend)
- B: the moving average of 5days is steady in the moving average of 120 days, 10 days is rising, and 20 days is edge along ⇒ steadiness
- C: the moving average of 5 days is under 10 days and 10 days is under 20 days ⇒ dead cross (falling trend).

2.2 Extraction of Knowledge Information

2.2.1 Knowledge Extraction for Fuzzy Rules

The inputs of the fuzzy system is defined by the general knowledge of the KOSPI 200 that is the pattern of the moving average such as the trifle cross method. This knowledge is called knowledge information. In the fuzzy system, the fuzzy rules have to be extracted by the expert's knowledge or data because the fuzzy logic is one of rule-base model. But in the neural network models, inputs are just data variables. It is easier model to operate than the fuzzy model.

As mentioned before, the outputs of the trifle cross method is employed for the inputs of the fuzzy system. Table 1 indicates the way to extract inputs from the moving average lines. As shown in the left table of Table 1, the trend of tomorrow is determined by the moving average of today. This basic concept of the stock price index variation can be expressed by the following flow.

$$\text{Rising1} \Rightarrow \text{Rising2} \Rightarrow \text{Rising} \Rightarrow \text{Falling1} \Rightarrow \text{Falling2} \Rightarrow \text{Falling} \quad (2)$$

All change is moving with 6 steps. The right table of Table 1 shows the calculation technique that is based upon the left table of Table 1. If we want to predict tomorrow's prices index, the first input of the fuzzy system is the difference moving average between the 5days and 10days before from today (*T*). The other input is the difference moving average between the 5days and 10days before from yesterday (*T*-1). From this procedure, the fuzzy rules are extracted based on the experienced knowledge kind of the trifle cross method.

Table 1. Basic knowledge for generating rules and quantization of each defined case

Triple Cross Model	
Golden Cross	Dead Cross
20days 5days 10days Rise 1	10days 5days 20days Fall 1
5days 20days 10days Rise 2	10days 20days 5days Fall 2
5days 10days 20days Rise	20days 10days 5days Fall

Fuzzy Variables	
5days – 10days (+) Rising 1	5days – 10days (–) Falling 1
5days – 10days (+) Rising 2	5days – 10days (–) Falling 2
5days – 20days (+) Rising	5days – 20days (–) Falling

2.2.2 Fuzzy Rule Generation for the Prediction

Through preprocessing two fuzzy inputs are generated as *T* and *T*-1. And the output of the fuzzy model is the real stock price index of tomorrow. In the fuzzy model

Table 2. Fuzzy look-up table and basic knowledge for extracting rules

D-1 \ *D*	Rise 1	Rise 2	Rise	Fall 1	Fall 2	Fall
Rise 1		PB	PB			NS
Rise 2	NS		PB	ZE		
Rise		NS	PB	NS	NB	
Fall 1			PS		NB	NB
Fall 2	ZE			PS		NB
Fall	PS	PB			PS	NB

5-period -10-period (Big ⊕): Rise 1
5-period -10-period (Small ⊕): Rise 2
5-period -20-period (Very Big ⊕) Rise
5-period -10-period (Big ⊖): Fall 1
5-period -10-period (Small ⊖): Fall 2
5-period -20-period (Very Small ⊖): Fall

Gaussian functions are employed for the fuzzy membership functions and the Mamdani fuzzy inference method is applied for the rule inference. Total rules based on the moving average are 20 numbers as shown in Table 2.

The left look-up table of Table 2 shows the finally extracted rules that are used in the fuzzy prediction system. And the right box of Table 2 shows how to extract rules from the basic knowledge that is from expert's experience. Following information explains how to extract logics for inference.

3 Experimental Results

3.1 Simulation Conditions

In this paper, the results of the neural network model, DPNN model, and fuzzy model optimized by simulated annealing are compared together. The two inputs of the networks are daily KOSPI 200 and the industrial index announced by Korea Stock Exchange. The all inputs consist of today's index and yesterday's index. The training data of the neural networks are KOSPI 200 that is chosen by the data time from Jan. 4 1999 to Dec. 26 2000 and the testing data are selected by the data time from Jan. 2 2001 to Aug. 28 2001. Figure 1 shows the three models to predict the KOSPI 200.

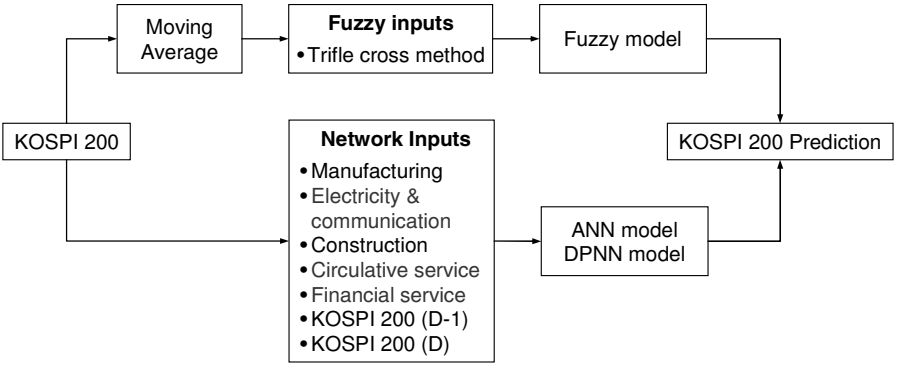
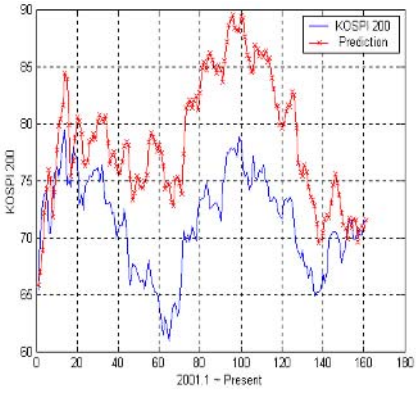


Fig. 1. The scheme of the approaches that include a fuzzy system and two network systems

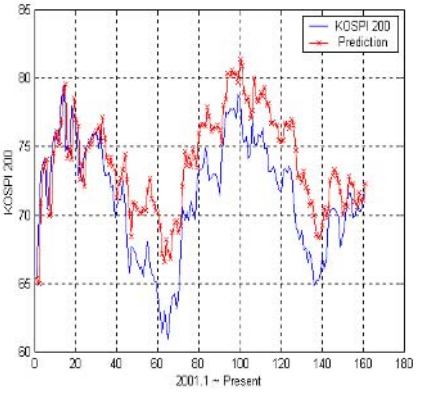
3.2 Prediction Results

Three models were trained by the data from Jan. 4 1999 to Dec. 26 2000. The training results show that three models are trained well by using the training data. It means that the procedure of the training is working properly.

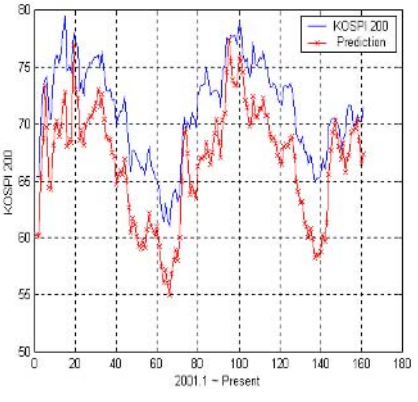
Figure 2 shows the prediction results of three models that are taken the data from Jan. 2 2001 to Aug. 8 2001. In the experimental results, the performance of the DPNN model is the best for the prediction of the stock price index. The fuzzy model shows the better performance than the performance of the neural network model. Table 3 presents RMSE (root-mean squared error) that is one of performance criteria. The performance is better with lower values. In the table, the DPNN model has the lowest



(a) Result of the neural network model



(b) Result of the DPNN model



(c) Result of the fuzzy model

Fig. 2. The prediction results of the three models

Table 3. The performance index using RMSE

Error \ Model	Neural network	DPNN	Fuzzy logic
RMSE	8.3852	**3.3703**	4.9131

RMSE value so this model is the best for the performance. This result is the same with the result of the scatter plot. Next good model is the fuzzy model that has the 4.9131 RMSE value.

4 Conclusions

The main goal of this research is to design the intelligent models using the history data and knowledge information. For the knowledge-based model, the fuzzy logic is

employed and the optimization method is applied for the optimization of the membership functions. In this paper, the prediction target is KOSPI 200 that is the typical non-linear data. The neural network model uses the just history data for the model inputs but the fuzzy model takes the knowledge-based rules that are extracted from data, user's knowledge, and so on. If the knowledge information is more correct to the real world index, the result of the prediction should be getting better. In this paper, the moving average is used for generating rules but it could be change with respect to the knowledge what the user wants to apply for the prediction models.

Acknowledgement

This paper was supported by Research Institute of Computer, Information and Communication (PNU).

References

1. Murphy, J.J.: Technical Analysis of the Financial Markets. Prentice Hall, New York (1999)
2. Wuthrich, B., Cho, V., Leung, S., Sankaran, K., and Zhang, J.: Daily Stock Market Forecast from Textual Web Data. 1998 IEEE International Conference on Systems, Man, and Cybernetics (1998) 2720-2725
3. Lee, C.C.: Fuzzy Logic in Control Systems: Fuzzy Logic Controller, Part I and II. IEEE Transactions on Systems, Man and Cybernetics **20**(2) (1990) 404-435
4. Zadeh, L.A.: Outline of a New Approach to the Analysis of Complex System and Decision Process. IEEE Transactions on Systems, Man and Cybernetics 3 (1973) 28-44
5. Pham, D.T. and Xing, L.: Neural Network for Identification, Prediction and Control. Springer-Verlag Inc, New York (1995)
6. Warakagoda, N.D. and Johnsen, M.H.: Multilayer Perceptron Based Optimal State Space Reconstruction for Time Series Modeling. International Joint Conference on Neural Networks (1999) 2639-2642
7. Ivakhnenko, A.G.: The Group Method of Data Handling in Prediction Problem. Soviet Automatic Control **9**(6) (1976) 21-30

Modular Neural Network Rule Extraction Technique in Application to Country Stock Cooperate Governance Structure

Dang-Yong Du[1], Hai-Lin Lan[1], and Wei-Xin Ling[2]

[1] School of Business Administration, South China University of Technology, Guangzhou, China
ddangyong@21cn.com
[2] School of Mathematical Science, South China University of Technology, Guangzhou, China
lingweixin@21cn.com

Abstract. Neural networks learn knowledge from data. For a monolithic structure, this knowledge can be easily used but not isolated. The many degrees of freedom while learning make ruler extraction a computationally intensive process as the representation is nor unique. Based on the technology of modular neural network data mining, this paper applied modular neural network ruler extraction to the data mining of country stock cooperate governance structure. Meanwhile, it investigated the relationship among gerentocratic constitutes of country stock cooperate, farmers' educational level, labor force and corporation performance of country stock cooperate.

1 Introduction

Neural networks are assumed to extract knowledge by learning from presented data. The monolithic neural network has serious learning problems, easily forgets initialization settings and stores the knowledge in a sparsely spread and irreproducible format that makes rule retrieval a complicated and ill-defines process [12]. The retrieval problem with monolithic networks can be solved by proper network design, but such scales very badly with increasing complexity. Therefore it has been suggested to use a modular structure for knowledge representation.

The major benefit of modular networks follows from the inherent separation of concerns. Modular networks can be based on a natural partition of the problem space reflecting the domain knowledge. Where such knowledge is not fully available, but a linguistic description is feasible, fuzzy modeling can be applied. The decide-and-conquer approach uses the domain knowledge displayed by the modular structuring for an hierarchical retrieval and will benceforth result in a more tractable rule set that is better suited for human inspection and interpretation.

The architecture of Parallel Cooperative Modular Neural Network based on Gradient (GPCMNN)[1] is fitted for solving practical problems on a large scale. By means of ' divide and conquer', its algorithm has the advantage of automatic decomposition of given tasks and training tactics judgment and module. Experiments show that the architecture is fast in training speed and effective in network performance, superior to

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 477 – 484, 2006.

non-modular methods due to its high efficiency for a parallel network system structure. Meanwhile, it keeps the PCMNN's [7] advantages and improves its drawbacks.

Extracting symbol rulers from neural networks will enhance the intelligibility of neural networks. Since Gallant [6] opened the domain of extracting rulers from neural networks, especially with the rise of data mining in early 1990s, more and more scholars in machine learning and neural computing fields have realized the significance of extraction rulers from neural networks. They devoted to the researches, and have made endless achievements [7], [8], [9].

Data mining can mine knowledge from large amounts of data. Neural network has become an important method for data mining due to its distributing memory and high-speed parallel computing ability. Neural network is a blackbox model whose learned knowledge is concealed in a large amount of connections. This has not only weakened the confidence of users in building intelligent systems using neural computing techniques, but also hindered the application of neural networks to data mining. Since extraction rules from neural networks help to solve those problems, this area has become a hot topic in both machine leaning and neural computing communities.

Country stock cooperate is an economic and organizational pattern which appeared in China in 1980s. It is viewed as a breakthrough in Chinese rural reform after the contractual household output-related responsibility system [13], and the target pattern of urban and rural economic reform in China. Moreover, scholars believes that the governance structure by legal persons under the stock cooperate is an optimal pattern for country credit cooperatives. At present, due to the lack of external corresponding policy and mature capital market, the governance structure of country stock cooperate is mainly reflected on its internal governance structure. Following stock corporate, it sets up the shareholder congress, the board of directors, supervisory committee and managers. Shareholder congress is the topmost power institution, its foundation that shareholder came into being is membership but not property right. This paper put forward a ruler extraction algorithm based on modular neural network, and applied it to the data mining of country stock cooperate governance structure, aiming to investigate the relationship among gerentocratic constitutes of country stock cooperate, farmers' educational level, labor force and corporation performance of country stock cooperate.

2 The Technique of Extraction Rulers Based Modular Neural Networks

The system of extraction rulers based modular neural networks proposed in this paper is made up of Data Preparation Module (DPM), Decomposition Module (DM), Sub-Task Net (STN) and Ensemble Module (EM). The structure is shown in Fig.1.

2.1 Data Preparation Module

The Data Preparation Module (DPM) mainly includes data integration, data cleaning, data transformation and data reduction. Data cleaning works to correct the conflicting data, through filling up missing data, eliminating abnormal data and smoothing noisy data. Data integration is a combination of many data from different sources. Data transformation is mainly to make the data formalized. The aim of data reduction is to reduce the scale of data mining without influencing the mining results.

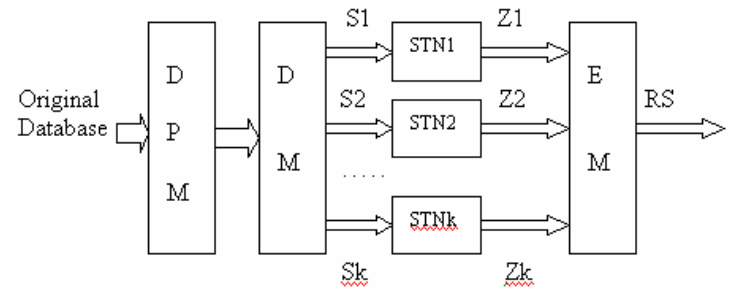


Fig. 1. The structure of extraction rulers based modular neural networks

2.2 Decomposition Module

By means of decomposition technique of modular neural networks in Decomposition Module (DM)[1], it can produce K sub data sets according to the classification of the attributive variable. The K sub data set is noted as S_1, S_2, … S_k.

$S_i=\{(X^i,T^i)|\ X^i$ is sub set of input vector, T^i is sub set of target vector$\}$.

2.3 Sub-task Net and Ruler Extraction

Each sub data set corresponds to a Sub-Task Net (STN). The STN is a 3 layers of feedforward neural network, which structure is 'n-m-p'. When the net is trained well, the following algorithm is used to extract original ruler based on structure analysis. It can produce K numbers of original rulers, noted as Z_1, Z_2, … Z_k.

Supposing in the well trained network, using w_{ji} to denote the weight between input layer node i to hidden layer node j, and w_{kj} denote the weight between hidden layer node j to output layer node k. The transfer function of hidden nodes be sigmoid function and output nodes be linear function.

1. First, produce the rulers of hidden layer:

$$if\ (\sum_{i=1}^{n} w_{ji} X_i \geq \theta_j + \Delta) \xrightarrow{cf} consequent_j \tag{1}$$

where θ_j is the threshold of hidden node j, and Δ is parameter of reliability. $\Delta \in [0.3, 0.5]$. *cf* is the reliability, which defined as followed,

$$cf = \frac{1}{1+\exp(\sum_{i=1}^{n} w_{ji} X_i - \theta_j - \Delta)} \tag{2}$$

2. Elicit the result of X_i to satisfy (1), $X_i = \{e_1, e_2, \ldots, e_n\}$
   Where $X_i = [a_i, b_i]$, ai, bi is the upper limit and the lower limit of X_i, $e_i \in [a_i, b_i]$
3. Produce a ruler for hidden layer.
   (1) Educe all the item for a ruler, each manely I_i

```
If      w_ji >0 then X_i > e_i
Else                 X_i < e_i
```

(2)Produce a ruler for hidden layer. The format as follows

$$if\ I_1\ and\ I_2\ and\ \ldots\ and\ I_n \xrightarrow{cf} h_j \tag{3}$$

4. Then the ruler of output layer is produced as bellowed format

$$if\ h_1\ and\ h_2\ and\ \ldots\ and\ h_m \xrightarrow{cf} O_k \tag{4}$$

5. The format of ruler is as fellows,

<boolean expression> [and <boolean expression>] →consequent

where, boolean expression::=variable | operator | constant; operator::=> | < |≥| ≤;

If the weight of node i to node j is nearly to zero then node i and node j are nearly irrelevant (or not largely relevant)is educed.

These original ruler sets are input to Ensemble Module (EM).

2.4 Ensemble Module

Ensemble Module (EM) integrates K original ruler sets into one ruler set. The ensemble algorithm can be described as follows.

1. Check each original ruler set Zi and remove those conflicting rulers that the preconditions are conflicted.
2. Sort all the original set Zi (i=1, … k) by input vector.
3. By means of logical ‘ or ’ to combine relating rulers if their consequents are the same and their correspond antecedents are different.
4. In the end, the finally, ruler set (RS) is produced.

3 Application in Country Stock Cooperate Governance Structure

Since 1980s, the urbanization of rural areas nearby the southern coastal cities has become speedy. Increasingly huge collective assets and a large number of financial compensation from levying ground by government turned farmers in some places into new-riches overnight. The existing institutions such as the village committee and party branch thus became the supervisor of stock company. Suddenly these common farmers were turned into manipulators of huge capital. On one hand, they are independent legal persons, and on the other hand, they are mostly illiterate farmers who have no business experiences and knowledge of running a company. What is the performance of these country stock corporations controlled by these farmers? What is the relationship between the corporation performance and the constitute of these administrators? These questions are worth exploring and investigating.

Over two hundred data from every village among hundreds in Guangzhou are obtained including such factors as the basic situation of village committee and party branch members. Along with these data, the whole number of labor, population, educational level, village asset value and the total profits in 2004 are all collected. This application aims to study the relations among profits, administrators, labors and their educational levels.

3.1 Data Preparation

First, data cleaning is carried out for the collected data based on related knowledge. Then according to data mining technique, data transformation and data reduction are processed. Finally independent variables and attributive variables are obtained as fellow:

independent variable: (22 in total)

Basis, namely set A, which include total population, educational level, business registration, years of reform, total assets, namely A1, A2, A3, A4 and A5 correspond)

Stock structure, namely set B, which include the numbers of representative stockholder, the numbers of stockholder, the numbers of meeting, is/isn't collectivity stock, namely B1, B2, B3 and B4 correspond.

Directorate member, namely set C, which include the number, educational level, age, working years, number of study and the numbers of directorate meeting, namely C1, C2, C3, C4, C5 and C6 correspond.

Party member, namely set E, which include the number, educational level, age, working years, numbers of study, namely D1, D2, D3, D4 and D5 correspond.

Prompting, namely set E, which include total income of managers, proportion of float salary, namely E1 and E2 correspond.

attributive variable:(only 1)

Agent cost, namely set H, which include total asset profit rate, expense rate, namely H1 and H2 correspond.

Because Sub Task Net is feedforward neural network and uses the algorithm (in section 2.3) to extract rulers which suppose all the input data to STN have the same value range, the original data must be deal with in advance as fellows:

$$z_i = \frac{1}{1+\exp(-(\frac{x_i - \mu_i}{2\sigma_i}))} \tag{5}$$

In the equation (5), z_i is the value after the treatment of original input variable x_i, σ_i is the standard deviation of input X, μ_i is the mean value of input X. 240 groups of data are chosen as the training data and 60 groups of data as the testing data from the total 300 groups of data.

3.2 Decomposition

The decomposition module technique of modular neural networks is employed in Decomposition Module (DM). It partitions attributive variable H1 and H2 into 3 classes according to Automatic Generating Hierarchy Formulation algorithm (AGHF algorithm) [11] correspond, namely SH1, MH1, LH1, SH2, MH2 and LH2.

SH1=[0, 0.4], MH1=[0.4, 0.6], LH1=[0.6, 1.0]
SH2=[0, 0.4], MH2=[0.4, 0.6], LH2=[0.6, 1.0]

Then original data set can be partitioned into 6 sub data sets as sh1, mh1, lh1, sh2, mh2, lh2. The independent variables have their 3 linguistic values S (small= [0, 0.4]), M (middle=[0.4, 0.6]) and L (large=[0.6, 1.0]).

3.3 Sub Task Net Train and Rulers Extraction

There are 4 combinations of data sets to be ruler extracted. Each STN is feedforward neural network. The updates weight and bias values of the neural network according to Levenberg-Marquardt optimization offered by MATLAB Toolbox and its transfer function of hidden nodes be sigmoid function and transfer function of output nodes be linear function. The structure of each combination and the extraction results are in Table 1.

Table 1. The results of 4 combination data sets.

Data sets	Structure of STN	Numbers of STN	Numbers of rulers for *cf*=1
A, B, H	9-2-1	6	10
A, C, H	11-2-1	6	16
A, D, H	10-2-1	6	14
A, E, H	7-2-1	6	9

3.4 Ruler Set Integration and Test

The extraction rulers for the combination of set A, B and H, which *cf*=1, is shown as follows:

1. if A1<M and A2<M and A4>M and A5>L and B1>M and B2<M and B4>=L → H1=SH1
2. if A1>M and A2>M and A3>=S and A4=L and A5>=L → H1=MH1
3. if A1>M and A2<M and A4>=L and A5>=L and B1<M and B2>M and B3>=L and B4>=S → H1=MH1
4. if A2>S and A3>=Land B3>M and B4>=L → H1=LH1
5. if A1<M and A2>M and A5>=L and B2>M and B3>M and B4>=S → H1=LH1
6. if A3>=L and A5<M and B1>M and B2<L and B4>=L → H2=SH2
7. if A2>M and A3>=L and A4<M and A5<M and B2<M and B4>=L → H2=SH2
8. if A1>=L and A2>=M and A3>=L and B1<M and B2<M and B3>=H and B4>=S → H2=MH2
9. if A1<M and A2<M and A3>=S and A4>=L and A5>=L and B1<M and B2>M and B4>=L → H2=MH2
10. if A1<M and A2>=M and A3>=S and B1>=M and B2>=L and B3>=L and B4>=L → H2=LH2

3.5 Result Analysis

Based on the above analysis, some findings are proposed for the study of modular neural network data mining technique in country stock cooperate governance structure:

1. The fewer party members and their shorter working years will lead to the lower capital profit of the country stock cooperate corporation;
2. The number of working years of party branch members is almost irrelevant to the capital profit of the country stock cooperate corporation;

3. The higher educational level of the party branch members will lead to the higher capital profit of the country stock cooperate corporation;
4. The higher salary level of party branch member, the larger number, the longer working years and the higher salary of directorate members can lead to the higher asset profit of the country stock cooperate corporation;
5. The larger number of directorate members and their higher educational level will lead to the higher capital profit of the country stock cooperate corporation;
6. The higher salary of directorate members and the higher educational level of villagers will lead to the higher capital profit of the country stock cooperate corporation;
7. The working years of directorate and party branch member are almost irrelevant to the capital profit margin of country stock cooperate corporation;
8. The higher educational level of villagers will lead to the higher capital profit of the country stock cooperate corporation.

4 Conclusion

The technique of extraction rulers based modular neural networks proposed in this paper partitions the whole data sets into sub-sets. By means of ‘ divide and conquer ’, it is fast in training speed and effective in network performance due to its high efficiency for a parallel network system structure. Applying this technique to the study of country stock cooperate governance structure, it is found that such factors as the mean educational level of party branch members and directorate members, their number, working years, the educational level of villagers and labor ratio can take great influence on the performance of country stock cooperate corporation, indicating the profound practical value of this technique.

Acknowledgment

This study belongs to the key philosophical and scientific research the growth and reorganization of Chinese enterprises under the environment of economic globalization’ put forward by ministry of education, whose project number is 04jzd0018, and the grant 04300102 from the NSF of Guangdong, China have supported this work, too.

References

1. Ling, W.X., Zheng, Q.L., Chen, Q.: GPCMNN: A Parallel Cooperative Modular Neural Network Architecture Based On Gradient. Chinese Journal of Computers 27(9)(2004) 1256-1263 (In Chinese)
2. Ling, W.X., Zheng, Q.L., Chen, Q., Lu, C.Y.: PCMNN: A Parallel Cooperative Modularied Neural Network Architecture. In: Proceedings of IEEE FUZZ2001 (2001), vol.1. 268-271
3. Ling, W.X., Zheng, Q.L., Chen, Q., Lu C.Y.: A Modular Neural Network Architecture Based on Parallel Cooperation, Journal of Beijing University of Post and Telecommunications 25(2)(2002) 14-17 (in Chinese)

4. Murre, J. M. J.: Learning and Categorization in Modular Neural Networks. London, U.K.: Harvester- Wheatsheaf (1992)
5. Lu, B., Kita, H., Nishikawa, Y.: A Multi-sieving Neural Network Architecture that Decomposes Learning Tasks Automatically. In: World Cong. Comput. Intell. 13 (1994)1319-1324.
6. Gallant, S.I.: Connectionist Expert Systems. Communications of the ACM 31(2) (1988) 152-169
7. Krishnan, R., Sivakumar, G., Bhattacharya, P.: Extraction Decision Trees from Trained Neural Networks. Pattern Recognition 32(12) (1999) 1999-2009
8. Setiono, R.: Extracting M-of-N Rules from Trained Neural Networks. IEEE Trans. on Neural Networks 11(2) (2000) 512-519
9. Castro, J.L., Mantas, C.J., Benitez, J.M.: Interpretation of Artificial Neural Networks by Means of Fuzzy Rules. IEEE Trans. on Neural Networks 13(1)(2002) 101-106
10. Aref, W.G., Samet, H.: Optimization Strategies for Spatial Query Processing. In Proceedings of 17th Int. Conf. on Very Large Data Bases. Barcelona. Spain. Sept. (1991) 81-90
11. Han, J., Fu, Y.: Dynamic Generation and Refinement of Concept Hierarchies for Knowledge Discovery in Databases. In: Proceedings of AAAI'94 Workshop in Knowledge Discovery and Databases. Seattle: WA (1994) 157-168
12. Auda, G., Kamel, M.: Modular Neural Networks: A Survey. Int. Journal of Neural Systems 9(2) (1999)129-151
13. Han, J., Zhang Q.Z.: Essential Signification of Importance Breach in Country Reform. Freely talk about agriculture stock cooperate systems 2(9) (1998)8-12 (In Chinese)

A Hybrid Support Vector Machines and Discrete Wavelet Transform Model in Futures Price Forecasting

Fan-yong Liu and Min Fan

Financial Engineering Research Center,
South China University of Technology,
Wushan, Guangzhou, P.R. China
fanyongliu@163.com

Abstract. This paper is motivated by evidence that different forecasting models can complement each other in approximating data sets, and presents a hybrid model of support vector machines (SVMs) and discrete wavelet transform (DWT) to solve the futures prices forecasting problems. The presented model greatly improves the prediction performance of the single SVMs model in forecasting futures prices. In our experiment, the performance of the hybrid is evaluated using futures prices. Experimental results indicate that the hybrid model outperforms the individual SVMs models in terms of root mean square error (RMSE) metric. This hybrid model yields better forecasting result than the SVMs model.

1 Introduction

The financial market is a complex, evolutionary, and nonlinear dynamical system. As explained by Deboeck [1], financial time series are inherently noisy, non-stationary and deterministically chaotic. Thus, forecasting financial time series has been regarded as the most challenging applications of time series forecasting.

Recently, a novel neural network algorithm, called support vector machines (SVMs), was developed by Vapnik and his co-workers [2]. SVMs implement the structural risk minimization (SRM) principle, which seeks to minimize an upper bound of the generalization error rather than minimize the training error. Based on this principle, SVMs have better generalization performance than other neural network models. Another merit of SVMs is that the training of SVMs is equivalent to solving a linearly constrained quadratic programming. This means that the solution of SVMs is unique, optimal and absent from local minima. Originally, SVMs have been developed for pattern recognition problems. However, with the introduction of Vapnik's ε-insensitive loss function, SVMs have been extended to solve non-linear regression estimation problems and have been shown to exhibit excellent performance [3, 4].

Different forecasting models can complement each other in capturing patterns of data sets, and both theoretical and empirical studies have concluded that a combination of forecast outperforms individual forecasting models [5, 6]. On the other hand, Wavelet methodology is able to decompose the time series data into

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 485–490, 2006.

several time scales based only on the subset of information that is of interest to researchers. Yousefi et al. [7] illustrate an application of wavelets as a possible vehicle for investigating the issue of market efficiency in futures markets for oil. He illustrates the extension of the last half of the signal based on the spline and sine fit. This study is partly motivated by these contributions.

This study consists of five sections. Section 2 provides a brief introduction to SVMs and DWT. while Section 3 contains the experimental data and the criteria of the prediction performance. Section 4 discusses the experimental results followed by the conclusions drawn from this study.

2 Hybrid Model in Forecasting

2.1 Support Vector Machines

Suppose there is a given set of data points $G = \{(x_i, d_i)|i = 1, 2, \cdots, n\}$($x_i$ is the input vector, d_i is the desired value, and n is the total number of data patterns) drawn independently and identically from an unknown function, SVMs approximate the function with three distinct characteristics: (i) SVMs estimate the regression in a set of linear functions, (ii) SVMs define the regression estimation as the problem of risk minimization, and (iii) SVMs minimize the risk based on the SRM principle. The linear function is formulated in the high dimensional feature space, with the following form of function:

$$y = f(x) = w\phi(x) + b \tag{1}$$

where $f(x)$ is the high dimensional feature space, which is nonlinearly mapped from the input space x. Characteristics (ii) and (iii) are reflected in the minimization of the regularized risk function (2) of SVMs, by which the coefficients w and b are estimated. The goal of this risk function is to find a function that has at most ε deviation from the actual values in all the training data.

$$R_{\mathrm{SVMs}}(C) = C\frac{1}{n}\sum_{i=1}^{n} L_\varepsilon(d_i, y_i) + \frac{1}{2}\|w\|^2 \ . \tag{2}$$

$$L_\varepsilon(d, y) = \begin{cases} |d - y| - \varepsilon, & |d - y| \geq \varepsilon, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

In (2), the first term is the empirical error (risk), which is measured by the ε-insensitive loss function (3). The second term in (2) is called the regularized term; ε is called the tube size of SVMs, and C is the regularization constant determining the trade-off between the empirical error and the regularized term.

Finally, by introducing Lagrange multipliers a_i, a_i^* and exploiting the optimality constraints, decision function (1) takes the following form:

$$f(x, a_i^{(*)}) = \sum_{i=1}^{n}(a_i - a_i^*)K(x, x_i) + b \ . \tag{4}$$

In (4), $K(x_i, x_j)$ is the kernel function. The value is equal to the inner product of two vectors x_i and x_j in the feature space $\phi(x_i)$ and $\phi(x_j)$. That is, $K(x_i, x_j) = \phi(x_i) * \phi(x_j)$. The kernel functions used in this study are the Gaussian radial basis function (RBF) and the splines. Gaussian RBF takes the following form:

$$K(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|}{2\sigma^2}) \ . \tag{5}$$

The splines kernel function takes the following form:

$$K(x_i, x_j) = 1 + < x_i, x_j > + \frac{1}{2} < x_i, x_j > \min(x_i, x_j) - \frac{1}{6}\{\min(x_i, x_j)\}^3 \ . \tag{6}$$

2.2 Discrete Wavelet Transform

In the wavelet world, any function $f(t)$ in a square-integrable functional space can be expressed as a sequence of projections onto father and mother wavelets generated from functions φ and ψ through scaling and translation as follows:

$$\varphi_{j,k}(t) = 2^{-\frac{j}{2}}\varphi(2^{-j}t - k) = 2^{-\frac{j}{2}}\varphi(\frac{t - 2^j k}{2^j}) \tag{7}$$

$$\psi_{j,k}(t) = 2^{-\frac{j}{2}}\psi(2^{-j}t - k) = 2^{-\frac{j}{2}}\psi(\frac{t - 2^j k}{2^j}) \ . \tag{8}$$

The wavelet representation of the function $f(t)$ can now be given as:

$$f(t) \approx \sum_k a_{J,k}\varphi_{J,k}(t) + \sum_k d_{J,k}\psi_{J,k}(t) + \cdots + \sum_k d_{1,k}\psi_{1,k}(t) \ . \tag{9}$$

The coefficients $a_{J,k}$, $d_{J,k}$, $\cdots$, $d_{1,k}$ are the wavelet transform coefficients. The detail coefficients $d_{J,k}$, $\cdots$, $d_{1,k}$ can capture the higher frequency oscillations, while $a_{J,k}$ can capture the trend.

The multi-resolution decomposition of a signal can now be defined as:

$$a_J = \sum_k a_{J,k}\varphi_{J,k}(t), \quad d_j = \sum_k d_{j,k}\psi_{j,k}(t), \quad \text{for} \quad j = 1, 2, \cdots, J \ . \tag{10}$$

Similar to the wavelet representation (9), a signal $f(t)$ can now be expressed as:

$$f(t) = a_J + d_J + d_{J-1} + \cdots + d_1 \ . \tag{11}$$

2.3 The Hybrid Methodology

First, the DWT is able to decompose the time series data $f(t)$ into five time scales a_5, d_5, d_4, d_3, d_2, d_1. Based on the different scales, the different kernel functions, such as Gaussian RBF and the splines, are selected in order to perform support vector regression (SVR) and forecasting. Finally, the forecasting values of different time scales, $\hat{a}_5$, $\hat{d}_5$, $\hat{d}_4$, $\hat{d}_3$, $\hat{d}_2$, $\hat{d}_1$, are reconstructed as $\hat{f}(t)$. This hybrid strategy can be expressed as:

$$f(t) \overset{DWT}{\longrightarrow} \left\{ \begin{array}{l} a_5 \overset{SVR-Spline}{\longrightarrow} \hat{a}_5 \\ d_5 \overset{SVR-Spline}{\longrightarrow} \hat{d}_5 \\ d_4 \overset{SVR-Spline}{\longrightarrow} \hat{d}_4 \\ d_3 \overset{SVR-Spline}{\longrightarrow} \hat{d}_3 \\ d_2 \overset{SVR-RBF}{\longrightarrow} \hat{d}_2 \\ d_1 \overset{SVR-RBF}{\longrightarrow} \hat{d}_1 \end{array} \right\} \overset{Reconstruct}{\longrightarrow} \hat{f}(t) \ . \tag{12}$$

3 Data Sets and Performance Criteria

Data come from the wheat futures prices (RMB/ton) for futures contracts nearby delivery month in Zhengzhou Commodity Exchange (ZCE). The original data cover the period May 1993 to December 2004. The number of data patterns is 140. In this study, only one-step-ahead forecasting is considered. There are a total of 100 data patterns in the training set, 40 data patterns in both the validation set and the test set in all the data sets. RMSE metric is used as measure of forecasting accuracy.

4 Experimental Results

4.1 Results of SVMs

For the SVMs models, three parameters: σ^2, ε and C, need to be prescribed. As described in [3], the choice of σ^2=50, ε=0.01 and C=1000 is appropriate. Fig.2 illustrates the results by SVMs with RBF and splines kernel functions. Based on Fig.2, RBF outperforms comparatively splines in forecasting performance.

4.2 Results of Hybrid Method

As described in [7], the Daubechies' wavelets of order 7 outperform the other wavelets. A five level wavelet decomposition of the given f is performed: $f(t) = a_5 + d_5 + d_4 + \cdots + d_1$. The smooth part of f is stored in a_5, and details on different levels are captured by d_1, d_2,..., and d_5.

Fig.1 illustrates the forecasting values of the last 40 data based on the first 100 data. The spline kernel function is applied to forecast the last 40 data of a_5, d_5, d_4, d_3 while the RBF kernel function is used to forecast the last 40 data of d_1, d_2. Table 1 shows RMSE values of forecast between the forecasting and original decomposed data on each level based on two kernel function types above.

Fig.2 illustrates the different results of SVMs and hybrid methods. Those results indicate that the hybrid model outperforms the other two individual SVMs models in terms of RMSE metric. The hybrid model is can reduce the overall forecasting errors. The results in Table 2 indicates the proposed hybrid model is superior to SVMs models.

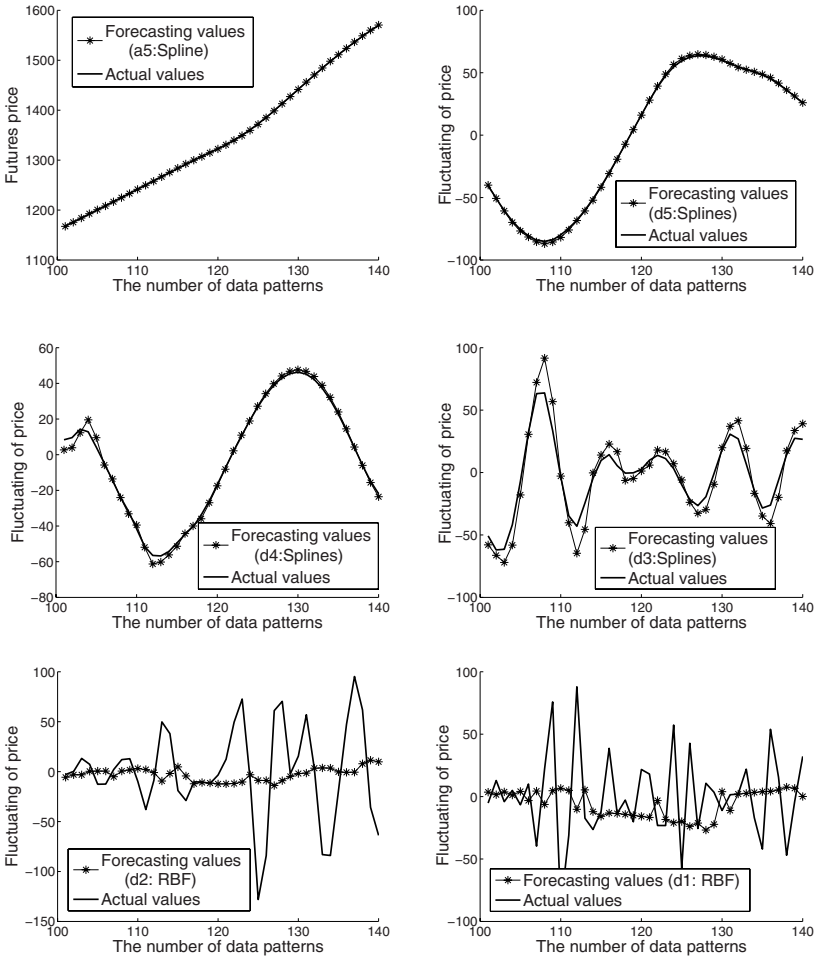


Fig. 1. The forecasting values and actual values of different time scales

Table 1. RMSE values based on two fit types

Level		d_1	d_2	d_3	d_4	d_5	a_5
Kernel function type	Spline	114.6	55.85	10.72	2.34	1.01	0.42
	RBF	38.47	48.37	33.48	37.10	80.76	46.26

Table 2. RMSE values based on the different models

Models	SVMs (Kernel: Spline)	SVMs (Kernel: RBF)	Hybrid model
RMSE	125.375	119.567	65.230

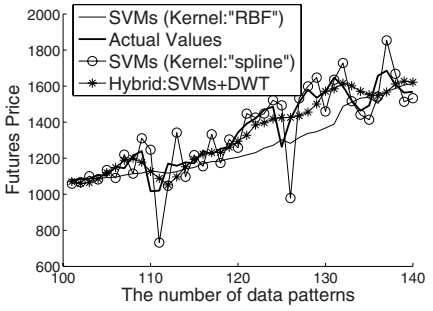


Fig. 2. Forecasting value of futures prices

5 Conclusion

This study presents a hybrid model of SVMs and DWT to solve the futures prices forecasting problem. The kernel functions are selected by taking into account the non-stationary characteristic of financial time series. The simulation results demonstrated that the hybrid model is effective in dealing with the forecasting problem of financial time series. This study demonstrated that a hybrid of two individual models produces the best results.

This study has several limitations. First, the study has considered and calculated only the out-of-sample forecast for one period ahead. Second, other types of kernel functions will be explored for further improving the performance of SVMs in financial time series forecasting. Additionally, the structured selection of optimal parameters of the hybrid model is of great interest.

References

1. Hall, J.W.: Adaptive Selection of US Stocks with Neural Nets. In: Deboeck, G.J. (ed.): Trading On the Edge. Neural, Genetic, and Fuzzy Systems for Chaotic Financial Markets. John Wiley & Sons, New York (1994) 45–65
2. Vapnik, V.N.: The Nature of Statistical Learning Theory. 1rd edn. Springer-Verlag, Berlin Heidelberg New York (1995)
3. Tay, F.E.H., Li-Juan, C.: Application of Support Vector Machines in Financial Time Series Forecasting. The International Journal of Management Science **29** (2001) 309–317
4. Tay, F.E.H., Li-Juan, C.: Modified Support Vector Machines in Financial Time Series Forecasting. Neurocomputing **48** (2002) 847–861
5. Peter, Z.G.: Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model. Neurocomputing **50** (2003) 159–175
6. Ping-Feng, P., Chih-Sheng, L.: A Hybrid ARIMA and Support Vector Machines Model in Stock Price Forecasting. The International Journal of Management Science **33** (2005) 497–505
7. Yousefi, S., Weinreich, I., Reinarz, D.: Wavelet-Based Prediction of Oil Prices. Chaos, Solitons and Fractals **25** (2005) 265–275

A Novel Learning Network for Option Pricing with Confidence Interval Information

Kyu-Hwan Jung, Hyun-Chul Kim, and Jaewook Lee

Department of Industrial and Management Engineering,
Pohang University of Science and Technology,
Pohang, Kyungbuk 790-784, Korea
jaewookl@postech.ac.kr

Abstract. Nonparametric approaches for option pricing have recently emerged as alternative approaches that complement traditional parametric approaches. In this paper, we propose a novel learning network for option-pricing, which is a nonparametric approach. The main advantages of the proposed method are providing a principled hyper-parameter selection method and the distribution of predicted target value. With these features, we do not need to adjust any parameters at hand for model learning and we can get confidence interval as well as strict predicted target value. Experiments are conducted for the KOSPI200 index daily call options and their results show that the proposed method works excellently to obtain prediction confidence interval and to improve the option-pricing accuracy.

1 Introduction

Prompted by shortcomings of modern parametric option-pricing models, nonparametric approaches, which do not rely on pre-assumed models but instead try to uncover/induce the model, have recently attracted attention of researchers. Among the various nonparametric methods, learning networks has been demonstrated that they can be successfully used to estimate a pricing formula for options, with good out-of-sample pricing and delta-hedging performance [2, 3, 6].

Although early proposed learning networks generated smaller pricing and delta-hedging errors than parametric methods for various option pricing applications, they have a drawback in that they just give predicted target values for some unknown data without any variance information to asses the prediction. In practice, the predicted values themselves do not contain useful information for investors since the predicted values do not match exactly to the real market values. Rather than that, if the distribution of the predicted value can be given, it can provide more useful information to investors or traders for their financial use such as risk hedging strategies and portfolio management.

To overcome such a drawback, in this paper, we propose a novel learning network for option pricing that gives the variance information which can be utilized to calculate the confidence interval of target value. The proposed algorithm is applied to KOSPI200 call options pricing and is shown to have superiority not

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 491–497, 2006.

only in its improved pricing performance but also in its availability of presenting prediction value distribution.

2 Parametric vs. Nonparametric Methods for Option Pricing

Since a major breakthrough in the pricing of stock options made by Black, Scholes, and Merton in the early 1970s, Black-Scholes model has had a huge influence on the way that traders price and hedge options. The celebrated original Black-Scholes pricing formula has a closed form based on a dynamic-hedging argument and no arbitrage condition, but has shown systematic and substantial bias because of its unrealistic assumptions. To improve pricing performance, Black-Scholes formula has been generalized to various parametric option pricing models while closed form expressions are not available. However, as shown in [7], these parametric option pricing models didn't succeed and the resulting models are too complex, have poor out-of-sample performance. Even the most complex modern parametric models are imperfect and are outperformed by simple, less general models. They often produce parameters which is inconsistent with underlying time series and inferior hedging, and retain systematic price bias they were intended to eliminate [1].

Nonparametric methods have recently emerged as alternative approaches. These methods typically include highly intensive, model-free approaches that complement traditional methods. These nonparametric methods have the distinct advantage of not relying on specific assumptions about the underlying asset price dynamics and is therefore robust to specification errors which might adversely affect parametric models [3].

To verify the merits of nonparametric and computational methods of option pricing, the performance of these methods needs to be measured against a widely used alternative model. In this paper, Black-Scholes option pricing model is used for this purpose, which is described by

$$\begin{aligned} p^{(BS)} &= f^{(BS)}(\mathbf{x}) = SN(d_1) - Xe^{-r(T-t)}N(d_2), \\ d_1 &= \frac{\ln(S/X) + (r + \sigma^2/2)(T-t)}{\sigma\sqrt{T-t}}, \quad d_2 = d_1 - \sigma\sqrt{T-t}, \end{aligned} \tag{1}$$

where S is the underlying asset price, X is the strike price of option, $T-t$ is the time to maturity, σ is the volatility (standard deviation) of the underlying asset, r is the continuously compounded risk-free interest rate, and $N(\cdot)$ is the cumulative normal distribution function. Unless otherwise specified, in this paper, we use this set of parameters $\mathbf{x} = (S, X, T-t, \sigma, r)$ as the training input data as various option pricing models including Black-Scholes model.

3 Proposed Method and Probabilistic Framework

Suppose that we have a data set $D = \{(\mathbf{x}_i, t_i)|i = 1, 2, \ldots, N\}$ of input vectors $\mathbf{x}_i = (S_i, X_i, (T-t)_i, \sigma_i, r_i)$ and target values $t_i = p_i$, where p_i is a option

price of the i-th item $\mathbf{x}_i$. Given this data set, we wish to find (the density of) the target value $\tilde{t}$ for a new data point $\tilde{\mathbf{x}}$. Let the target function $f(\cdot)$. We put $\mathbf{f}(= \mathbf{f}_N)$ as $[f_1, f_2, \ldots, f_N] = [f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_N)]$, and put $\mathbf{f}_{N+1}$ as $[f_1, f_2, \ldots, f_N, f_{N+1}] = [f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_N), f(\mathbf{x}_{N+1})]$, where $\mathbf{x}_{N+1} = \tilde{\mathbf{x}}$.

3.1 Gaussian Process Models

Gaussian processes for regression [4, 10, 12] assume that the target function has a Gaussian process prior. This means that the density of any collection of target function values is modeled as a multivariate Gaussian density. Let $X = \{\mathbf{x}_i | i = 1, 2, \ldots, N\}, T = \{t_i | i = 1, 2, \ldots, N\}$. Writing the dependence of $\mathbf{f}$ on X implicitly, the GP prior over functions can be written

$$p(\mathbf{f}|X, \Theta) = \frac{1}{(2\pi)^{N/2}|\mathbf{C}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{f} - \boldsymbol{\mu})^\top \mathbf{C}^{-1}(\mathbf{f} - \boldsymbol{\mu})\right\}, \tag{2}$$

where the mean $\boldsymbol{\mu}$ is usually assumed to be zero $\boldsymbol{\mu} = \mathbf{0}$ and each term of a covariance matrix C_{ij} is a function of $\mathbf{x}_i$ and $\mathbf{x}_j$, i.e. $c(\mathbf{x}_i, \mathbf{x}_j)$.

As $\mathbf{f}$ has a Gaussian density, $\mathbf{f}_{N+1}$ also has a Gaussian density. Using this relationship we get the predictive density of f_{N+1} as a conditional Gaussian density

$$p(f_{N+1}|X, \Theta) = \frac{p(\mathbf{f}_{N+1}|\Theta, \mathbf{x}_{N+1}, X)}{p(\mathbf{f}_N|\Theta, X)} \tag{3}$$

$$= \frac{Z_N}{Z_{N+1}} \exp[-\frac{1}{2}(\mathbf{f}_{N+1}^T \mathbf{C}_{N+1}^{-1} \mathbf{f}_{N+1} - \mathbf{f}_N^T \mathbf{C}_N^{-1} \mathbf{f}_N)], \tag{4}$$

where Z_N and Z_{N+1} are normalization constants of $p(\mathbf{f}_N|\Theta, X)$ and $p(\mathbf{f}_{N+1}|\Theta, \mathbf{x}_{N+1}, X)$, respectively. The relationship between $\mathbf{C}_N$ and $\mathbf{C}_{N+1}$ is as follows.

$$\mathbf{C}_{N+1} = \begin{pmatrix} \mathbf{C}_N & \mathbf{k} \\ \mathbf{k}^T & \kappa \end{pmatrix}, \tag{5}$$

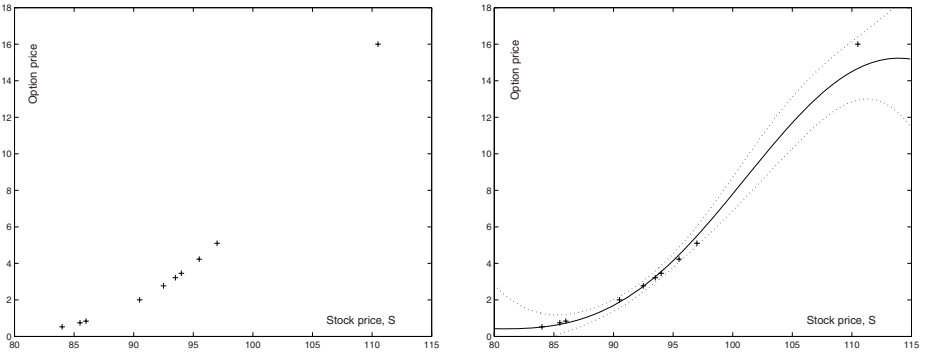
where $\mathbf{k} = [c(\tilde{\mathbf{x}}, \mathbf{x}_1), c(\tilde{\mathbf{x}}, \mathbf{x}_2), \ldots, c(\tilde{\mathbf{x}}, \mathbf{x}_n)]^T$, and $\kappa = c(\tilde{\mathbf{x}}, \tilde{\mathbf{x}})$.

By rearranging the terms in Eq. (4) we get the Gaussian density of predicted value $f_{N+1}(= \tilde{f})$ for a new data point $\mathbf{x}_{N+1}(= \tilde{\mathbf{x}})$

$$p(f_{N+1}|D, \Theta, \mathbf{x}_{N+1}) = \mathcal{N}(\mathbf{k}^T \mathbf{C}^{-1} \mathbf{t}, \kappa - \mathbf{k}^T \mathbf{C}^{-1} \mathbf{k}). \tag{6}$$

From this density, the predicted target value and its variance information can be obtained directly by the two parameters of $\mathcal{N}$. The first parameter which is the mean of distribution can be used to get interpolant while the second variance term can be used to get error bars as well demonstrated in Figure 1.

A Bayesian treatment of multilayer perceptrons for regression has been shown to converge to a Gaussian process as the number of hidden nodes approaches to infinity, if the prior on input-to-hidden weights and hidden unit biases are independent and identically distributed [9]. Empirically, Gaussian processes have been shown to be an excellent method for nonlinear regression [11].



(a) (b)

Fig. 1. (a) Simulated option training data where the stock prices and their corresponding traded option price are represented with respect to $x - y$ axes. (b) Interpolant and its 1σ error bars. The error bars represent how uncertain we are about the interpolant at each point assuming that the model is correct. Note that some points such as the point near $S = 110$ are located outside the predicted value of the constructed model but they are still located within the error interval provided by the model.

3.2 Covariance Function

The *covariance function* is the *kernel* which defines how data points generalize to nearby data points. In this paper, to improve prediction performance of option pricing, a neural network kernel is used as a covariance function. It is given by

$$K(x, x') = v_0 \frac{2}{\pi} \arcsin\left(\frac{2x^T \Sigma x'}{\sqrt{(1 + x^T \Sigma x)(1 + 2x'^T \Sigma x')}}\right) + \delta_{ij} v_2. \tag{7}$$

The covariance function is parameterized by the hyperparameters $\Theta = \{l_m | m = 1, \ldots, d\} \cup \{v_0, v_1, v_2\}$, which we can learn from the data. The covariance between the targets at two different points is a decreasing function of their distance in input space. This decreasing function is controlled by a small set of *hyperparameters* that capture interpretable properties of the function, such as the length scale of autocorrelation, the overall scale of the function, and the amount of noise.

3.3 Hyperparameter Determination

The hyperparameters are selected by maximizing the marginal likelihood (a.k.a. the evidence). The log marginal likelihood $\mathcal{L}$ is given by

$$\mathcal{L} = -\frac{1}{2} \log \det \mathbf{C} - \frac{1}{2} \mathbf{t}^T \mathbf{C}^{-1} \mathbf{t} - \frac{N}{2} \log 2\pi, \tag{8}$$

and its derivative with respect to a hyperparameter θ is

$$\frac{\partial \mathcal{L}}{\partial \theta} = -\frac{1}{2} \mathrm{tr}\left(\mathbf{C}^{-1} \frac{\partial \mathbf{C}}{\partial \theta}\right) + \frac{1}{2} \mathbf{t}^T \mathbf{C}^{-1} \frac{\partial \mathbf{C}}{\partial \theta} \mathbf{C}^{-1} \mathbf{t}. \tag{9}$$

Appropriate hyperparameters can be determined by the conjugate gradient method with this derivative as in [4]. On the other hand, the posterior distributions of the hyperparameters given the data can be inferred in a Bayesian way via Markov Chain Monte Carlo (MCMC) methods [10, 12] or they can be selected by maximizing the marginal likelihood (a.k.a. the evidence) [4].

3.4 Predictions with Confidence Interval

GPR(Gaussian Process for Regression) provides the distribution of a predicted target value, not just a strict value. In our case that noise is presumably assumed to be Gaussian, we get the Gaussian distribution of a predicted target value as in Eq(6).

$$\mu - z_{\alpha/2}\sigma \;\; \leq \;\; \tilde{f} \;\; \leq \;\; \mu + z_{\alpha/2}\sigma \tag{10}$$

with $(1-\alpha)\%$ confidence level, where the predicted mean is given by

$$\mu = \mathbf{k}^T\mathbf{C}^{-1}\mathbf{t} \tag{11}$$

and the variance for the confidence interval is given by

$$\sigma^2 = \kappa - \mathbf{k}^T\mathbf{C}^{-1}\mathbf{k} \tag{12}$$

This variance information helps the user to make a decision based on confidence intervals.

4 Experimental Results

In this section, we verify how well the proposed method works compared with Black-Scholes (BS) model and other neural network models and algorithms which include: MLP with Error Backpropagation(EBP)[6], MLP with Regularized EBP(R-EBP)[5], MLP with Bayesian Regularization Algorithm(BR)[3], Radial Basis Function Network (RBF) [5]. The data used in the empirical analysis are daily closing quotes of the KOSPI200, during the time periods January 1, 2000 - December 31, 2000. The KOSPI200 index is a value-weighted combination of the 200 most traded securities in Korea Stock Exchange market. To avoid non-synchronicity caused by trading effect, we used the closing prices of KOSPI call options as price data. We trained a network using 500 samples(76%) as a training data set and 150 samples(24%) are used to test each neural network and our proposed method. The input vector x consists of stock price, S, strike price, K, time to maturity, $T-t$, volatility, σ. Here we used σ available in the market, which was given according to the general implied volatility formula. We excluded interest rate, r, from the input vector because it changes little during the time periods for the KOSPI200 data.

Table 1. Simulation result using various prediction techniques(MSE)

Algorithm	BS model	EBP	R-EBP	BR	RBF	Proposed
Training Error	1.7681	0.6563	0.6727	0.1095	0.8620	0.2397
Test Error	2.2684	0.9939	0.7534	1.2597	0.9677	0.5273

Table 2. Simulation result for the prediction confidence interval

cf. level	Training Samples in the Range	Test Samples in the Range
95%	98.40%	98.67%
99%	99.00%	98.67%

Simulation results are shown in Table 1. We use mean squared error (MSE) on the test error reduction to measure the network pricing accuracy. In terms of MSE, the proposed method demonstrates excellent performance. EBP and R-EBP provide the moderate performances both in training and test error. BR algorithm has comparatively less train error, but shows severe test error compared to other algorithms while RBF shows relatively higher errors in both training and test. The propose method shows relatively small training and test error than various neural network models and algorithms, and especially it reduced 86% and 77% of MSE for each training and test error than the Black-Scholes model. In addition to the improved prediction performance of the proposed method for option pricing, the proposed method can provide us with the information of predicted confidence intervals for unknown data. Table 2 shows a simulation results for the correctly predicted data ratio in terms of confidence interval. In this simulation, the predicted confidence intervals verified to contain sufficient sample points, which fully demonstrate the additional benefit of the proposed method.

5 Conclusion

In this paper, we've proposed a novel method to improve prediction performance of option pricing. The proposed method employed a Gaussian process regression model with a neural network kernel for prediction. The method automatically selects optimal hyper-parameters and provides the confidence interval of prediction. To show the superiority of the proposed method, we've conducted an experiment with daily KOSPI200 Index European options obtained from the Korea Stock Exchange. The experimental results demonstrate that the proposed method is successful not only in improving generalization but also in providing the confidence intervals of predicted option prices. Application of the method is opened widely in the field of finance.

Acknowledgement. This work was supported by the KOSEF under the grant number R01-2005-000-10746-0.

References

1. Bakshi, G., Cao, C., Chen, Z.: Empirical Performance of Alternative Option-Pricing Models. The Journal of Finance 52(5) (1997) 2003-2049
2. Choi, H.-J., Lee, D., Lee, J.: Efficient Option Pricing via a Globally Regularized Neural Network . Lecture Notes in Computer Science 3174 (2005) 988-993
3. Gencay, R., and Qi, M.: Pricing and hedging derivative securities with neural networks: Bayesian regularization, early stopping, and begging. IEEE Transactions on Neural Networks 12(4) (2001) 726-734
4. Gibbs, M., Mackay, D.J.C.: Efficient Implementation of Gaussian Processes. Draft Manuscript (http://citeseer.nj.nec.com/6641.html) (1992)
5. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice-Hall New York (1999)
6. Hutchinson, J.M., Lo, A.W., Poggio, T.: A Nonparametric Approach to Pricing and Hedging Derivative Securities via Learning Networks. Journal of Finance 49 (1994) 851-889
7. Lajbcygier, P.: Literature Review: The Problem with Modern Parametric Option Pricing. Journal of Computational Intelligence in Finance 7(5) (1999) 6-23
8. Lee, J., Lee, D.: An Improved Cluster Labeling Method for Support Vector Clustering. IEEE Trans. on Pattern Analysis and Machine Intelligence 27(3) (2005) 461-464
9. Neal, R.M.: Bayesian Learning for Neural Networks. Lecture Notes in Statistics Springer-Verlag, New York (1996)
10. Neal, R.M.: Regression and Classification Using Gaussian Process Priors. Bayesian Statistics 6 (1998) 465-501
11. Rasmussen, C.E.: Evaluation of Gaussian Processes and Other Methods for Non-Linear Regression. PhD Thesis University of Toronto (1996)
12. Williams, C.K.I., Rasmussen, C.E.: Gaussian Processes for Regression. NIPS 8 (1995) 514-520

An Adaptive BP Algorithm with Optimal Learning Rates and Directional Error Correction for Foreign Exchange Market Trend Prediction

Lean Yu [1,3], Shouyang Wang [1,2], and Kin Keung Lai [2,3]

[1] Institute of Systems Science, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100080, China
{yulean, sywang}@amss.ac.cn
[2] College of Business Administration, Hunan University, Changsha 410082, China
[3] Department of Management Sciences, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong
mskklai@cityu.edu.hk

Abstract. In this study, a novel adaptive BP learning algorithm with optimal learning rates and directional error correction is proposed. In this algorithm, the optimal adaptive learning rates are used to adjust the weights in an adaptive and dynamical way. Meantime, the directional errors are fed back to learning process to increase the directional accuracy. For illustration and testing purposes the proposed algorithm is applied to the foreign exchange market trend prediction.

1 Introduction

Back-propagation neural network (BPNN) is the popular neural networks which have been widely applied to many fields such as prediction and classification problems. The basic learning rule of BPNN is based on the gradient descent method and the chain rule [1]. However, a deadly drawback of the BPNN learning rule based on the gradient descent method is its slowness and its frequent confinement to local minima [2]. For this reason, many improved BP algorithms are developed such as conjugate gradient algorithm [3] and Levenberg-Marquardt algorithm [4]. Generally, these algorithms have an improved convergence property, but most of these methods do not use the optimal learning rates. In their studies, the learning rate is set to a fixed value. However, it is critical to determine a proper fixed learning rate for BPNN learning. If the learning rate is large, learning may occur quickly, but it may also become unstable and even will not learn at all. To ensure stable learning, the learning rate must be sufficiently small, but with a small learning rate the BPNN may be lead to a long learning time. Also, just how small the learning rate should be is unclear. In such situations, it is necessary to derive optimal learning rate and then allow an adaptive change at each iteration step during the learning process. In addition, in the trend prediction, the directional error information should be fed back to the learning process to increase accuracy. In this study, a binary directional error function is used.

The main motivation of this study is to propose an adaptive BP algorithm for foreign exchange trend predication. The rest of this study is organized as follows.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 498 – 503, 2006.

Section 2 presents the proposed adaptive learning algorithm. Section 3 gives an experiment and reports the results. Finally, Section 4 concludes the paper.

2 The Adaptive Learning Algorithm

Consider a three-layer BPNN, which has p input nodes, q hidden nodes and k output nodes. Mathematically, the basic structure of the BPNN model is described by

$$Y(t+1) = \begin{bmatrix} y_1(t+1) \\ y_2(t+1) \\ \cdots \\ y_k(t+1) \end{bmatrix} = \begin{bmatrix} f_2[\sum_{i=0}^{q} f_1(\sum_{j=0}^{p} w_{ij}(t)x_j(t))v_{1i}(t)] \\ f_2[\sum_{i=0}^{q} f_1(\sum_{j=0}^{p} w_{ij}(t)x_j(t))v_{2i}(t)] \\ \cdots \\ f_2[\sum_{i=0}^{q} f_1(\sum_{j=0}^{p} w_{ij}(t)x_j(t))v_{ki}(t)] \end{bmatrix} = \begin{bmatrix} f_2[V_1^T F_1(W(t)X(t))] \\ f_2[V_2^T F_1(W(t)X(t))] \\ \cdots \\ f_2[V_k^T F_1(W(t)X(t))] \end{bmatrix}$$

$$= F_2[V^T(t)F_1(W(t)X(t))] \tag{1}$$

where $x_j(t)$ (j = 1, 2, …, p) and y are the inputs and output of the BPNN; $w_{ij}(t)$, i = 0, …, q, j = 0, …, p, are the weights from the input layer to the hidden layer; $v_{ij}(t)$, i = 0, …, q, j = 1, …, k, are the weights from the hidden layer to the output layer; t is a time factor; f_1 is the activation function of the hidden nodes and f_2 is the activation function of the output nodes. Generally, the activation function for nonlinear nodes is assumed to be a symmetric hyperbolic tangent function, i.e., $f_1(x) = \tanh(u_0^{-1}x)$, and its derivative is $f_1'(x) = u_0^{-1}[1 - f_1^2(x)]$, $f_1''(x) = -2u_0^{-1}f_1(x)[1 - f_1^2(x)]$, where u_0 is the shape factor of the activation function. For later convenience, let $net_i(t) = \sum_{j=0}^{p} w_{ij}(t)x_j(t)$.

Usually, by estimating model parameter vector (W, V) via BPNN training, we can realize the corresponding tasks such as function approximation and pattern prediction. In fact, the model parameter vector (W, V) can be obtained by iteratively minimizing a cost function $E(X: W, V)$. In general, $E(X: W, V)$ is a sum of the error squares cost function with k output nodes and N training pairs or patterns, that is,

$$E(X:W,V) = \frac{1}{2}\sum_{j=1}^{N}\sum_{i=1}^{p} e_{ij}^2 = \frac{1}{2}\sum_{j=1}^{N} e_j^T e_j = \frac{1}{2}\sum_{j=1}^{N}[y_j - y_j(X:W,V)]^T[y_j - y_j(X:W,V)] \tag{2}$$

where y_j is the jth actual value and $y_j(X: W,V)$ is the jth estimated value.

Given the time factor t and direction factor d, Eq. (2) can be rewritten as

$$E_d(t) = \frac{1}{2}\sum_{j=1}^{N}\sum_{i=1}^{k} f_d \cdot e_{ij}^2(t) = \frac{1}{2}\sum_{j=1}^{N} f_d \cdot e_j^T(t)e_j(t) = \frac{1}{2}\sum_{j=1}^{N} f_d(y_j)\cdot[y_j - y_j(t)]^T[y_j - y_j(t)] \tag{3}$$

where $f_d(y_j)$ is a directional function, i.e., $f_d(y_j) = \begin{cases} p_1, [y_j(t) - y_j(t-1)][\hat{y}_j(t) - y_j(t-1)] \le 0, \\ p_2, \text{otherwise} \end{cases}$

Comparing with the traditional mean squared error, directional error function is preferred in the trend prediction. In the directional error function, incorrectly predicted directions are penalized more heavily than the correct prediction. In this study, p_1 = 5 and p_2 = 2.5 are used in the directional error function.

By applying the steepest descent method to the error cost function $E_d(t)$ (i.e., Eq. (3)), we can obtain the gradient of $E_d(t)$ with respect to V and W, respectively.

$$\nabla_V E_d(t) = \frac{\partial E_d(t)}{\partial V(t)} = \sum_{j=1}^{N}\sum_{i=1}^{k} f_d e_{ij}(t)\frac{\partial e_{ij}(t)}{\partial V(t)} = -\sum_{j=1}^{N}\sum_{i=1}^{k} f_d e_{ij}(t)\frac{\partial y_{ij}(t)}{\partial V(t)}$$
$$= -\sum_{j=1}^{N} f_d e_j(t) F'_{2(j)}[V^T F_{1(j)}(WX)]' = -\sum_{j=1}^{N} f_d F_{1(j)}(WX) e_j^T(t) F'_{2(j)} = -\sum_{j=1}^{N} f_d F_{1(j)} e_j^T(t) F'_{2(j)}$$

$$\nabla_W E_d(t) = \frac{\partial E_d(t)}{\partial W(t)} = \sum_{j=1}^{N}\sum_{i=1}^{k} f_d e_{ij}(t)\frac{\partial e_{ij}(t)}{\partial W(t)} = -\sum_{j=1}^{N}\sum_{i=1}^{k} f_d e_{ij}(t)\frac{\partial y_{ij}(t)}{\partial W(t)}$$
$$= -\sum_{j=1}^{N} f_d e_j(t) F'_{2(j)}[V^T F_{1(j)}(WX)]' = -\sum_{j=1}^{N} f_d F'_{1(j)} \overline{V} F'_{2(j)} e_j(t) x_j^T(t) = -\sum_{j=1}^{N} f_d \overline{F}'_{1(j)} \overline{V} F'_{2(j)} e_j x_j^T$$

So, the updated formulae of weights are given by, respectively

$$\Delta V = -\eta \nabla_V E_d(t) = \eta \sum_{j=1}^{N} f_d F_{1(j)} e_j^T F'_{2(j)} \tag{4}$$

$$\Delta W = -\eta \nabla_W E(t) = \eta \sum_{j=1}^{N} \overline{F}'_{1(j)} \overline{V} F'_{2(j)} e_j x_j^T \tag{5}$$

Where η is the learning rate;

$$\overline{F}'_{1(j)} = \mathrm{diag}[f'_{1(1)} \quad f'_{1(2)} \quad \cdots \quad f'_{1(q)}] \in R^{q\times q},\ f'_{1(i)} = f'_1(net_i) = \frac{\partial f_1(net_i)}{\partial net_i}, i = 1,2,\cdots,q,$$

$$F'_2 = diag[f'_{2(1)} \quad f'_{2(2)} \quad \cdots \quad f'_{2(k)}] \in R^{k\times k},\ f'_{2(i)} = f'_2[v_i^T F_1(WX)] = \frac{\partial f_2[v_i^T F_1(WX)]}{\partial [v_i^T F_1(WX)]}, i = 1,2,\cdots k,$$

$$\overline{V} = \begin{bmatrix} v_{11} & v_{21} & \cdots & v_{k1} \\ v_{12} & v_{22} & \cdots & v_{k2} \\ \cdots & \cdots & \cdots & \cdots \\ v_{1q} & v_{2q} & \cdots & v_{kq} \end{bmatrix} = [\overline{v}_1\ \overline{v}_2\ \cdots\ \overline{v}_k] \in R^{q\times p},\ \overline{v}_i = [v_{i1} \quad \cdots \quad v_{iq}]^T \in R^{q\times 1}, i = 1,2,\cdots,q.$$

To derive the optimal learning rate, let Δ be an increment operator and consider the general error equation:

$$\Delta e(t+1) = e(t+1) - e(t) = y - y(t+1) - y + y(t) = -\Delta y(t+1) \tag{6}$$

If $\Delta y(t+1) \equiv 0$, then this means there is no change in the pattern during the neural networks' learning procedure and the change of output of neural networks is

$$\Delta y(t+1) = \eta \xi(t) e(t) \tag{7}$$

where $\xi(t) = \mathbf{F}'_2[(\mathbf{F}_1^T \mathbf{F}_1) \otimes \mathbf{I}_{k^2} + (\mathbf{X}^T \mathbf{X}) \times (\overline{V}\mathbf{F}_1'\mathbf{F}_1'\overline{V})]\mathbf{F}'_2$ (Here $\otimes$ indicates a direct product, and $\times$ indicates a cross product) with

$$\mathbf{F}'_2 = \begin{bmatrix} F'_{2(1)} & 0 & \cdots & 0 \\ 0 & F'_{2(2)} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & F'_{2(N)} \end{bmatrix},\quad \mathbf{F}_1^T\mathbf{F}_1 = \begin{bmatrix} F_{1(1)}^T F_{1(1)} & F_{1(1)}^T F_{1(2)} & \cdots & F_{1(1)}^T F_{1(N)} \\ F_{1(2)}^T F_{1(1)} & F_{1(2)}^T F_{1(2)} & \cdots & F_{1(2)}^T F_{1(N)} \\ \cdots & \cdots & \cdots & \cdots \\ F_{1(N)}^T F_{1(1)} & F_{1(N)}^T F_{1(2)} & \cdots & F_{1(N)}^T F_{1(N)} \end{bmatrix},$$

$$\mathbf{X}^T\mathbf{X} = \begin{bmatrix} x_1^T x_1 & x_1^T x_2 & \cdots & x_1^T x_N \\ x_2^T x_1 & x_2^T x_2 & \cdots & x_2^T x_N \\ \cdots & \cdots & \cdots & \cdots \\ x_N^T x_1 & x_N^T x_2 & \cdots & x_N^T x_N \end{bmatrix},\quad \mathbf{F}_1\mathbf{F}_1' = \begin{bmatrix} \overline{F}'_{1(1)}\overline{F}'_{1(1)} & \overline{F}'_{1(1)}\overline{F}'_{1(2)} & \cdots & \overline{F}'_{1(1)}\overline{F}'_{1(N)} \\ \overline{F}'_{1(2)}\overline{F}'_{1(1)} & \overline{F}'_{1(2)}\overline{F}'_{1(2)} & \cdots & \overline{F}'_{1(2)}\overline{F}'_{1(N)} \\ \cdots & \cdots & \cdots & \cdots \\ \overline{F}'_{1(N)}\overline{F}'_{1(1)} & \overline{F}'_{1(N)}\overline{F}'_{1(2)} & \cdots & \overline{F}'_{1(N)}\overline{F}'_{1(N)} \end{bmatrix}.$$

In order to prove Eq. (7), a lemma must be introduced.

Lemma: The total time derivative of the BPNN single output $v^T F_1(WX)$ is given by

$$\frac{d[v^T F_1(WX)]}{dt} = F_1(WX)\frac{dv}{dt} + \overline{v}^T \overline{F}'_1(WX)\frac{dW}{dt}X = \frac{dv^T}{dt}F_1(WX) + \overline{v}^T \overline{F}'_1(WX)\frac{dW}{dt}X$$

Proof:
$$\frac{d[v^T F_1(WX)]}{dt} = \frac{d\left[\sum_{i=0}^{q} v_i f_1\left(\sum_{j=0}^{p} w_{ij} x_j\right)\right]}{dt}$$
$$= \sum_{i=0}^{q} \frac{\partial\left[\sum_{i=0}^{q} v_i f_1\left(\sum_{j=0}^{p} w_{ij} x_j\right)\right]}{\partial v_i}\frac{dv_i}{dt} + \sum_{i=0}^{q}\sum_{j=0}^{p} \frac{\partial\left[\sum_{i=0}^{q} v_i f_1\left(\sum_{j=0}^{p} w_{ij} x_j\right)\right]}{\partial w_{ij}}\frac{dw_{ij}}{dt}$$
$$= \sum_{i=0}^{q} f_1\left(\sum_{j=0}^{p} w_{ij}x_j\right)\frac{dv_i}{dt} + \sum_{i=0}^{q}\sum_{j=0}^{p} v_i f_1'\left(\sum_{j=0}^{p} w_{ij}x_j\right)x_j\frac{dw_{ij}}{dt} = f_1(net_0)\frac{dv_0}{dt} + f_1(net_1)\frac{dv_1}{dt} + \cdots + f_1(net_q)\frac{dv_q}{dt}$$
$$+ v_0 f_1'(net_0)\left[x_0\frac{dw_{00}}{dt} + x_1\frac{dw_{01}}{dt} + \cdots + x_p\frac{dw_{0p}}{dt}\right] + \cdots + v_q f_1'(net_q)\left[x_0\frac{dw_{q0}}{dt} + x_1\frac{dw_{q1}}{dt} + \cdots + x_p\frac{dw_{qp}}{dt}\right]$$
$$= \begin{bmatrix} f_1(net_0) & f_1(net_1) & \cdots & f_1(net_q)\end{bmatrix}\begin{bmatrix}\frac{dv_0}{dt} & \frac{dv_1}{dt} & \cdots & \frac{dv_q}{dt}\end{bmatrix}^T + \begin{bmatrix} v_1 & v_2 & \cdots & v_q\end{bmatrix}\begin{bmatrix} f'(net_1) & \cdots & 0 \\ \cdots & \cdots & \cdots \\ 0 & \cdots & f'(net_q)\end{bmatrix}$$
$$\times\begin{bmatrix}\frac{dw_{00}}{dt} & \frac{dw_{01}}{dt} & \cdots & \frac{dw_{0p}}{dt} \\ \frac{dw_{10}}{dt} & \frac{dw_{11}}{dt} & \cdots & \frac{dw_{1p}}{dt} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{dw_{q0}}{dt} & \frac{dw_{q1}}{dt} & \cdots & \frac{dw_{qp}}{dt}\end{bmatrix}\begin{bmatrix} x_0 \\ x_1 \\ \cdots \\ x_p\end{bmatrix} = F_1(WX)\frac{dv}{dt} + \bar{v}^T\overline{F_1'}(WX)\frac{dW}{dt}X = \frac{dv^T}{dt}F_1(WX) + \bar{v}^T\overline{F_1'}(WX)\frac{dW}{dt}X \quad \square$$

In the following, we start to prove Eq. (7) with above Lemma together with Eqs. (4) and (5). Now consider the change of output of BPNN for all patterns, that is

$$\Delta y(t+1) = \begin{bmatrix}\Delta y_1(t+1) \\ \Delta y_2(t+1) \\ \cdots \\ \Delta y_N(t+1)\end{bmatrix} = \begin{bmatrix}\Delta F_{2,1}(v^T F_1(WX)) \\ \Delta F_{2,2}(v^T F_1(WX)) \\ \cdots \\ \Delta F_{2,N}(v^T F_1(WX))\end{bmatrix} = \begin{bmatrix}\eta\sum_{j=1}^{N} F_{2,1}'\cdot(F_{1,1}^T F_{1j} I_{k^2} + \bar{V}^T F_{1,1}' F_{1j}'\bar{V}x_j^T x_1)\cdot F_{2j}' e_j \\ \eta\sum_{j=1}^{N} F_{2,2}'\cdot(F_{1,2}^T F_{1j} I_{k^2} + \bar{V}^T F_{1,2}' F_{1j}'\bar{V}x_j^T x_2)\cdot F_{2j}' e_j \\ \cdots \\ \eta\sum_{j=1}^{N} F_{2,N}'\cdot(F_{1,N}^T F_{1j} I_{k^2} + \bar{V}^T F_{1,N}' F_{1j}'\bar{V}x_j^T x_N)\cdot F_{2j}' e_j\end{bmatrix}$$
$$= \eta\begin{bmatrix} F_{2,1}' & 0 & \cdots & 0 \\ 0 & F_{2,2}' & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & F_{2,N}'\end{bmatrix}\times\begin{bmatrix}(F_{1,1}^T F_{11} I_{k^2} + \bar{V}^T F_{1,1}' F_{11}'\bar{V}x_1^T x_1) & (F_{1,1}^T F_{12} I_{k^2} + \bar{V}^T F_{1,1}' F_{12}'\bar{V}x_1^T x_2) & \cdots & (F_{1,1}^T F_{1N} I_{k^2} + \bar{V}^T F_{1,1}' F_{1N}'\bar{V}x_1^T x_N) \\ (F_{1,2}^T F_{11} I_{k^2} + \bar{V}^T F_{1,2}' F_{11}'\bar{V}x_2^T x_1) & (F_{1,2}^T F_{12} I_{k^2} + \bar{V}^T F_{1,2}' F_{12}'\bar{V}x_2^T x_2) & \cdots & (F_{1,2}^T F_{1N} I_{k^2} + \bar{V}^T F_{1,2}' F_{1N}'\bar{V}x_2^T x_N) \\ \cdots & \cdots & \cdots & \cdots \\ (F_{1,N}^T F_{11} I_{k^2} + \bar{V}^T F_{1,N}' F_{11}'\bar{V}x_N^T x_1) & (F_{1,N}^T F_{12} I_{k^2} + \bar{V}^T F_{1,N}' F_{12}'\bar{V}x_N^T x_2) & \cdots & (F_{1,N}^T F_{1N} I_{k^2} + \bar{V}^T F_{1,N}' F_{1N}'\bar{V}x_N^T x_N)\end{bmatrix}$$
$$\times\begin{bmatrix} F_{21}' e_1 \\ F_{22}' e_2 \\ \cdots \\ F_{2N}' e_N\end{bmatrix} = \eta\begin{bmatrix} F_{21}' & 0 & \cdots & 0 \\ 0 & F_{22}' & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & F_{2N}'\end{bmatrix}\left\{\begin{bmatrix} F_{11}^T F_{11} I_{k^2} & F_{11}^T F_{12} I_{k^2} & \cdots & F_{11}^T F_{1N} I_{k^2} \\ F_{12}^T F_{11} I_{k^2} & F_{12}^T F_{12} I_{k^2} & \cdots & F_{12}^T F_{1N} I_{k^2} \\ \cdots & \cdots & \cdots & \cdots \\ F_{1N}^T F_{11} I_{k^2} & F_{1N}^T F_{12} I_{k^2} & \cdots & F_{1N}^T F_{1N} I_{k^2}\end{bmatrix}\right.$$
$$\left. + \begin{bmatrix} x_1^T x_1 & x_1^T x_2 & \cdots & x_1^T x_N \\ x_2^T x_1 & x_2^T x_2 & \cdots & x_2^T x_N \\ \cdots & \cdots & \cdots & \cdots \\ x_N^T x_1 & x_N^T x_2 & \cdots & x_N^T x_N\end{bmatrix}\times\left(\bar{V}^T\begin{bmatrix}\overline{F_{11}'}\,\overline{F_{11}'} & \overline{F_{11}'}\,\overline{F_{12}'} & \cdots & \overline{F_{11}'}\,\overline{F_{1N}'} \\ \overline{F_{12}'}\,\overline{F_{11}'} & \overline{F_{12}'}\,\overline{F_{12}'} & \cdots & \overline{F_{12}'}\,\overline{F_{1N}'} \\ \cdots & \cdots & \cdots & \cdots \\ \overline{F_{1N}'}\,\overline{F_{11}'} & \overline{F_{1N}'}\,\overline{F_{12}'} & \cdots & \overline{F_{1N}'}\,\overline{F_{1N}'}\end{bmatrix}\bar{V}\right)\right\}\times\begin{bmatrix} F_{21}' & 0 & \cdots & 0 \\ 0 & F_{22}' & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & F_{2N}'\end{bmatrix}\begin{bmatrix} e_1 \\ e_2 \\ \cdots \\ e_N\end{bmatrix}$$
$$= \eta\mathbf{F}_2'[(\mathbf{F}_1^T\mathbf{F}_1)\otimes\mathbf{I}_{k^2} + (\mathbf{X}^T\mathbf{X})\times(\bar{V}\mathbf{F}_1'\mathbf{F}_1'\bar{V})]\mathbf{F}_2' e(t) = \eta\xi(t)e(t) \quad \square$$

Substituting (7) into (6), we obtain

$$e(t+1) = e(t) - \eta\xi(t)e(t) \tag{8}$$

The objective here is to derive an optimal learning rate η. That is, at iteration t, an optimal value of the learning rate, $\eta^*(t)$, which minimizes $E_d(t+1)$ is obtained, i.e.,

$$E_d(t+1) = 0.5 f_d[e^T(t+1)e(t+1)] \tag{9}$$

Using (8), Eq. (9) may be written as

$$E_d(t+1) = 0.5 f_d[e(t) - \eta\xi(t)e(t)]^T[e(t) - \eta\xi(t)e(t)] \tag{10}$$

which gives the error e, at iteration t+1, as a function of the learning rate η, which minimizes $E_d(t+1)$. Now we use the first and second order conditions

$$\left.\frac{dE_d(t+1)}{d\eta}\right|_{\eta=\eta^*(t)} = -\frac{1}{2} f_d[\xi(t)e(t)]^T[e(t) - \eta^*(t)\xi(t)e(t)] - \frac{1}{2} f_d[e(t) - \eta^*(t)\xi(t)e(t)]^T\xi(t)e(t) = 0$$

$$\left.\frac{d^2E_d(t+1)}{d\eta^2}\right|_{\eta=\eta^*(t)} = f_d[e^T(t)\xi^T(t)\xi(t)e(t)] > 0$$

Since $\xi(t)$ is positively definite, the second condition is met and the optimum value of the learning rate is found to be

$$\eta^*(t) = \frac{e^T(t)\xi^T(t)e(t)}{e^T(t)\xi^T(t)\xi(t)e(t)} \tag{11}$$

From the Eq. (11), the learning rate is not affected by the direction factor f_d. Finally, the increments of the BP neural network parameters, by using the optimal learning rate, are obtained by replacing the η^* given by (11) to (4)-(5), which yield

$$\Delta V = -\eta\nabla_V E_d(t) = \frac{e^T(t)\xi^T(t)e(t)}{e^T(t)\xi^T(t)\xi(t)e(t)} \sum_{j=1}^{N} f_d[F_{1(j)}' e_j^T F_{2(j)}'] \tag{12}$$

$$\Delta W = -\eta\nabla_W E_d(t) = \frac{e^T(t)\xi^T(t)e(t)}{e^T(t)\xi^T(t)\xi(t)e(t)} \sum_{j=1}^{N} f_d[\overline{F}_{1(j)}'\overline{V}F_{2(j)}' e_j x_j^T] \tag{13}$$

Using the new weight update formulae with optimal learning rates and direction error correction, a new BP learning algorithm is generated. For verification, one major foreign exchange rate (EUR/USD) is used in the following section.

3 Empirical Analysis

In the experiments, the daily data of EUR/USD is obtained from Pacific Exchange Rate Service (http://fx.sauder.ubc.ca/). The entire data set covers the period from January 1 2000 to December 31 2004. In this experiment, we use the first 4 years' data to train and the last year's data for testing. As the main goal is to predict the trend of foreign exchange market, then the directional statistics (D_{stat}) [5] is employed in this study. For comparison, the standard three-layer BPNN (BPNN I) is used as benchmark model. In addition, the BPNN with optimal learning rates (BPNN II) and the BPNN with direction error correction (BPNN III) are also compared. The proposed BPNN with optimal learning rates and direction error correction is called "BPNN IV". The experiment results are reported in Table 1.

From Table 1, we can find the following conclusions. (1) The best prediction performance for the testing data is generally produced when the training epochs are 1500. (2) Usually, too few training epochs and hidden nodes can not lead to a good forecasting result. (3) Of the four BPNN models, the proposed adaptive BPNN model with optimal learning rates and directional error correction performs the best, following

Table 1. The performance comparison for various BPNN models on EUR/USD

Training epochs	Hidden nodes	BPNN I	BPNN II	BPNN III	BPNN IV
1000	5	54.76	61.90	63.49	68.65
	10	58.33	66.67	69.84	72.22
	15	58.73	65.87	70.63	71.43
1500	5	57.94	63.49	70.23	70.63
	10	60.32	***69.44***	***75.40***	***77.78***
	15	***61.51***	67.85	72.22	75.00
3000	5	55.95	65.48	69.84	72.22
	10	59.92	68.25	67.46	73.02
	15	60.32	67.86	68.25	70.24

with the BPNN with directional error correction and the BPNN with optimal learning rates. The standard BPNN performs the worst. The main reason is that the directional error information seems to be more important than the optimal learning rates in the trend prediction. (4) These results also indicate the feasibility of the proposed adaptive BPNN model in foreign exchange market trend forecasting.

4 Conclusions

In this study, an adaptive BP learning algorithms with optimal learning rates and directional error correction is first proposed. And then this exploratory research examines the potential of using an adaptive BPNN model to predict an important foreign exchange rates — EUR/USD. Empirical investigation reveals that the proposed adaptive BP algorithm provides an alternative solution to the foreign exchange market trend prediction.

Acknowledgements

This work is partially supported by National Natural Science Foundation of China (NSFC No. 70221001); Chinese Academy of Sciences and Strategic Research Grant of City University of Hong Kong (SRG No. 7001677).

References

1. Widrow, B., Lehr, M.A.: 30 Years of Adaptive Neural Networks: Perception, Madaline, and Backprpagation. Proceedings of the IEEE Neural Networks I: Theory & Modeling (Special issue) 78 (1990) 1415-1442
2. Rumelhart, D.E., McClelland, J.L.: Parallel Distributed Processing. MIT Press, Cambridge, MA (1986)
3. Brent, R.P.: Fast Training Algorithms for Multilayer Neural Nets. IEEE Transactions on Neural Networks 2 (1991) 346-35
4. Hagan, M.T., Menhaj, M.: Training Feedforward Networks with Marquardt Algorithm. IEEE Transactions on Neural Networks 5 (1994) 989-993
5. Yu, L., Wang, S.Y., Lai, K.K.: A Novel Nonlinear Ensemble Forecasting Model Incorporating GLAR and ANN for Foreign Exchange Rates. Computers & Operations Research 32 (2005) 2523-2541

Recurrent Self-Organising Maps and Local Support Vector Machine Models for Exchange Rate Prediction

He Ni and Hujun Yin

School of Electrical and Electronic Engineering, University of Manchester, Manchester, UK
he.ni@manchester.ac.uk, hujun.yin@manchester.ac.uk

Abstract. This paper considers the problem of predicting non-linear, non-stationary financial time sequence data, which is often difficult for traditional regressive models. The Self-Organising Map (SOM) is a vector quantisation method that represents statistical data sets in a topology preserving fashion. The method, which uses the Recurrent Self-Organising Map(RSOM) to partition the original data space into several disjointed regions and then uses Support Vector Machines (SVMs) to make the prediction as a regression method. It is model free and does not require a prior knowledge of the data. Experiments show that the method can make certain degree of profits and outperforms the GARCH method.

1 Introduction

The problem of financial time series prediction is of primary importance to modern economical world. Financial time series forecasting is a challenging problem since most conventional statistical algorithms are not naturally suited for time varying patterns. Sudden changes in policy, natural and man-made disasters, institutional changes, new discoveries and revisions among others cause occasional large forecast errors in the standard, constant-parameter models. The traditional way, for example autoregressive moving average (ARMA) [1], to approach the problem is to model the underlying time series structures globally. In order to improve the quality of regression models in economic data modeling, autoregressive conditional heteroskedastic (ARCH) models were introduced by Engle [2] and extended by Bollerslev to GARCH model [3].

However, statistical models still suffer from being distorted by the non-stationary nature of most financial time series. If data series are divided into different regions, the noise level may vary from region to region. Therefore, over-fitting or under-fitting will inevitably exist in one single model. Alternatively, artificial neural networks have been proved with various success in modeling such non-stationary data [4]. In addition, inspired by a so-called divide-and-conquer principle, a discretisation function (clustering) and several local models that can be combined to solve a complicated problem [5, 6, 7]. Local models are then

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 504–511, 2006.

built up on the divided sub sets, which would be of relatively smaller size and smaller variation comparing to the global set.

The data partition is normally done by clustering methods, for example neural gas [8] and SOM [9]. Afterwards, the local models are set up by using the Support Vector Machines(SVMs) [10]. In this work, SVMs serve as regression methods to complete the prediction tasks.

The rest of the paper is organised as follows, Section 2 introduces the so-called divide-and-conquer architecture and the algorithm of tree type RSOM and SVM for regression task. Experimental results are provided in Section 3. Finally, a brief conclusion and future work is given.

2 Architecture and the Learning Algorithm

For time series prediction, one has to capture the dependency between the past and the future behaviour of the time sequence. A typical one step ahead prediction can be formulated as

$$x(n) = [\gamma_1 x(n-1) + \gamma_2 x(n-2) + \ldots + \gamma_i x(n-i)] \ . \tag{1}$$

$x(n)$ is the value of time series at time step n, where $n = 1, 2, \ldots, i$. During the learning process, provided the actual process can be described by such a linear model, the coefficients $\gamma_k, k = 1, 2, \ldots, i$ should gradually converge under the least square method. However, in realities, γ_k may not always converge [11].

Global models give the best results with stationary time series. Whilst local models can often overcome some problems of global models and have gained growing interests during the last decade. By using local models, the possibility to converge in each smaller local region is highly enhanced. Thereafter, predictions are then done on local models.

2.1 Tree Type Recurrent Self-Organising Map

The Self-Organising Map (SOM) is a vector quantisation method to map patterns from a high dimensional input space ϕ to a lower dimensional prototype space φ while preserving the topological relationships among the nodes [9]. The network consists of a lattice of nodes, and in this case it is a tree. Starting from a main stem, then it gradually grows to branches, then the secondary branches, etc. Such a tree structure has been used in document management [12]. Defined by its position r, each node of the tree has a reference weight w_r of the same dimension as the input. In order to preserve the topology during the training as well as keep the training processing from being trapped in local minima, a proper neighbourhood function h has to be used.

The training algorithm of the SOM is: randomly choose one input vector $x(n)$ from an input space ϕ and compare it with the reference vector w_r of the neuron r on the output space φ. The best matching unit v to the input is the one having its weight w_v closest to it. Reference vectors (weights) are randomly set at the beginning, and then are updated by

$$w_r(n+1) = w_r(n) + \alpha(n) h_{rv}(n)(x(n) - w_r(n)) \ . \tag{2}$$

where $0 < \alpha(n) < 1$ is the learning rate, as n increases, $\alpha(n)$ decreases, with the constraints $\lim_{n\to\infty} \int_0^n \alpha(n)dn = \infty, \lim_{n\to\infty} \int_0^n \alpha(n)^2 dn < \infty$.

In order to store the temporal context in the time series, Chappell and Tayor [13] proposed a modification to the original SOM. This modified SOM, termed Temporal Kohonen Map (TKM), can not only function like a normal SOM but is also able to give context to vectors appearing in sequences. The main different between the TKM and the SOM is that the outputs of SOM are reset to zero after feeding of each input vector, while in the TKM the original outputs are replaced by leaky integrator outputs, which once activated, gradually lose their activity. The original TKM was then extended by Koskela, et al. [14] to a modified version called Recurrent Self-Organising Map(RSOM). RSOM moves the leaky integrator from the output into the input gives,

$$y_i(n) = (1-\beta)y_i(n-1) + \beta(x(n) - w_i(n)) \ . \tag{3}$$

$y_i(n)$ is the temporally leaked difference vector at map unit i, time n. Above $0 < \beta < 1$ is the leaking coefficient while $x(n)$ and $w_i(n)$ have their previous meanings. Large β corresponds to short memory while small β corresponds to longer memory. When $\beta = 1$, RSOM behaves like a normal SOM and on the other extreme when $\beta = 0$ all $w_i(n)$ equal to the mean of input vectors. After certain mathematical reformulation, $y_(n)$ end up with a similar form as an exponentially weighted linear FIR filter,

$$y(n) = \beta \sum_{k=1}^{n} (1-\beta)^{(n-k)}(x(k) - w) \ . \tag{4}$$

The updating rule of the map should include part of the history of the input, as the impulse response of the recurrent filter decay below 5% of its initial value after a few steps. For further explanation, please see [14]. The length of episode is 3 and $\beta = 0.95$ in the experiments.

The tree type RSOM has the similar mechanism as the Neural Gas [8]. Firstly the map starts with a main stem, an one dimensional RSOM chain. After certain iterations, the stem is trained to go through the backbone of the "density". According to the number of input vectors attracted by each neuron, the main stem starts to split into sub-chains at the neurons that attract more input vectors than a preset threshold. Also the length of sub-chain depends on how much the number of input vectors.

2.2 Support Vector Machine in Regression

SVM [10] is a novel regressor with three distinct advantages. First, SVM solves a risk minimisation problem by balancing the empirical error and a regularisation term, where the risk is measured by Vapnik's ε-insensitive loss function. Second, SVM usually estimates a set of linear functions defined in a high dimensional feature space. Third, SVM only uses a limited number of the support vectors.

Suppose we are given a training data set $\{(x1, y1), \ldots, (x_l, y_l)\} \subset \chi \times \mathbb{R}$, where χ denotes the space of the input patterns (e.g. $\chi = \mathbb{R}^d$). We try to do the prediction by estimating a linear function f

$$f(x) = w * \phi(x) + b \ . \tag{5}$$

$\phi(x)$ represents a nonlinear mapping of x in a high-dimensional feature space. The function f can be estimated by minimising a regularised risk function

$$\text{minimise} \quad \frac{1}{2} \parallel w \parallel^2 + C\frac{1}{l}\sum_{i=1}^{l} L_\varepsilon \ . \tag{6}$$

$$L_\varepsilon = \begin{cases} |y_i - w * \phi(x_i) - b| - \varepsilon, & |y_i - w * \phi(x_i) - b| \geq \varepsilon \\ 0, & \text{otherwise} \end{cases} \ . \tag{7}$$

The term $\parallel w \parallel^2$ is called the regularised term, which we want to make as flat as possible. The second term is the empirical error measured by Vapnik's ε-insensitive loss function. C is the regularisation constant. Transfer above two equations to the primal objective function as follows by introducing slack variables ξ_i^*, ξ_i

$$\text{minimise} \quad \frac{1}{2} \parallel w \parallel^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \ . \tag{8}$$

$$\text{subject to} \quad \begin{cases} y_i - w * \phi(x) - b \leq \varepsilon + \xi_i \\ w * \phi(x) + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \ \xi_i^* \geq 0 \end{cases} \ . \tag{9}$$

Finally, by introducing Lagrange multipliers α_i, α_i^*, we can solve a linearly constrained quadratic problem. The decision function then has a explicit form

$$f(x) = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)K(x_i, x) + b \ . \tag{10}$$

where $K(x_i, x)$ is defined as the kernel function. The elegance of using the kernel function is that one can deal with arbitrary dimensional feature space without having to compute the nonlinear mapping function explicitly. Any function that satisfies Mercer's condition [10] can be used as the kernel function. This technique has also been implicitly used in SOMs, as the neighbourhood functions are equivalent to Gaussian kernel functions [15]. Since no prior knowledge of the data distribution is available, we use Gaussian kernel in SVM regressor.

Based on Karush-Kuhn-Tucker(KKT) conditions [16], only a number of coefficients $(\alpha_i - \alpha_i^*)$ will be nonzero, which corresponding to training data points referred to as support vectors. Approximation errors at the point of support vectors are larger than or equal to ε.

2.3 Two-Stages Architecture

The idea underlying the two-stage architecture is to firstly partition the whole input space into several non-overlapped regions and then use regression models to produce forecasting results on those partitioned regions. These regression models

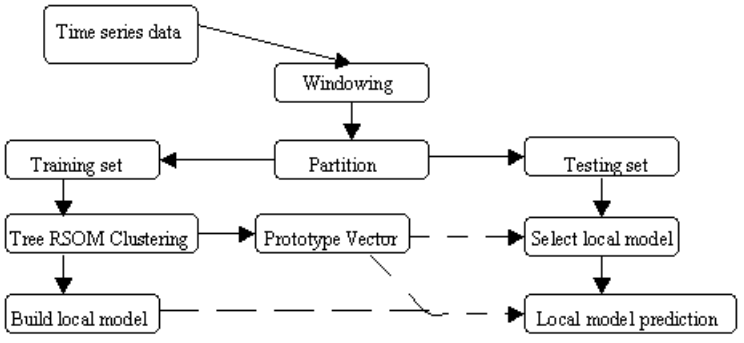


Fig. 1. A generalised scheme for constructing local models and their predictions

are called local models. A generalised scheme for constructing and evaluation the local models is depicted in Fig. 1.

The learning algorithm for the proposed two-stage architecture is outlined as follows:

1. Window the time series data into vectors. Separate into training, validation, testing parts.
2. Create a one dimensional chain with a number of terminal nodes and initialise it with random weights.
3. Recursively [1] training the Recurrent SOMs with the training data. Use the trained weight to partition input space into different regions.
4. Calculate the number of training data points in each regions. If the number of data points large than the preset threshold, expand the terminal node to a branch(a sub-RSOM chain) with at least two nodes on it. Otherwise, stop.
5. Repeat (3)-(4) until the partition cannot be preceded in any region.
6. Train SVM for the partitioned regions and find out the most adequate SVM based on the training data points in each partitioned region.

A validation set is used to early stop the RSOM in step (3) when the quantisation error of validation data starts to increase.

3 Experimental Results

We evaluate the proposed approach using real market data (exchange rate of U.S. Dollar to British Pound). The data is retrieved from the PACIFIC Exchange Rate Service provided by Prof. Werner Antwiler at UBCs Sauder School of Business. There are around 15 years' daily data excluding weekends and bank

[1] Only the initial weights are set randomly. The final weights of the first training phrase is brought to the second training phrase as its initial weights. Same as the learning rate, it is a continues variable rather than reset at the each time as a new submap is created.

holidays when currency markets were closed. We applied the price-return convert(i.e. $x'_n = \ln \frac{x_{n+1}}{x_n}$ here the x_n is the scalar values of the original data at the time n) to the original data. Considering the highly varying nature of the exchange rate, we train the neural networks with 3000 consecutive data points and we test the performance of the prediction method on the following 100 data points. Then, we window both of the training and testing sets with a window length of 13 to form two vector spaces. The window length is chosen empirically according to relatively good performances in SVM prediction.

We compare the various modeling and prediction methods on three measurements. The first is the Mean Square Error between target time series and predicted ones. The second measurement is the correct prediction percentage, which is a criterion to check whether the prediction is made on the right direction (i.e. we calculate how many percents predicted returns have the same sign as their corresponding actual returns). The third measurement is called profit earned, which is a program based on a virtual real trader's reaction in a real market. A set of rules have to be pre-set, such as if the increasing percentage is larger than a threshold, we invest certain amount of money to buy. If either increase or decrease is less than another threshold, no action is taken.

Fig. 2 shows the results of predictions of difference methods. The first sub-figure shows the prediction of the proposed local model method combining Recurrent SOM and SVM. The second sub-figure shows the local model method

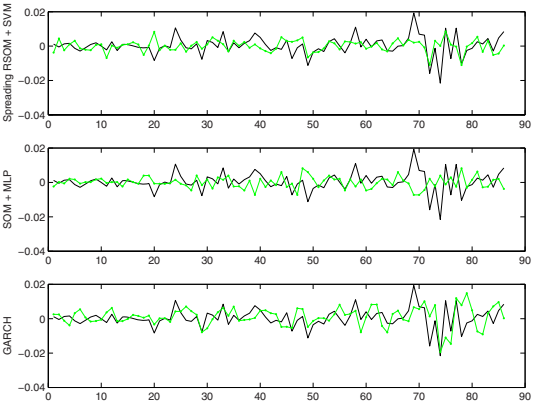


Fig. 2. The prediction results of three different approaches, the solid lines represent the original time series and the dot lines represent the predicted curves

Table 1. Total errors of various methods on the test set

	RSOM+SVM	SOM+MLP	GARCH
Mean Square Error (e-005)	3.0886	3.0999	3.9857
Correct Prediction (%)	67.5275	62.7907	46.6667
Profit earned (%)	7.20	4.29	5.56

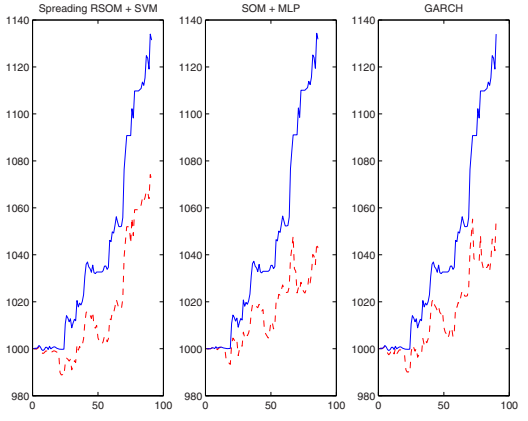


Fig. 3. The profits made based on three approaches. The closer the dash line is to the solid line, the better the performance. The solid line is the best profit curve, which means in case that prediction is 100% correct, the best performance a trader can achieve based on the strategy. The dash lines are the profits actually achieved.

combing SOM and MLP [2], and the third shows GARCH model prediction. Fig. 3 shows the profits made on these predictions of different approaches. Table 1 shows the measurements of three methods. The results of the first two methods are average of several experiments, because of the random initially weights and randomly feed in input vectors sequence.

4 Conclusion

Based on the principal of "divide-and-conquer", the proposed two-stage hybrid system outperforms other traditionary modeling techniques. There are a few inherent advantages of this approach. Firstly, since the prediction is based on smaller sets of data points in local models, it is thus possible to choose the best SVM for each individual model. Secondly, SOM is inherently an efficient clustering technique as it has a neighbourhood structure which can prevent the clustering being trapped to local minima. Thirdly, the convergence speed of SVMs can be greatly increased as the number of support vector decreases in smaller data regions (local models).

The future work includes alternative validation methods such as entropy [12] and nongaussian measurement [17] which can be used as a self-validation method. In this paper, we use Gaussian kernels as we know little about the distribution of exchange rate data. We will try to use more appropriate kernels if we can explore more on the data distribution in the future. Furthermore, we will also extend this work to multi-step prediction and consider bid-offer margins in profit estimation.

[2] The MLP has linear activation function and a number of inputs equals to the window size of the RSOM with one output.

References

1. Hamilton, J.: Time Series Analysis. Princeton University Press (1994)
2. Engle, R.: Autoregressive Conditional Heteroskedasticity with Estimates of The Variance of United Kingdom Inflation. Econometrics 50 (1982) 987-1007
3. Bollerslev, T.: Generalized Autoregressive Conditional Heteroskedasticity. Journal of Econometrics 31 (1986) 307-327
4. Kim, T.Y.,Oh, K.J., Kim, C., Do, J.D.: Artificial Neural Networks for Non-stationary Time Series. Neurocomputing 61 (2004) 439-447
5. Vesanto, J.: Using The SOM and Local Models in Time-Series Prediction. Helsinki University of Technology. (1997) B15
6. Cao, L.J.: Support Vector Machines Experts for Time Series Forecasting. Neurocomputing 51 (2002) 321-339
7. Koskela, T.: Time Series Prediction Using Recurrent SOM with Local Linear Models. Helsinki University of Technology (2001)
8. Martinetz, T.M., Berkovich, S.G., Schulten, K.J.: "Neural-Gas" Network for Vector Quantization and Its Application to Time Series Prediction. IEEE Transactions on Neural Networks 4 (1993) 558-569
9. Kohonen, T.: Self-Organizing Maps. Springer, Berlin (1995)
10. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
11. Walter, J., Ritter, H., Schulten, K.: Nonlinear Prediction with Self-Organizing Maps. Beckman-Institute and Department of Physics, UIUC, Urbana.
12. Freeman, R.T., Yin, H.: Adaptive Topological Tree Structure for Document Organisation and Visualisation. Neural Networks 17 (2004) 1255-1271
13. Chappell, G.J., Taylor, J.G.: The Temporal Kohonen Map. Neural Networks 4 (1993) 441-445
14. Koskela, T., Varsta, M., Heikkonen, J., Kaski, K.: Time series Prediciton Using Recurrent SOM with Local Linear Model. Report B15, Lab. of Computational Engineering, Helsinki University of Technology, Oct (1997)
15. Yin, H.: Self-Organising Map as A Natural Kernel Method. Proc. ICNN&B'05, Beijing, China, 13-15 Oct. (2005)
16. Kuhn, H.W., Tucker, A.W.: Nonlinear Programming. Proc. 2nd Berkeley Symposium on Mathematical Statistics and Probabilistics. 481C492, (1951) University of California Press, Berkeley
17. Hyvarinen, A., Oja, E.: Independent Component Analysis: Algorithms and Applications. Neural Networks 13(4-5) (2000) 411-430

Selection of the Appropriate Lag Structure of Foreign Exchange Rates Forecasting Based on Autocorrelation Coefficient

Wei Huang[1,2], Shouyang Wang[2], Hui Zhang[3,4], and Renbin Xiao[1]

[1] School of Management, Huazhong University of Science and Technology, Wuhan, 430074, China
rbxiao@163.com
[2] Institute of Systems Science, Academy of Mathematics and Systems Sciences, Chinese Academy of Sciences, Beijing, 100080, China
{whuang, sywang}@amss.ac.cn
[3] School of Knowledge Science, Japan Advanced Institute of Science and Technology, Asahidai 1-1, Tatsunokuchi, Ishikawa, 923-1292, Japan
zhang-h@jaist.ac.jp
[4] School of Computer Science, Southwest University of Science and Technology, Mianyang, Sichuan, 621002, China

Abstract. We propose a new criterion, called autocorrelation coefficient criterion (ACC) to select the appropriate lag structure of foreign exchange rates forecasting with neural networks, and design the corresponding algorithm. The criterion and algorithm are data-driven in that there is no prior assumption about the models for time series under study. We conduct the experiments to compare the prediction performance of the neural networks based on the different lag structures by using the different criterions. The experiment results show that ACC performs best in selecting the appropriate lag structure for foreign exchange rates forecasting with neural networks.

1 Introduction

Financial time series forecasting is one of the most challenging problems among the many applications of neural networks. Several modeling issues have been discussed [1-7]. In many situations, we have to predict the future value of foreign exchange rates through past measurements of it in the following way:

$$\hat{y}_{t+n} = F\left(y_{t-s_1}, y_{t-s_2}, \ldots, y_{t-s_i}, \ldots\right) \tag{1}$$

where $\hat{y}_{t+n}$ is the predicted value when we make a prediction of n periods ahead from the present period t; y_j is the actual value at period j; s_i is the lag period from the present period t; $F(\bullet)$ is a nonlinear function determined by the neural networks. The problem is to select the appropriate lag structure $\{s_1, s_2, ..., s_i, ...\}$.

In previous work, the criteria of selecting the lag structure include Akaike Information Criterion (AIC), Hannan-Quinn Criterion (HQC), Bayesian Information Criterion

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 512 – 517, 2006.

(BIC), Schwarz Information Criterion (SIC), the general-to-specific sequential Likelihood Ratio test (LR) and the specific-to-general sequential Portmanteau test. For monthly VAR models, AIC produces the most accurate structural and semi-structural impulse response estimates for realistic sample sizes. For quarterly VAR models, HQC appears to be the most accurate criterion with the exception of sample sizes smaller than 120, for which SIC is more accurate. For persistence profiles based on quarterly vector error correction (VEC) models, the SIC is the most accurate criterion for all realistic sample sizes. Sequential Lagrange-multiplier and likelihood ratio tests cannot be recommended [8]. AIC, SIC, HQC have an important role to play in model selection for nonlinear threshold models, while Generalized Information criterion (GIC) and especially Informational Complexity Criterion (ICOMP) prove less reliable [9]. A class of Modified Information Criteria (MIC) was considered which account for the fact that the bias in the sum of the autoregressive coefficients is highly dependent on the lag order [10]. For symmetric lag and asymmetric lag models, the results are more ambiguous in terms of forecasting performance in comparisons of AIC, SIC, Posterior Information Criterion (PIC), and Keating's modification of the AIC and SIC (KAIC and KSIC, respectively) [11]. It is thus difficult to generalize about the preferred lag selection technique when there is uncertainty about the true lag length and whether the lags are symmetric or asymmetric. AIC and BIC as well as several extensions have been used as information-based in-sample model selection criteria in selecting neural networks for foreign exchange rates forecasting [12]. However, the in-sample model selection criteria are not able to provide a reliable guide to out-of-sample performance and there is no apparent connection between in-sample model fit and out-of-sample forecasting performance. A nonparametric version of the Final Prediction Error (FPE) was analyzed for lag selection in nonlinear autoregressive time series under relative general conditions including heteroskedasticity [13]. There are assumptions that can't necessarily be satisfied in most cases. It requires suitable kernel and bandwidth choices to compute the lag selection criteria.

Our contribution is to propose a new criterion of selecting the appropriate lag structure of foreign exchange rates forecasting with neural networks and design the corresponding algorithm. The remainder of this paper is organized as follows. Section 2 describes the new criterion and algorithm. In Section 3, we conduct the experiments to compare the prediction performance of the neural networks based on the different lag structures by using the different criterions. Finally, conclusions are given in Section 4.

2 Autocorrelation Coefficient Criterion and Algorithm

In foreign exchange rates forecasting with neural networks, the contribution of one input variable to the output variable is affected by the other input variable. The input variables should be as predictive as possible. On the other hand, the input variables should not be much correlated, because the correlated input variables may degrade the prediction performance by interacting with each other and producing a biased effect [14]. Actually, the correlated input variables contribute the similar information for the output variable of neural networks. Therefore, the neural networks get confused and do not know to use which one. Considering the above features of foreign exchange rates forecasting with neural networks, we propose a new criterion of selecting the lag

structure of foreign exchange rates forecasting, called autocorrelation coefficient criterion (ACC) as follows:

(1) The absolute value of autocorrelation coefficient between the lag period and the forecasting period ahead should be as large as possible. In this case, the input variable is more correlated to the output variable. Therefore, the input variable contributes more predictive information for the output variable.

(2) The sum of absolute value of autocorrelation coefficients between the lag period and the other selected lag periods should be as small as possible. In this case, the input variable will not be much correlated to the other selected input variables.

One of the main features of foreign exchange rates is that as the lag period becomes long, the absolute value of autocorrelation coefficient will becomes small. Accordingly, we design an algorithm to select the lag structure of foreign exchange rates forecasting with neural networks as follows:

Step 1. Let $i=1$, $s_1=0$; and set the maximum lag period N, the forecasting period ahead n.

Step 2. Let $s_{i+1} = \arg_{s_i<k\le N} Max\left(\frac{|r_{k+n}|}{\sum_{j=1}^{i} |r_{k-s_j}|} \right)$, and $i = i+1$.

Step 3. If s_i less than $N-1$, go to Step 2; otherwise, exit and the appropriate lag structure is $\{s_1, s_2, ..., s_i\}$.

In order to further compare the effect of the different lag structures on the prediction performance, we add another two criterions of selecting the lag periods of foreign exchange rates forecasting, namely short criterion (SC) and long criterion (LC) as follows:

(1) Short criterion (SC)
If the lag structure based on ACC is $\{s_1, s_2, ..., s_m\}$, the lag structure based on SC is $\{0,1,..., m-1\}$.

(2) Long criterion (LC)
If the initial parameter maximum lag period of ACC is N, the lag structure based on LC is $\{0,1,..., N\}$.

3 Experiments Analysis

We conduct the experiments to compare the prediction performance of the neural networks based on the different lag structures by using the different criterions.

3.1 Neural Network Model

We employ the popular three layers back-propagation network with adaptive learning rate and momentum. The logistic function is used for all hidden nodes as the activation function. The linear activation function is employed for the output node. For time

series forecasting problem, the network inputs are the past, lagged observations of the data and the output is the future value. Therefore, the number of input nodes corresponds to the number of past lagged data. Generally speaking, too many nodes in the hidden layer produce a network that memorizes the input data and lacks the ability to generalize. Parsimony is a principle for designing neural networks. Hence, the number of hidden nodes is equal to the number of input nodes.

3.2 Performance Measure

Normalized mean squared error (NMSE) is used to evaluate the prediction performance of neural networks. Given a set P comprising pairs of the actual value x_k and predicted value $\hat{x}_k$, the NMSE can be defined as follows:

$$\text{NMSE} = \frac{\sum_{k \in P} (x_k - \hat{x}_k)^2}{\sum_{k \in P} (x_k - \bar{x}_k)^2} \tag{2}$$

where $\bar{x}_k$ is the mean of actual values.

3.3 Data Preparation

From Pacific Exchange Rate Service provided by Professor Werner Antweiler, University of British Columbia, Canada, we obtain 1005 daily observations covering the period from 2001 to 2004 for the three exchange rates respectively. For each exchange rate, we select the appropriate size of training set by using the method in [15]. The test set contains 209 daily observations covering the period from Jan 2005 to Oct, 2005 for the three exchange rates respectively.

3.4 Results

We set the initial parameters of ACC: (1) the maximum lag period $N = 10$; (2) the forecasting period ahead $n = 1$. Table 1 shows the lag structures of the three exchange rates forecasting by using the different criterions. Table 2 shows the three exchange rates prediction performance of the neural networks based on the different lag structures by using the different criterions. The value of NMSE under ACC is the

Table 1. The lag structures of the three exchange rates forecasting by using the different criterions (EUR, GBP and JPY against USD)

Criterion of selecting lag structure	Lag structure of EUR	Lag structure of GBP	Lag structure of JPY
ACC	{0, 4, 6, 9}	{0, 4, 6, 9}	{0, 4, 6, 9}
AIC	{0, 1, 6, 8}	{0, 1, 6, 8}	{0, 1, 6, 8}
BIC	{0}	{0}	{0}
HQC	{0, 1, 6, 8}	{0, 1, 6, 8}	{0, 1, 6, 8}
SC	{0, 1, 2, 3}	{0, 1, 2, 3}	{0, 1, 2, 3}
LC	{0, 1, …, 10}	{0, 1, …, 10}	{0, 1, …, 10}

Table 2. The three exchange rates prediction performance of the neural networks based on the different lag structures by using the different criterions (EUR, GBP and JPY against USD)

Criterion of selecting lag structure	NMSE of EUR	NMSE of GBP	NMSE of JPY
ACC	0.0861	0.0859	0.0852
AIC	0.1095	0.1092	0.1084
BIC	0.0994	0.0986	0.0962
HQC	0.1095	0.1092	0.1084
SC	0.1191	0.1187	0.1151
LC	0.3731	0.3728	0.3692

smallest among the different criterions in each exchange rates forecasting. It shows that ACC performs best in selecting the appropriate lag structure. ACC doesn't require any assumptions, completely independent of particular class of model. The criterion makes full uses of information among sample observations even if the underlying relationships are unknown or hard to describe. It integrated the features of autocorrelation coefficient of foreign exchange rates and the requirements of inputs for neural networks.

4 Conclusions

In this paper, we propose a new criterion, called autocorrelation coefficient criterion (ACC) to select the lag structure of foreign exchange rates forecasting with neural networks, and design the corresponding algorithm. The criterion and algorithm are data-driven in that there is no prior assumption about the models for time series under study. We conduct the experiments to compare the prediction performance of the neural networks based on the different lag structures by using the different criterions. The experiment results show that ACC outperforms the other criterions in selecting the lag structure for foreign exchange rates forecasting with neural networks.

Acknowledgements

This work is partially supported by National Natural Science Foundation of China (NSFC No. 70221001) and the Key Research Institute of Humanities and Social Sciences in Hubei Province-Research Center of Modern Information Management.

References

1. Lai, K.K., Yu, L.A., Wang, S.Y.: A Neural Network and Web-Based Decision Support System for Forex Forecasting and Trading. Lecture Notes in Artificial Intelligence, Vol. 3327. Springer-Verlag Berlin Heidelberg (2004) 243–253
2. Huang, W., Lai, K.K., Nakamori, Y., Wang, S.Y.: Forecasting Foreign Exchange Rates with Artificial Neural Networks: a Review. International Journal of Information Technology & Decision Making, **3**(1) (2004) 145-165

3. Zhang, H, Ho, T. B., Huang, W.: Blind Feature Extraction for Time-Series Classification Using Haar Wavelet Transform. Lecture Notes in Computer Science, Vol. 3497. Springer-Verlag Berlin Heidelberg (2005) 605–610
4. Yu, L.A., Wang, S.Y., Lai, K.K.: Adaptive Smoothing Neural Networks in Foreign Exchange Rate Forecasting. Lecture Notes in Computer Science, Vol. 3516. Springer-Verlag Berlin Heidelberg (2005) 523 – 530
5. Yu, L.A., Wang, S.Y., Lai, K.K.: A Novel Nonlinear Ensemble Forecasting Model Incorporating GLAR and ANN for Foreign Exchange Rates. Computers & Operations Research, **32** (2005) 2523–2541
6. Yu, L.A., Lai, K.K., Wang, S.Y.: Double Robustness Analysis for Determining Optimal Feedforward Neural Network Architecture. Lecture Notes in Computer Science, Vol. 3610. Springer-Verlag Berlin Heidelberg (2005) 382 – 385
7. Huang, W., Nakamori, Y., Wang, S.Y.: Forecasting Stock Market Movement Direction with Support Vector Machine. Computers & Operations Research, **32** (2005) 2513-2522
8. Ivanov, V., Kilian, L.: A Practitioner's Guide to Lag-Order Selection for Vector Autoregressions. Working paper, Centre for Economic Policy Research, (2000) 90-98
9. Kapetanios, G.: Model Selection in Threshold Models. Journal of Time Series Analysis, **22** (2001) 733-754
10. Ng, S., Perron, P.: Lag Length Selection and the Construction of Unit Root Tests with Good Size and Power. Boston College Working Papers in Economics 369, Boston College Department of Economics (2000)
11. Ozcicek, O., Mcmillian, W.D.: Lag Length Selection in Vector Autoregressive Models: Symmetric and Asymmetric Lags. Applied Economics, **31** (1999) 517- 524
12. Qi, M., Zhang, G.P.: An Investigation of Model Selection Criteria for Neural Network Time Series Forecasting. European Journal of Operational Research, **132** (2001) 666-680
13. Tschernig, R., Yang, L.: Nonparametric Lag Selection for Time Series. Journal of Time Series Analysism, **21** (2000) 457-487
14. Zhang, G.P.: Neural Networks in Business Forecasting. Idea Group Inc., (2003)
15. Huang, W., Nakamori, Y., Wang, S.Y., Zhang, H.: Select the Size of Training Set for Financial Forecasting with Neural Networks. Lecture Notes in Computer Science, Vol. 3497. Springer-Verlag Berlin Heidelberg (2005) 879–884

Exchange Rate Forecasting Using Flexible Neural Trees

Yuehui Chen[1], Lizhi Peng[1], and Ajith Abraham[1,2]

[1] School of Information Science and Engineering, Jinan University, Jinan 250022, P.R. China
yhchen@ujn.edu.cn
[2] IITA Professorship Program, School of Computer Science and Engg., Chung-Ang University, Seoul, Republic of Korea
ajith.abraham@ieee.org

Abstract. Forecasting exchange rates is an important financial problem that is receiving increasing attention especially because of its difficulty and practical applications. This paper proposes a Flexible Neural Tree (FNT) model for forecasting three major international currency exchange rates. Based on the pre-defined instruction sets, a flexible neural tree model can be created and evolved. This framework allows input variables selection, over-layer connections and different activation functions for the various nodes involved. The FNT structure is developed using the Extended Compact Genetic Programming and the free parameters embedded in the neural tree are optimized by particle swarm optimization algorithm. Empirical results indicate that the proposed method is better than the conventional neural network forecasting models.

1 Introduction

Forecasting exchange rates is an important financial problem that is receiving increasing attention especially because of its difficulty and practical applications. Exchange rates are affected by many highly correlated economic, political and even psychological factors. These factors interact in a very complex fashion. Exchange rate series exhibit high volatility, complexity and noise that result from an elusive market mechanism generating daily observations [1].

Much research effort has been devoted to exploring the nonlinearity of exchange rate data and to developing specific nonlinear models to improve exchange rate forecasting, i.e., the autoregressive random variance (ARV) model [2], autoregressive conditional heteroscedasticity (ARCH) [3], self-exciting threshold autoregressive models [4]. There has been growing interest in the adoption of neural networks, fuzzy inference systems and statistical approaches for exchange rate forecasting problem [5][6][15][8]. For a recent review of neural networks based exchange rate forecasting, please consult [9].

The input dimension (i.e. the number of delayed values for prediction) and the time delay (i.e. the time interval between two time series data) are two critical factors that affect the performance of neural networks. The selection of dimension and time delay has great significance in time series prediction.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 518–523, 2006.

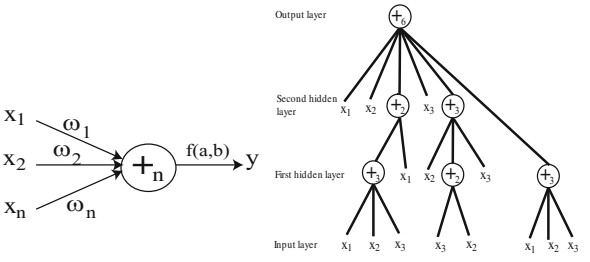


Fig. 1. A flexible neuron operator (left), and a typical representation of the FNT with function instruction set $F = \{+_2, +_3, +_4, +_5, +_6\}$, and terminal instruction set $T = \{x_1, x_2, x_3\}$ (right)

This papers proposes a Flexible Neural Tree (FNT) [10][11] for selecting the input variables and forecasting exchange rates. Based on the pre-defined instruction/operator sets, a flexible neural tree model can be created and evolved. FNT allows input variables selection, over-layer connections and different activation functions for different nodes. In our previous work, the hierarchical structure was evolved using Probabilistic Incremental Program Evolution algorithm (PIPE) with specific instructions. In this research, the hierarchical structure is evolved using the Extended Compact Genetic Programming (ECGP), a tree-structure based evolutionary algorithm. The fine tuning of the parameters encoded in the structure is accomplished using particle swarm optimization (PSO). The novelty of this paper is in the usage of flexible neural tree model for selecting the important inputs and/or time delays and for forecasting foreign exchange rates.

2 The Flexible Neural Tree Model

The function set F and terminal instruction set T used for generating a FNT model are described as $S = F \bigcup T = \{+_2, +_3, \ldots, +_N\} \bigcup \{x_1, \ldots, x_n\}$, where $+_i (i = 2, 3, \ldots, N)$ denote non-leaf nodes' instructions and taking i arguments. x_1,x_2,...,x_n are leaf nodes' instructions and taking no other arguments. The output of a non-leaf node is calculated as a flexible neuron model (see Fig.1). From this point of view, the instruction $+_i$ is also called a flexible neuron operator with i inputs. In the creation process of neural tree, if a nonterminal instruction, i.e., $+_i (i = 2, 3, 4, \ldots, N)$ is selected, i real values are randomly generated and used for representing the connection strength between the node $+_i$ and its children. In addition, two adjustable parameters a_i and b_i are randomly created as flexible activation function parameters. For developing the forecasting model, the flexible activation function $f(a_i, b_i, x) = e^{-(\frac{x-a_i}{b_i})^2}$ is used. The total excitation of $+_n$ is $net_n = \sum_{j=1}^{n} w_j * x_j$, where $x_j (j = 1, 2, \ldots, n)$ are the inputs to node $+_n$. The output of the node $+_n$ is then calculated by $out_n = f(a_n, b_n, net_n) = e^{-(\frac{net_n - a_n}{b_n})^2}$. The overall output of flexible neural tree can be computed from left to right by depth-first method, recursively.

Tree Structure Optimization. Finding an optimal or near-optimal neural tree is formulated as a product of evolution. In our previously studies, the Genetic Programming (GP), Probabilistic Incremental Program Evolution (PIPE) have been explored for structure optimization of the FNT [10][11]. In this paper, the Extended Compact Genetic Programming (ECGP) [13] is employed to find an optimal or near-optimal FNT structure.

ECGP is a direct extension of ECGA to the tree representation which is based on the PIPE prototype tree. In ECGA, Marginal Product Models (MPMs) are used to model the interaction among genes, represented as random variables, given a population of Genetic Algorithm individuals. MPMs are represented as measures of marginal distributions on partitions of random variables. ECGP is based on the PIPE prototype tree, and thus each node in the prototype tree is a random variable. ECGP decomposes or partitions the prototype tree into sub-trees, and the MPM factorises the joint probability of all nodes of the prototype tree, to a product of marginal distributions on a partition of its sub-trees. A greedy search heuristic is used to find an optimal MPM mode under the framework of minimum encoding inference. ECGP can represent the probability distribution for more than one node at a time. Thus, it extends PIPE in that the interactions among multiple nodes are considered.

Parameter Optimization with PSO. The Particle Swarm Optimization (PSO) conducts searches using a population of particles which correspond to individuals in evolutionary algorithm (EA). A population of particles is randomly generated initially. Each particle represents a potential solution and has a position represented by a position vector $\mathbf{x_i}$. A swarm of particles moves through the problem space, with the moving velocity of each particle represented by a velocity vector $\mathbf{v_i}$. At each time step, a function f_i representing a quality measure is calculated by using $\mathbf{x_i}$ as input. Each particle keeps track of its own best position, which is associated with the best fitness it has achieved so far in a vector $\mathbf{p_i}$. Furthermore, the best position among all the particles obtained so far in the population is kept track of as $\mathbf{p_g}$. In addition to this global version, another version of PSO keeps track of the best position among all the topological neighbors of a particle. At each time step t, by using the individual best position, $\mathbf{p_i}$, and the global best position, $\mathbf{p_g}(\mathbf{t})$, a new velocity for particle i is updated by

$$\mathbf{v_i}(\mathbf{t+1}) = \mathbf{v_i}(\mathbf{t}) + c_1\phi_1(\mathbf{p_i}(\mathbf{t}) - \mathbf{x_i}(\mathbf{t})) + c_2\phi_2(\mathbf{p_g}(\mathbf{t}) - \mathbf{x_i}(\mathbf{t})) \quad (1)$$

where c_1 and c_2 are positive constant and ϕ_1 and ϕ_2 are uniformly distributed random number in [0,1]. The term $\mathbf{v_i}$ is limited to the range of $\pm\mathbf{v_{max}}$. If the velocity violates this limit, it is set to its proper limit. Changing velocity this way enables the particle i to search around its individual best position, $\mathbf{p_i}$, and global best position, $\mathbf{p_g}$. Based on the updated velocities, each particle changes its position according to the following equation:

$$\mathbf{x_i}(\mathbf{t+1}) = \mathbf{x_i}(\mathbf{t}) + \mathbf{v_i}(\mathbf{t+1}). \quad (2)$$

For detailed description of general algorithm, please refer to [10] and [11].

3 Exchange Rates Forecasting Using FNT Paradigms

3.1 The Data Set

We used three different datasets in our forecast performance analysis. The data used are daily forex exchange rates obtained from the Pacific Exchange Rate Service [14], provided by Professor Werner Antweiler, University of British Columbia, Vancouver, Canada. The data comprises of the US dollar exchange rate against Euros, Great Britain Pound (GBP) and Japanese Yen (JPY). We used the daily data from 1 January 2000 to 31 October 2002 as training data set, and the data from 1 November 2002 to 31 December 2002 as evaluation test set or out-of-sample datasets (partial data sets excluding holidays), which are used to evaluate the good or bad performance of the predictions, based on evaluation measurements.

The forecasting evaluation criteria used is the normalized mean squared error (NMSE),

$$NMSE = \frac{\sum_{t=1}^{N}(y_t - \hat{y}_t)^2}{\sum_{t=1}^{N}(y_t - \bar{y}_t)^2} = \frac{1}{\sigma^2}\frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t)^2, \tag{3}$$

where y_t and $\hat{y}_t$ are the actual and predicted values, σ^2 is the estimated variance of the data and $\bar{y}_t$ the mean. The ability to forecast movement direction or turning points can be measured by a statistic developed by Yao and Tan [12]. Directional change statistics (Dstat) can be expressed as

$$D_{stat} = \frac{1}{N}\sum_{t=1}^{N} a_t \times 100\%, \tag{4}$$

where $a_t = 1$ if $(y_{t+1} - y_t)(\hat{y}_{t+1} - \hat{y}_t) \geq 0$, and $a_t = 0$ otherwise.

3.2 Experimental Results

For simulation, the five-day-ahead data sets are prepared for constructing FNT models. A FNT model was constructed using the training data and then the model was used on the test data set. The instruction sets used to create an optimal FNT forecaster is $S = F \bigcup T = \{+_2, +_3\} \bigcup \{x_1, x_2, x_3, x_4, x_5\}$. Where $x_i (i = 1, 2, 3, 4, 5)$ denotes the 5 input variables of the forecasting model.

The optimal FNTs evolved for three major internationally traded currencies: British pounds, euros and Japanese yen are shown in Figure 2. It should be noted that the important features for constructing the FNT model were formulated in accordance with the procedure mentioned in the previous section.

For comparison purpose, the forecast performances of a traditional multi-layer feed-forward network (MLFN) model and an adaptive smoothing neural network (ASNN) model are also shown in Table. The actual daily exchange rates and the predicted ones for three major internationally traded currencies are shown in Figure 3. From Table 1, it is observed that the proposed

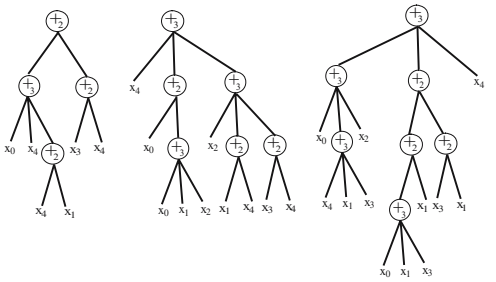


Fig. 2. The evolved FNT trees for forecasting euros (left), British pounds (middle) and Japanese yen (right)

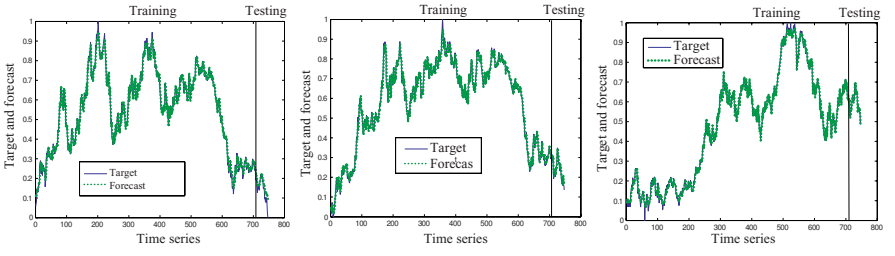


Fig. 3. The actual exchange rate and predicted ones for training dan testing data set, from left to right: Euros, British pounds Japanese yen

FNT forecasting models are better than other neural networks models for three major internationally traded currencies.

4 Conclusions

In this paper, we presented a Flexible Neural Tree (FNT) model for forecasting three major international currency exchange rates. We have demonstrated that the FNT forecasting model may provide better forecasts than the traditional MLFN forecasting model and the ASNN forecasting model. The comparative evaluation is based on a variety of statistical measures such as NMSE and D_{stat}. Our experimental analyses reveal that the NMSE and D_{stat} for three currencies using the FNT model are significantly better than those using the MLFN model

Table 1. Forecast performance evaluation for the three exchange rates: NMSE(D_{stat}) for testing)

Exchange rate	euros	British pounds	Japanese yen
MLFN [15]	0.5534(57.5%)	0.2137(55.0%)	0.2737(52.5%)
ASNN [15]	0.1254(72.5%)	0.0896(77.5%)	0.1328(67.5%)
FNT (This paper)	0.0180(81.0%)	0.0142(84.5%)	0.0084(74.5%)

and the ASNN model. This implies that the proposed FNT model can be used as a feasible solution for exchange rate forecasting.

Acknowledgments

This research was partially supported the Natural Science Foundation of China under contract number 60573065, and The Provincial Science and Technology Development Program of Shandong under contract number SDSP2004-0720-03.

References

1. Theodossiou, P.: The Stochastic Properties of Major Canadian Exchange Rates. The Financial Review 29(2)(1994) 193 - 221
2. So, M. K. P., Lam, K. and Li, W. K.: Forecasting Exchange Rate volatility using Autoregressive Random Variance Model. Appl. Finan. Economics 9(1999) 583 - 591
3. Hsieh, D.A.: Modeling Heteroscedasticity in Daily Foreign-Exchange Rates. J. of Business and Economic Statistics 7(1989) 307 - 317
4. Chappel, D., Padmore, J., Mistry P. and Ellis, C.: A Threshold Model for French Franc/Deutsch Mark Exchange Rate. J. of Forecasting 15(1996) 155 - 164
5. Refenes, A.N.: Constructive Learning and Its Application to Currency Exchange Rate Forecasting. In Neural Networks in Finance and Investing: Using Artificial Intelligence to Improve Real-World Performance. Probus Publishing Company, Chicago (1993) 777-805
6. Refenes, A.N., Azema-Barac, M., Chen L. and Karoussos, S.A.: Currency Exchange Rate Prediction and Neural Network Design Strategies. Neural Computing and Application 1(1993) 46-58
7. Yu, L., Wang, S, and Lai, K.-K.: Adaptive Smoothing Neural Networks in Foreign Exchange Rate Forecasting. Lecture Notes in Computer Science, Vol. 3516, Springer-Verlag, Berlin Heidelberg New York (2005)523-530
8. Yu, L., Wang, S. and Lai, K.-K.: A Novel Nonlinear Ensemble Forecasting Model Incorporating GLAR and ANN for Foreign Exchange Rates. Computers & Operations Research 32(2005) 2523 - 2541
9. Wang, W., Lai, K.-K., Nakamori, Y. and Wang, S.: Forecasting Foreign Exchange Rates with Artificial Neural Networks: A Review. Int. J. of Information Technology & Decision Making 3(1)(2004)145 - 165
10. Chen, Y., Yang, Y. and Dong, J.: Nonlinear System Modeling via Optimal Design of Neural Trees. Int. J. of Neural Systems 14(2)(2004) 125 - 137
11. Chen, Y., Yang, B., Dong, J. and Abraham, A.: Time-Series Forecasting using Flexible Neural Tree Model. Information Science 174(3-4)(2005) 219 - 235
12. Yao, J.T., Tan, C.L.: A Case Study on Using Neural Networks to Perform Technical Forecasting of Forex. Neurocomputing 34(2000) 79 - 98
13. Sastry, K., and Goldberg, D.E.: Probabilistic Model Building and Competent Genetic Programming. In R. L. Riolo and B. Worzel, editors, Genetic Programming Theory and Practise, Chapter 13 (2003)205 - 220
14. http://fx.sauder.ubc.ca/
15. Yu, L., Wang, S. and Lai, K.-K.: Adaptive Smoothing Neural Networks in Foreign Exchange Rate Forecasting. Lecture Notes in Computer Science, Vol. 3516, Springer-Verlag, Berlin Heidelberg New York (2005)523-530

Local Volatility Function Approximation Using Reconstructed Radial Basis Function Networks

Bo-Hyun Kim, Daewon Lee, and Jaewook Lee

Department of Industrial and Management Engineering,
Pohang University of Science and Technology,
Pohang, Kyungbuk 790-784, Korea
jaewookl@postech.ac.kr

Abstract. Modelling volatility smile is very important in financial practice for pricing and hedging derivatives. In this paper, a novel learning method to approximate a local volatility function from a finite market data set is proposed. The proposed method trains a RBF network with fewer volatility data and finds an optimized network through option pricing error minimization. Numerical experiments are conducted on S&P 500 call option market data to illustrate a local volatility surface estimated by the method.

1 Introduction

Volatility is known as one of the most important market variable in financial practice as well as in financial theory, although it is not directly observable in the market. The celebrated Black-Scholes (BS) model that assumes constant volatility has been widely used to estimate volatility (often called *implied volatility* since it calculates the volatility by inverting the Black-Scholes formula with option price data given by the market) [3, 14]. If the assumption of BS model is reasonable, the implied volatility should be the same for all option market prices.

In reality, however, it has been observed that the implied volatility shows strong dependence on strike price and time to maturity. This dependence, called the *volatility smile*, cannot be captured in BS model, and results in failure to produce an appropriate volatility for the corresponding option [17, 18].

One practical solution for the volatility smile is the constant implied volatility approach which is to simply use different volatilities for options with different strikes and maturities. In other words, if we had options with the whole range of strikes and maturities, we could simply calculate the implied volatility for each pair of strike and maturity by inverting the BS formula for each option. Although it works well for pricing simple European options, it cannot provide appropriate implied volatilities for pricing more complicated options such as exotic options or American options. Moreover, this approach can produce incorrect hedge factors like Gamma, Vega, Delta, etc. even for simple options [8, 12].

During the last decade, a number of researches based on different type of models have been conducted to model the volatility smile [15, 8, 1, 2]. One of the

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 524–530, 2006.

successful type of model is to use a 1-factor continuous diffusion model. This model, describing volatility as a function of strikes (or stocks) and maturities (called a *local volatility function*), is turned out to be a complete model by excluding non-traded source of risks and allows for arbitrage pricing and hedging [5]. When the underlying asset follows this model, one important task is then to accurately approximate the local volatility function for accurately pricing exotic options and computing correct hedge factors [4]. Estimating a local volatility function is, however, a nontrivial task since it is generally an ill-posed problem due to insufficient market option price data.

In this paper, we propose a novel learning method to approximate a local volatility function from a finite market data set in pricing and hedging derivatives. The proposed method consists of two phases. In the first phase, we train a preparatory radial basis function (RBF) network from an available volatility data to estimate the local volatility surface and then obtain the estimated volatilities for the whole range of strikes and maturities. In the second phase, we reconstruct an optimized RBF network by controlling the volatility values of a preparatory RBF network to minimize the error between estimated option prices and real market option prices.

The organization of this paper is as follows. In section 2, we propose a learning method to model a volatility smile. Computational simulation applied to the S& P 500 index options are conducted in Section 3. Section 4 concludes the result.

2 The Proposed Method

The 1-factor continuous diffusion model assumes the underlying asset follows the following process with the initial value S_{init}:

$$\frac{dS_t}{S_t} = \mu(S_t, t)dt + \sigma(S_t, t)dW_t, \quad t \in [0, \tau], \quad \tau > 0 \tag{1}$$

where τ is a fixed time horizon, W_t is a standard Brownian motion and $\mu(s,t)$, $\sigma(s,t) : \mathbb{R}^+ \times [0,\tau] \to \mathbb{R}$ are deterministic functions sufficiently well behaved to guarantee that (1) has a unique solution [9]. $\sigma(s,t)$ is the local volatility function.

When we estimate a local volatility function from a finite data set, we can avoid over-fitting problem by regularizing with some kind of smoothness of the local volatility function. A RBF network is a well-known method that is capable of solving ill-posed problems with regularization [7].

In this section, we propose a method to approximate the local volatility function using RBF networks when the underlying asset follows a 1-factor model (1). The local volatility function $\sigma(s,t)$ will be explicitly represented by a reconstructed RBF network.

2.1 Phase I: Initial Local Volatility Function Approximation Using RBF Networks

In the first phase, we initially approximate a local volatility function by using RBF networks. A RBF network is constructed with a given training volatility

data set. Each training input has two attributes; strike price and time to maturity, (K_j^{tr}, T_j^{tr}), $j = 1, ..., l$ and training output has its corresponding local volatility value, σ_j. A RBF network involves searching for a suboptimal solution in a lower-dimensional space that approximates the interpolation solution where the approximated solution $\hat{\sigma}_{RBF}(\mathbf{w})$ can be expressed as follows:

$$\hat{\sigma}_{RBF}(\mathbf{w}; K, T) = \sum_{j=1}^{l} w_j \phi \left(\frac{\|(K, T) - (K_j^{tr}, T_j^{tr})\|}{\eta} \right) \tag{2}$$

where each $\phi = \phi((K, T), (K_j^{tr}, T_j^{tr}))$ is a radial basis function centered at (K_j^{tr}, T_j^{tr}) and η is an user pre-specified scale parameter. The training procedure of the RBF network is to estimate the weights that connect the hidden and the output layers and these weights will be directly estimated by using the least squares algorithm [7, 6, 11].

To determine the optimal network weights that connect the hidden and the output layers from a training data set, $\{K_j^{tr}, T_j^{tr}, \sigma_j\}_{j=1}^{l}$, we fit an initial local volatility surface by minimizing the following criterion function with a weight decay regularization term

$$J(\mathbf{w}) = \sum_{j=1}^{l} \|\sigma_j - \hat{\sigma}_{RBF}(\mathbf{w}; K_j^{tr}, T_j^{tr})\|^2 + \lambda \|\mathbf{w}\|^2. \tag{3}$$

where λ is a regularization parameter introduced to avoid over-fitting. The network weights can then be explicitly given by

$$\mathbf{w} = (\mathbf{\Phi} + \lambda I)^{-1} \boldsymbol{\sigma} \tag{4}$$

where $\mathbf{\Phi} = [\phi((K_i^{tr}, T_i^{tr}), (K_j^{tr}, T_j^{tr}))]_{i,j=1,...,l}$ and $\boldsymbol{\sigma} = (\sigma_1, ..., \sigma_l)^T$. Substituting Eq. (4) into Eq. (2) makes us to rewrite $\hat{\sigma}_{RBF}(\mathbf{w}; K, T)$ as $\hat{\sigma}_{RBF}(\boldsymbol{\sigma}; K, T)$.

Initially, we do not use the training volatility information $\boldsymbol{\sigma}$ at this stage for the estimated option prices to match the market option prices as closely as possible. Instead, we randomly generate the initial volatility vector, $\boldsymbol{\sigma}^{(0)} \in \Re^l$ in Eq. (4), as nonnegative values. For this reason, we will call the RBF network obtained in the first phase a preparatory RBF network.

2.2 Reconstructing RBF Networks Via Pricing Error Minimization

To obtain a better local volatility function that minimizes the option pricing error, in the second phase, we reconstruct the RBF network by iteratively updating the current volatility vector $\boldsymbol{\sigma}^{(0)}$ to a better one.

Let p_i be the i-th option market price with (K_i, T_i) as its strike price and maturity for $i = 1, ..., m$. Note that the set $\{(K_i, T_i)\}_{i=1}^{m}$ has its corresponding option price p_i and is a different set from the training input data set $\{(K_j^{tr}, T_j^{tr})\}_{j=1}^{l}$. Also let P_i be the i-th option pricing formula (highly nonlinear function with respect to its input) where its strike price, maturity, and volatility are given at $K_i, T_i, \hat{\boldsymbol{\sigma}}_{RBF}$, respectively. Utilizing this information, Phase II tries to find

an optimal volatility, $\boldsymbol{\sigma}^* = (\sigma_1^*, ..., \sigma_l^*)$, to reconstruct a final RBF network by solving the following nonlinear optimization:

$$\min_{\boldsymbol{\sigma}} E(\boldsymbol{\sigma}) = \sum_{i=1}^{m} \{p_i - P_i[K_i, T_i, \hat{\boldsymbol{\sigma}}_{RBF}(\boldsymbol{\sigma}; K_i, T_i)]\}^2 + \alpha \sum_{j=1}^{l} \|\sigma_j^{tr} - \hat{\sigma}_{RBF}(\boldsymbol{\sigma}; K_j^{tr}, T_j^{tr})\|^2 \tag{5}$$

where σ_j^{tr} is a volatility output at a training input (K_j^{tr}, T_j^{tr}) and α is an user-controllable compensation parameter. In this paper, we will call the RBF corresponding to this optimal volatility $\boldsymbol{\sigma}^*$ as a reconstructed RBF.

To get the optimal solution that minimizes Eq. (5) efficiently, we employ a trust region algorithm described as follows. For a volatility vector $\boldsymbol{\sigma}(n)$ at iteration n, the quadratic approximation $\hat{E}$ is defined by the first two terms of the Taylor approximation to E at $\boldsymbol{\sigma}(n)$;

$$\hat{E}(\mathbf{s}) = E(\boldsymbol{\sigma}(n)) + \mathbf{g}(n)^T \mathbf{s} + \frac{1}{2}\mathbf{s}^T \mathbf{H}(n)\mathbf{s} \tag{6}$$

where $\mathbf{g}(n)$ is the local gradient vector and $\mathbf{H}(n)$ is the local Hessian matrix. A trial step $\mathbf{s}(n)$ is then computed by minimizing (or approximately minimizing) the trust region subproblem stated by

$$\min_{\mathbf{s}} \hat{E}(\mathbf{s}) \quad \text{subject to} \quad \|\mathbf{s}\|_2 \leq \Delta_n \tag{7}$$

where $\Delta_n > 0$ is a trust-region parameter. According to the agreement between predicted and actual reduction in the function E as measured by the ratio

$$\rho_n = \frac{E(\boldsymbol{\sigma}(n)) - E(\boldsymbol{\sigma}(n) + \mathbf{s}(n))}{\hat{E}(\mathbf{0}) - \hat{E}(\mathbf{s}(n))}, \tag{8}$$

Δ_n is adjusted between iterations as follows:

$$\Delta_{n+1} = \begin{cases} \|\mathbf{s}(n)\|_2/4 & \text{if } \rho_n < 0.25 \\ 2\Delta_n & \text{if } \rho_n > 0.75 \text{ and } \Delta_n = \|\mathbf{s}(n)\|_2 \\ \Delta_n & \text{otherwise} \end{cases} \tag{9}$$

The decision to accept the step is then given by

$$\boldsymbol{\sigma}(n+1) = \begin{cases} \boldsymbol{\sigma}(n) + \mathbf{s}(n) & \text{if } \rho_n \geq 0 \\ \boldsymbol{\sigma}(n) & \text{otherwise} \end{cases} \tag{10}$$

which means that the current weight vector is updated to be $\boldsymbol{\sigma}(n) + \mathbf{s}(n)$ if $E(\boldsymbol{\sigma}(n) + \mathbf{s}(n)) < E(\boldsymbol{\sigma}(n))$; Otherwise, it remains unchanged and the trust region parameter Δ_n is shrunk and the trial step computation is repeated. One nice property of the trust region method is that this method retains the rapid rate of convergence of Newton's method, but is also generally applicable and globally convergent [10, 13, 16].

3 Computational Examples

We use the S&P 500 Index European call option data of October 1995, which is also used in [1, 4]. The market option price data is given in Table 1. Only the options with no more than two years maturity are used for accuracy. The initial index, interest rate, and dividend rate are set as follows:

$$S_{init} = \$590,\ r = 0.06,\ q = 0.0262$$

In Phase I, we used the following 16 training volatility input data to train a preparatory RBF network:

$$(K_j^{tr}, T_j^{tr}) = (K(p), T(q)),\ j = 4(p-1)+q, \quad p, q = 1, ..., 4$$
$$K = [0.8000S_{init}, 0.9320S_{init}, 1.0640S_{init}, 1.3940S_{init}]$$
$$T = [0, 0.66, 1.32, 1.98]$$

In Phase II, we used 70 S&P 500 Index option price data given in Table 1.

Table 1. S&P 500 Index call option price

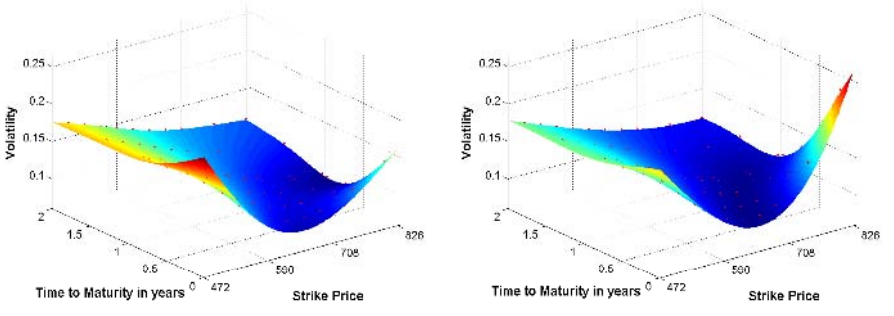
Maturity	Strike(% of spot price)									
	85%	90%	95%	100%	105%	110%	115%	120%	130%	140%
.175	$9.13e^{1}$	$6.28e^{1}$	$3.52e^{1}$	$1.29e^{1}$	2.11	$1.21e^{-1}$	$3.73e^{-2}$	$1.62e^{-2}$	$1.65e^{-3}$	$5.14e^{-4}$
.425	$9.63e^{1}$	$6.91e^{1}$	$4.40e^{1}$	$2.33e^{1}$	8.54	2.26	$4.21e^{-1}$	$1.94e^{-1}$	$2.77e^{-2}$	$8.72e^{-3}$
.695	$1.02e^{2}$	$7.61e^{1}$	$5.26e^{1}$	$3.26e^{1}$	$1.64e^{1}$	5.95	1.90	$6.04e^{-1}$	$7.25e^{-2}$	$2.66e^{-2}$
.94	$1.07e^{2}$	$8.22e^{1}$	$5.99e^{1}$	$3.99e^{1}$	$2.38e^{1}$	$1.13e^{1}$	4.71	1.78	$1.82e^{-1}$	$4.48e^{-2}$
1	$1.08e^{2}$	$8.36e^{1}$	$6.16e^{1}$	$4.16e^{1}$	$2.54e^{1}$	$1.28e^{1}$	5.50	2.13	$2.27e^{-1}$	$5.44e^{-2}$
1.5	$1.17e^{2}$	$9.44e^{1}$	$7.31e^{1}$	$5.40e^{1}$	$3.73e^{1}$	$2.37e^{1}$	$1.43e^{1}$	7.65	1.85	$3.10e^{-1}$
2	$1.26e^{2}$	$1.04e^{1}$	$8.36e^{1}$	$6.49e^{1}$	$4.82e^{1}$	$3.42e^{1}$	$2.36e^{1}$	$1.47e^{1}$	5.65	1.78

The proposed method is compared with a popularly used natural cubic spline method (cf. [4]). The spline approach normally uses the same amount of training data (with a different set of strike prices and maturities) for its knot points as that of option market data to get a reasonable performance. The spline function is then reconstructed from the 70 option market data by minimizing option pricing error with respect to 70 volatility input variables. The criteria for comparison are pricing error (MSE; mean squared error) and hedging error between predicted volatility and true implied volatility.

Table 2 demonstrates the feature of our proposed method: that is, the proposed method achieved a better or comparable performance compared to spline approach in pricing and hedging derivatives with fewer input variables (16 in our method and 70 in spline approach). Fig. 1 shows the estimated volatility surfaces generated by the spline and the proposed. The proposed method showed more skewed volatility smile effects as observed in practice.

Table 2. Accuracy of Pricing and Hedging

	Measure	Reconstructed Spline	Reconstructed RBF
Pricing	Price	0.0317	0.0284
Hedging	Vega	7.1153	5.8509
	Delta	$1.1535e^{-5}$	$1.0255e^{-5}$
	Gamma	$1.3316e^{-8}$	$1.0839e^{-8}$
	Rho	1.5961	1.6449
	Theta	0.2935	0.3129



(a) (b)

Fig. 1. Estimated volatility surfaces. (a) spline approach and (b) proposed method.

4 Conclusion

In this paper, we've proposed a novel learning method to approximate the local volatility function. The proposed method first trains a preparatory RBF network with a training volatility data set to estimate volatility surface and then reconstructs an optimized RBF network by minimizing the errors between estimated prices and real market prices. A simulation has been conducted with a S&P500 option data example. The experimental results demonstrated a performance improvement of the proposed method compared to other approach in terms of pricing and hedging error.

Acknowledgement. This work was supported by the Korea Research Foundation under grant number KRF-2004-041-D00785.

References

1. Andersen, L.B.G., Brotherton-Ratcliffe, R.: The Equity Option Volatility Smile: An Implicit Finite Difference Approach. Journal of Computational Finance 1(2) (1997)
2. Avellaneda, M., Friedman, M., Holmes, C., Samperi, D.: Calibrating Volatility Surfaces via Relative Entropy Minimazation. Applied Mathematical Finances 4 (1997) 37-64

3. Black, F., Scholes, M.: The Pricing of Options and Corporate Liabilities. Journal of Political Economy 81 (1973) 637-659
4. Coleman, T.F., Li, Y., Verma, A.: Reconstructing the Unknown Local Volatility Function. The Journal of Computational Finance 2(3) (1999)
5. Dupire, B.: Pricing with a Smile. Risk 7 (1994) 18-20
6. Han, G.S., Lee, D., Lee J. : Estimating The Yield Curve Using Calibrated Radial Basis Function Networks. Lecture Notes in Computer Science 3497 (2005) 885-890
7. Haykin, S.: Neural Networks: a Comprehensive Foundation. Prentice-Hall New York (1999)
8. Hull, J., White, A.: The Pricing of Options on Assets with Stochastic Volailities. Journal of Finance 3 (1987) 281-300
9. Lamberton, D., Lapeyre, B.: Introduction to Stochastic Calculus Applied to Finance. Chapman & Hall (1996)
10. Lee, J.: Attractor-Based Trust-Region Algorithm for Efficient Training of Multilayer Perceptrons. Electronics Letters 39 (2003) 71-72
11. Lee, D., Lee, J.: A Novel Three-Phase Algorithm for RBF Neural Network Center Selection. Lecture Notes in Computer Science 3173 (2005) 350-355
12. Lee, H.S., Lee, J., Yoon, Y.G., Kim, S.: Coherent Risk Meausure Using Feedfoward Neural Networks. Lecture Notes in Computer Science 3497 (2005) 904-909
13. Lee, J., Chiang, H.D.: A Dynamical Trajectory-Based Methodology for Systematically Computing Multiple Optimal Solutions of General Nonlinear Programming Problems. IEEE Trans. on Automatic Control 49(6) (2004) 888 - 899
14. Merton, R.: The Theory of Rational Option Pricing. Bell Journal of Economics and Managemenet Science 4 (1973) 141-183
15. Merton, R.: Option Pricing when Underlying Stock Returns Are Discontinuous. Journal of Financial Economics 3 (1976) 124-144
16. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer New York (1999)
17. Rubinstein, M.: Implied Binomial Trees. The Journal of Finance 49 (1994) 771-818
18. Shimko,D.: Bounds of Probability. Risk (1993) 33-37

Visualization of Dynamic Brain Activities Based on the Single-Trial MEG and EEG Data Analysis

Jianting Cao[1,2], Liangyu Zhao[1], and Andrzej Cichocki[2]

[1] Department of Electronic Engineering, Saitama Institute of Technology, 1690 Fusaiji, Fukaya-shi, Saitama 369-0293, Japan
cao@sit.ac.jp

[2] The Lab. for Advanced Brain Signal Processing, Brain Science Institute, RIKEN, 2-1 Hirosawa, Wako-shi, Saitama 351-0198, Japan

Abstract. Treating an averaged evoked-fields (EFs) or event-related potentials (ERPs) data is a main approach in the topics on applying Independent Component Analysis (ICA) to neurobiological signal processing. By taking the average, the signal-noise ratio (SNR) is increased, however some important information such as the strength of an evoked response and its dynamics (trial-by-trial variations) will be lost. The single-trial data analysis, on the other hand, can avoid this problem but the poor SNR is necessary to be improved.

This paper presents a robust multi-stage data analysis method for the single-trial Magnetoencephalograph (MEG) and Electroencephalograph (EEG) recorded data. In the pre-processing stage, a robust subspace method is firstly applied for reducing a high-level unique component (additive noise) in single-trial raw data. In the second stage, a parameterized **t**-distribution ICA method is applied for further decomposing the overlapped common components (sources). In the post-processing stage, the source localization or scalp mapping technique and post-averaging technique are applied for visualizing the dynamic brain activities. The results on single-trial MEG and EEG data analysis both illustrate the high performances not only in the visualization of the behavior and location but also in the visualization of the trial-by-trial variations of individual evoked brain response.

1 Introduction

Applying the ICA to physiological data has received a lot of attentions due to many practical results have been achieved[1-4]. The analysis of single-trial data is more interested since many pieces of important information such as the strength and dynamics of an evoked response can be visualized from the data. This paper presents a practical method for decomposing and localizing MEG and EEG single-trial data.

The problem of the single-trial data analysis is formulated by

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \boldsymbol{\xi}(t), \quad t = 1, 2, \cdots, \tag{1}$$

where $\mathbf{x}(t) = [x_1(t), \cdots, x_m(t)]^T$ represent the transpose of m observations at time t. Each observation $x_i(t)$ contains n common components (sources) $\mathbf{s}(t) =$

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 531–540, 2006.

$[s_1(t), \cdots, s_n(t)]^T$ and a unique component (additive noise) which is included in the vector $\boldsymbol{\xi}(t) = [\xi_1(t), \cdots, \xi_m(t)]^T$. Since the tissue and skull do not attenuate magnetic fields in an MEG measurement, $\mathbf{A} \in \mathbf{R}^{m \times n} = (a_{ij})$ can be represented by a numerical matrix whose element a_{ij} is a quantity related to the physical distance between i-th sensor and j-th source. In EEG signal processing, $\mathbf{A} = (a_{ij})$ is also usually assumed as a numerical matrix.

In this model, the sources $\mathbf{s}$ and their number n, additive noise $\boldsymbol{\xi}$ and matrix $\mathbf{A}$ are unknown but the sensor signals $\mathbf{x}$ are accessible. It is assumed that the components of $\mathbf{s}$ are mutually statistically independent, as well as statistically independent to the noise components $\boldsymbol{\xi}$. Moreover, the noise components $\boldsymbol{\xi}$ themselves are assumed to be mutually independent.

Some remarks about the model :

- Regardless of the nature of components, according to the model, a source component s_i contributes to at least two sensors and a noise component ξ_i contributes at most to only one sensor. Based on this definition, we can easily distinguish a noise component from the source components by checking the distribution of amplitude of the observed signals on the nearby sensors. If the distribution of amplitude of these signals is not smooth, this means a noise component is added in one of these sensors. This technique sometimes can help us in selecting of a suitable model.
- Depending on the condition of the experiment, the sources $\mathbf{s}$ usually include: (1) 'brain sources' such as evoked responses, spontaneous and artifacts; (2) 'interference sources' such as power interference; environmental interferences. These sources contain either a positive kurtosis or a negative kurtosis. Therefore, the separating of the mixture of sub-Gaussian and super-Gaussian signals is necessary.
- The number of sensors m is fixed depending on the MEG or EEG machine. The number of sources n is unknown in the model, and it has to be estimated or to be conjectured by using a prior.

There are two kinds of undesirable components have to be removed in our task. The first one is additive noises $\boldsymbol{\xi}$, their power will be reduced by the robust subspace method in the pre-processing step. The second one, they are usually to be called 'noise' (interference, brain noises etc.) but they are attributed to the source in our model. They will be discarded after ICA source decomposition.

2 Method of Single-Trial Data Analysis

2.1 Robust Pre-whitening with Noise Reduction

In this subsection, we first describe the standard Principal Component Analysis (PCA) approach for the pre-whitening. Next, we show that this standard PCA approach can be extended to pre-whitening with a high-level noise reduction.

Let us rewrite Eq. (1) in a data matrix form as

$$\mathbf{X}_{(m \times N)} = \mathbf{A}_{(m \times n)} \mathbf{S}_{(n \times N)} + \boldsymbol{\Xi}_{(m \times N)}, \tag{2}$$

where N denotes data samples. When the sample size N is sufficiently large, the covariance matrix of the observed data can be written as

$$\boldsymbol{\Sigma} = \mathbf{A}\mathbf{A}^T + \boldsymbol{\Psi}, \tag{3}$$

where $\boldsymbol{\Sigma} = \mathbf{X}\mathbf{X}^T/N$, and $\boldsymbol{\Psi} = \boldsymbol{\Xi}\boldsymbol{\Xi}^T/N$ is a diagonal matrix. For convenience, we assume that $\mathbf{X}$ has been divided by $\sqrt{N}$ so that the covariance matrix can be given by $\mathbf{C} = \mathbf{X}\mathbf{X}^T$.

For the averaged data, the noise variance $\boldsymbol{\Psi}$ is small or zero. A cost function for fitting the model to the data is to make $\mathbf{C} - \mathbf{A}\mathbf{A}^T$ as small as possible. It is well known that the standard PCA can find the principal components by employing the eigen-value decomposition. That is, the solution of $\mathbf{A}\mathbf{A}^T$ for seeking n principal components can be obtained by

$$\hat{\mathbf{A}}\hat{\mathbf{A}}^T = \mathbf{U}_n \boldsymbol{\Lambda}_n \mathbf{U}_n^T, \tag{4}$$

where $\boldsymbol{\Lambda}_n$ is a diagonal matrix whose elements are the n largest eigenvalues of $\mathbf{C}$. The columns of $\mathbf{U}_n$ are the corresponding eigenvectors. In Eq. (4), let one possible solution for $\hat{\mathbf{A}}$ is

$$\hat{\mathbf{A}} = \mathbf{U}_n \boldsymbol{\Lambda}_n^{1/2}, \tag{5}$$

and then the scores can be obtained from as $\mathbf{z} = \boldsymbol{\Lambda}_n^{-1/2}\mathbf{U}_n^T\mathbf{x}$. Note that the covariance matrix is $E\{\mathbf{z}\mathbf{z}^T\} = \mathbf{I}_n$, it means that $\mathbf{z}$ are orthogonal. Applying this algorithm for the averaged data analysis, several successful results have been reported [3].

For unaveraged single-trial data, the SNR is usually very low. This means that the diagonal elements of $\boldsymbol{\Psi}$ cannot be ignored in the model. In this case, we can fit $\mathbf{A}\mathbf{A}^T$ to $\mathbf{C} - \boldsymbol{\Psi}$ by the eigen-value decomposition. That is, choosing the columns of $\mathbf{A}$ as eigenvectors of $\mathbf{C} - \boldsymbol{\Psi}$ corresponding to the n largest eigenvalues so that the sum of the squares in each column is identical to the corresponding eigenvalue.

Noted that the noise variance $\boldsymbol{\Psi}$ is assumed to be known. However, in practice $\boldsymbol{\Psi}$ is unknown, and it has to be estimated. Motivated by this, we employ the cost function

$$L(\mathbf{A}, \boldsymbol{\Psi}) = tr[\mathbf{A}\mathbf{A}^T - (\mathbf{C} - \boldsymbol{\Psi})]^2 \tag{6}$$

and minimize it by $\frac{\partial L(\mathbf{A}, \boldsymbol{\Psi})}{\partial \boldsymbol{\Psi}} = 0$, whereby the estimate

$$\hat{\boldsymbol{\Psi}} = diag(\mathbf{C} - \hat{\mathbf{A}}\hat{\mathbf{A}}^T) \tag{7}$$

is obtained. The estimate $\hat{\mathbf{A}}$ can be obtained in the same way as in Eq. (5).

Both the matrix $\mathbf{A}$ and the diagonal elements of $\boldsymbol{\Psi}$ have to be estimated together from data. The estimate $\hat{\mathbf{A}}$ is obtained by the standard PCA. The estimate $\hat{\boldsymbol{\Psi}}$ is obtained by the so called unweighted least squares method that is one of the estimation methods in factor analysis. Once the estimates $\hat{\mathbf{A}}$ and $\hat{\boldsymbol{\Psi}}$ converge to stable values, we can finally compute the score matrix by using the Bartlett method as

$$\mathbf{Q} = [\hat{\mathbf{A}}^T \hat{\boldsymbol{\Psi}}^{-1} \hat{\mathbf{A}}]^{-1} \hat{\mathbf{A}}^T \hat{\boldsymbol{\Psi}}^{-1}. \tag{8}$$

Using the above result, the new transformation data can be obtained by $\mathbf{z} = \mathbf{Q}\mathbf{x}$. Note that the covariance matrix is $E\{\mathbf{z}\mathbf{z}^T\} = \mathbf{I}_n + \mathbf{C}\boldsymbol{\Psi}\mathbf{C}^T$, which implies that the subspace of the source signals are decorrelated.

The robust approach plays the same role in decorrelation as the standard PCA, but the noise variance $\boldsymbol{\Psi}$ is taken into account. The difference is that the standard PCA is to fit both diagonal and off-diagonal elements of $\mathbf{C}$, whereas the robust pre-whitening technique is to fit off-diagonals elements of $\mathbf{C}$ only. Based on this property, the robust approach is enable us to reduce a high-level noise, and which is very important in the single-trial data analysis.

2.2 Optimal Dimensionality Reduction

The cross-validatory techniques have been wildly applied in multivariate statistics. It usually divides the data into two groups, and uses one group to determine some characteristics of the data, and then uses the other groups to verify the characteristics. Extending this concept, we propose a criterion for determining the estimation of the source number $\hat{n}$ by using the error of estimating the noise variance.

Let us first divides the data matrix $\mathbf{X}$ into several disjoint groups such as $\mathbf{X}_i \in \mathbf{R}^{m \times N/K}$, where N is data samples and the group number $i = 1, \cdots, K$. Next, we use each group data to compute one estimate of the noise variance $\mathrm{diag}(\hat{\boldsymbol{\Psi}}_i)$ and use remaining data to compute another estimate of the noise variance $\mathrm{diag}(\hat{\boldsymbol{\Psi}}_j)$ where $j \neq i$. In general, when the estimate of source number $\hat{n}$ has not been marched to its true value, a larger error will arise between the noise variance and its estimate. Based on this property, we propose a criterion for $\hat{n}$ as,

$$\mathrm{Error}(\hat{n}) = \frac{1}{K}\sum_{i=1}^{K} \mathrm{tr}[\mathrm{diag}(\hat{\boldsymbol{\Psi}}_i^{(\hat{n})}) - \mathrm{diag}(\hat{\boldsymbol{\Psi}}_j^{(\hat{n})})]^2. \tag{9}$$

It should be noted that we are not necessary to compute all of the estimates of the source number such as from $\hat{n} = 1$ to $\hat{n} = m$ when applying a sufficient condition as $\hat{n} \leq \frac{1}{2}(2m + 1 - \sqrt{8m+1})$. Within this bound, we know that the estimate of the source number is reliable.

2.3 Robust Nonlinear Function in ICA Algorithm

After pre-processing the data, a new data vector $\mathbf{z} = \mathbf{Q}\mathbf{x}$ is obtained in which the power of noises, mutual correlation and dimensionality have been reduced. The decomposed independent sources $\mathbf{y} \in \mathbf{R}^n$ then can be obtained from a linear transformation as

$$\mathbf{y}(t) = \mathbf{W}\mathbf{z}(t), \tag{10}$$

where $\mathbf{W} \in \mathbf{R}^{n \times n}$ is the demixing matrix that can be computed by using several standard ICA methods. For example, applying the natural gradient based approach, an updating rule is

$$\Delta\mathbf{W}(t) = \eta \left[\mathbf{I} - \boldsymbol{\varphi}(\mathbf{y}(t))\mathbf{y}^T(t)\right] \mathbf{W}(t), \tag{11}$$

where $\eta > 0$ is a learning rate, and $\boldsymbol{\varphi}(\cdot)$ is the vector of activation functions whose optimal components are

$$\varphi_i(y_i) = -\frac{d}{dy_i}\log p_i(y_i) = -\frac{\dot{p}_i(y_i)}{p_i(y_i)}, \tag{12}$$

where $\dot{p}_i(y_i) = dp_i(y_i)/dy_i$.

Typical ICA algorithms rely on the choice of nonlinear functions. The form of the optimal function depend on the probability distribution of the source which is usually not available in the ICA task. Several algorithms have been developed for separating the mixtures of sub- and super-Gaussian sources. In this paper, we will use the developed the parameterized **t**-distribution model (unimodal) [2]. We will not go into detail in the theoretical analysis but show some advantages for the derived nonlinear function : (1) the function is robust to outliers; (2) the nonlinear function is determined by the value of the kurtosis which corresponds to the source distribution; (3) the algorithm holds stability as well as it robust to the misestimation of kurtosis. The proposed nonlinear functions have the forms:

$$\varphi_i(y_i) = \alpha\lambda_\alpha \operatorname{sgn}(y_i)|\lambda_\alpha y_i|^{\alpha-1}, \qquad \hat{\kappa}_i \leq 0, \tag{13}$$

$$\varphi_i(y_i) = \frac{(1+\beta)y_i}{y_i^2 + \frac{\beta}{\lambda_\beta^2}} \qquad \hat{\kappa}_i > 0. \tag{14}$$

where α and β are the parameters which control the shape of a family of distributions (sub-Gaussian or super-Gaussian), respectively.

Implementation of the proposed ICA algorithm is summarized as [2]:

- Calculate the output $\mathbf{y}$ from given observations $\mathbf{z}$ and an initial value $\mathbf{W}$.
- Calculate the kurtosis by $\hat{\kappa}_i = \hat{m}_4/\hat{m}_2^2 - 3$ where the 2nd- and 4th-order moments are estimated by $\hat{m}_j(t) = [1-\eta(t)]\hat{m}_j(t-1) + \eta(t)y_i^j(t), (j = 2, 4)$.
- Establish the look-up tables by $\kappa_\alpha = \frac{\Gamma(\frac{5}{\alpha})\Gamma(\frac{1}{\alpha})}{\Gamma^2(\frac{3}{\alpha})} - 3$ and $\kappa_\beta = \frac{3\Gamma(\frac{\beta-4}{2})\Gamma(\frac{\beta}{2})}{\Gamma^2(\frac{\beta-2}{2})} - 3$, and determine α or β corresponding to the value of $\hat{\kappa}_i$.
- Calculate the scaling constants $\lambda_\alpha = \left[\frac{\Gamma(\frac{3}{\alpha})}{m_2\Gamma(\frac{1}{\alpha})}\right]^{\frac{1}{2}}$ or $\lambda_\beta = \left[\frac{\beta\Gamma(\frac{\beta-2}{2})}{2m_2\Gamma(\frac{\beta}{2})}\right]^{\frac{1}{2}}$ according to the value of $\hat{\kappa}_i$.
- Calculate the nonlinear function by Eq. (13) or Eq. (14) and update $\mathbf{W}$ by Eq. (11).

2.4 Single-Dipole Source Localization

After pre-processing and applying ICA to the single-trial data, the individual sources have been extracted. We will post-processing the decomposed sources for seeking their location, amplitude and dynamics by using the spatio-temporal dipole fit method.

Using the matrices $\mathbf{W}$ and $\mathbf{Q}$ or $\hat{\mathbf{A}}$, we can obtain the estimated behavior of the brain activities as

$$\hat{\mathbf{s}}(t) = \mathbf{y}(t) = \mathbf{G}\mathbf{x}(t), \tag{15}$$

where $\mathbf{G} = \mathbf{WQ}$. To seek the location of the decomposed independent sources, let $\hat{\mathbf{A}}_g = \hat{\mathbf{A}}\mathbf{W}^{-1}$, where $\hat{\mathbf{a}}_i$ is i-th column of matrix $\hat{\mathbf{A}}_g$ and $\hat{s}_i$ is ith component in the estimated sources vector $\hat{\mathbf{s}}$ $(i = 1, \cdots, n)$. The virtual contribution of multi-sources or a single source to the sensors can be represented by

$$\hat{\mathbf{x}}(k) = \hat{\mathbf{A}}_g \hat{\mathbf{s}}(k) \qquad \text{or} \qquad \hat{\mathbf{x}}(k) = \hat{\mathbf{a}}_i \hat{s}_i(k), \tag{16}$$

where k is the sample index of data samples of a single-trial, $\hat{s}_i$ is an independent source which is usually selected from our interest.

3 Results for Single-Trial Data Analysis

3.1 MEG Single-Trial Data Analysis

The auditory-evoked-fields (AEF) data was recorded by an MEG system (made in CTF Systems Inc., Canada) in Japan. The sensor arrays consists of 64 MEG channels. The AEF experiment was performed on a normal male adult whose both ears were stimulated by 1 kHz tone. There were 630 sets of trial data recorded in 379.008 sec. The duration of each single-trial was 0.6016 sec and the stimulus was given at 0.2 sec. The sampling rate was 312.5 Hz and the samples were 188 in each trial.

Taking an average of 630 trials, and localizing the evoked fields by using the dipole fitting method, we obtain averaged map as shown in Fig. 1. This is a typical result of averaged AEF analysis in which the two dipoles appear in the left and right temporal cortex. The latency was set at 0.096 sec in the fitting process. As seen from this map, the maximum evoked response is 330 fT. That represents the averaged strength of evoked fields.

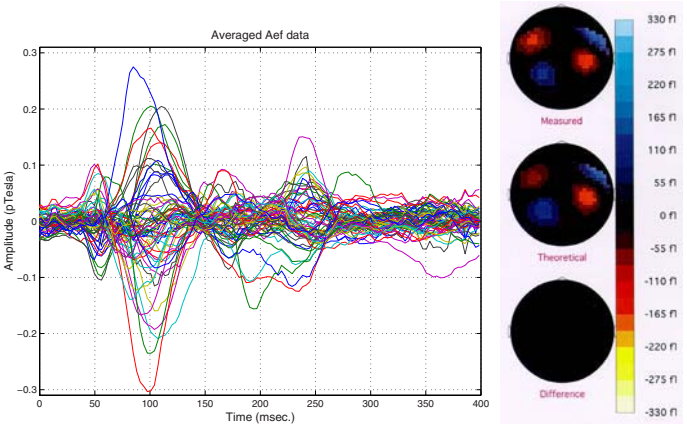


Fig. 1. Results for averaged AEF data and AEF's locations

Further beyond the behavior and location of the evoked response as in averaged data, we search the activity strength information of each single evoked response and its dynamics (the strength of the evoked response corresponding to related stimulus). As a typical example, we show the results for the first single-trial data in Fig. 2. Two independent components (IC1 and IC3) are successfully extracted corresponding to the N100 evoked responses. IC2 is a typical alpha-wave component of 11 Hz and IC4 has a high frequency which may has been effected by environment interference. Projecting IC1 and IC3 onto the sensor space by using Eq. (16) and localizing them independently, the head map for the individual components is obtained (see Fig. 2). The map in Fig. 2 left shows that the magnetic field distribution for the decomposed IC1 is in the right side of the temporal cortex. The maximum response for IC1 is 184 fT evoked at 110 ms in the first trial. The map in Fig. 2 right shows that the magnetic field distribution for the decomposed IC3 is in the left side of the temporal cortex. The maximum response for IC3 is 721 fT at 101 ms in the first trial.

Let us focus the discussion on N100 evoked responses by the ICA decomposition and the average of stimulus trials. Comparing the two maps derived by ICA in Figs. 2 with the averaged map in Fig. 1, we can easily find that the two evoked individual responses IC1 and IC3 correspond to the averaged map in their locations. It is impossible to obtain the amplitude information of a single evoked response from the averaged map. However, by using ICA approach for the unaveraged single-trial data, the amplitude information (activity strength) for each individual evoked response has been obtained. Moreover, based on the results in Fig. 2, we note that the evoked response in the left side of the temporal cortex IC3 is stronger than that in the right side IC1 when given the first stimulus.

In order to observe the dynamics of the evoked response, we investigated sample data from 1st to 100th trial (100 trials). As an example, we show the

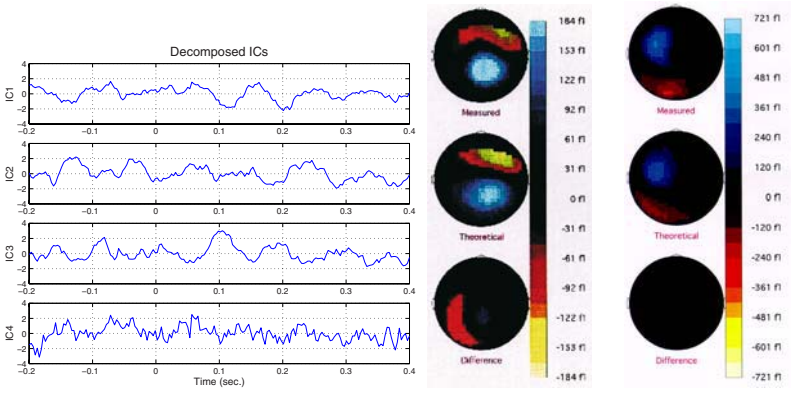


Fig. 2. Results for the first single-trial MEG data. Left: decomposed time-course ICs. Right: source localization of IC1 and IC3.

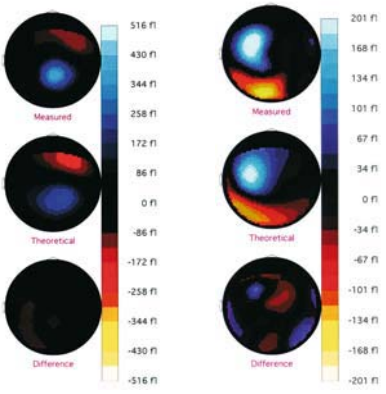


Fig. 3. Evoked responses on the 4th. trial

results of the decomposed IC1 and IC3 maps corresponding to the 4th stimulus in Fig. 3. Comparing these results with the results shown in Fig. 2, we have found that the locations of the IC1 and IC3 are almost similar for both the 1st and 4th stimulus. However, the maximum amplitude of IC1 changes as 184 → 516 fT and IC3 as 721 → 201 fT. This implies that the evoked response IC3 in the left side maybe 'familiar' since the first evoked response is the strongest one in the 100 trials, and the IC1 in the right side maybe 'novel' since the 4th evoked response is the strongest one in 100 trials.

Other phenomena for IC1 and IC3 have been observed in other single-trials. The locations are similar as the 1st and 4th trial but their amplitude are relatively weaker. For example, the alpha-wave component IC2 is the strongest in a trial but IC1 and IC3 are relatively weaker, this may indicate the subject lost his attentions.

3.2 EEG Single-Trial Data Analysis

The EEG data was recorded using NEUROSCAN system with 64 channels in Japan. The auditory evoked potentials experiment was performed on 10 adults whose both ears were stimulated by 1 kHz tone. There were 30 sets of trial data recorded in 324 sec. The stimulus is given at 0, and the time window between -100 ms to 600 ms is defined as a single-trial.

Using the robust pre-whitening with the ICA algorithm, we have analyzed all the ten subjects, each of 30 sets of the single-trial data one by one. As a typical example, the analytical result of subject-2's the 23th single-trial data is shown in Fig.4. In Fig.4, we find that the 3rd component appears with a peak around 100 ms, which is related to an AEPs. With the indication of the scalp map and the frequency 10 (Hz), the 5th component represent an alpha wave. Other components are line noise or eye movement artifacts.

The scalp map of other six different trials is shown in Fig.5. From the scalp maps of the respective single-trials in Fig.5, it is evident that the strength of the

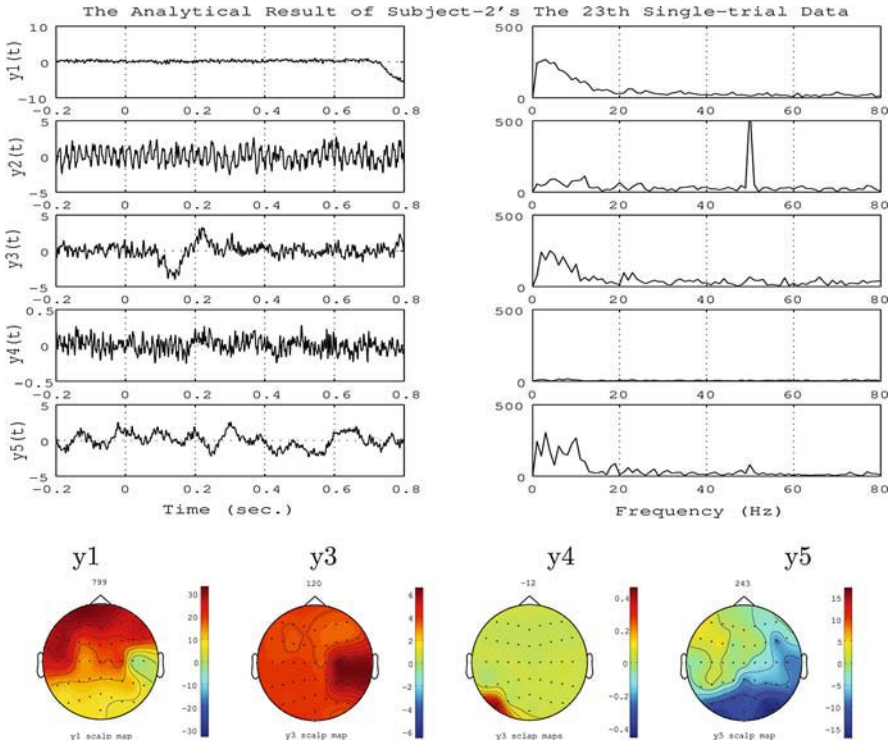


Fig. 4. Result of decomposing the subject 2's the 23th single-trial data, in the time-wise, frequency representations and scalp map representations

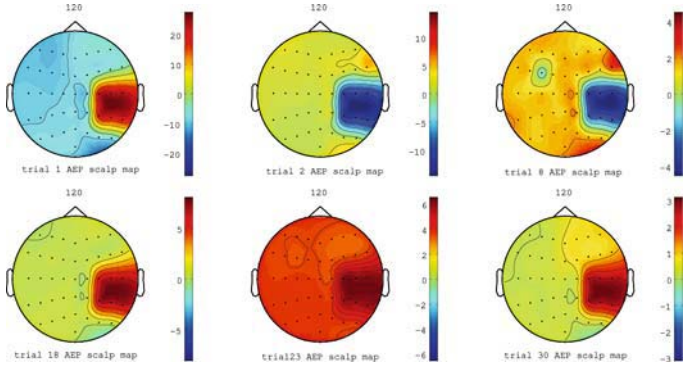


Fig. 5. Scalp maps of the evoked response in the 3rd; 6th; 11th; 19th and 26th trial

evoked responses varies from each other, though the locations of the AEPs appear almost at the same place (on the left side of the brain) for all the cases. Similar to MEG single-trial data analysis, comparing these single-trial maps with the averaged case, we also find that the locations of the evoked responses are almost

the same as the averaged result. It is notable that the trial-by-trial variation of the evoked strength can be visualized when using the proposed method.

When we investigated the relationship between the evoked response component and α-wave component, we observe that the strength of the α-wave varies from trial to trial. Comparing the scalp maps of the α-wave with the AEP maps, we observe that, when the AEP is stronger, the α-wave becomes, in opposite, weaker; inversely, when the AEP becomes weaker, the α-wave is going to be stronger. This may indicate that the subject may have become familiar with the auditory stimulus presentation (i.e habituation) or lost their attention during performing the EEG experiment.

4 Conclusions

This paper presents a robust multi-stage data analysis method for the single-trial MEG and EEG recorded data. Based on this method, some novel features of AEFs or AEPs have been observed : (1) The activity strength of each evoked response is possible to be visualized; (2) The locations of evoked individual responses are correspond to those in the averaged map, and the trial-by-trial variations of the individual evoked response was visualized; (3) Related to the evoked responses, some phenomena such as 'familiar', 'novel' and 'lost attentions', have been observed. The author hope that this kind of data analysis procedure can be useful to study the mechanism of the temporal cortex.

Acknowledgments

This research is partially supported by the Japan Society for the Promotion Science (JSPS) in the Japan-China Research Cooperative Program.

References

1. Cao, J., Murata, N., Amari, S., Cichocki, A., Takeda, T.: Independent Component Analysis for Unaveraged Single-Trial MEG Data Decomposition and Single-Dipole Source Localization. Neurocomputing 49 (2002) 255-277
2. Cao, J., Murata, N., Amari, S., Cichocki, A., Takeda, T.: A Robust Approach to Independent Component Analysis with High-Level Noise Measurements. IEEE Trans. on Neural Networks 14 (3) (2003) 631-645
3. Makeig, S., Bell, A. J., Jung, T-P., Sejnowski, T. J. : Independent Component Analysis of Electroencephalographic Data. In: Touretzky, D.S., Mozer, M.C., Hasselmo, M.E. (eds.): Advances in Neural Information Processing System, Vol. 8. The MIT press, (1996) 145-151
4. Jung, T-P., Makeig, S., Westerfield, M., Townsend, J. Courchesne, E., Sejnowski, T.J. : Independent Component Analysis of Single-Trial Event Related Potentials. Proc. of ICA'99 (1999) 173-179

Multichannel Classification of Single EEG Trials with Independent Component Analysis

Dik Kin Wong, Marcos Perreau Guimaraes, E. Timothy Uy, Logan Grosenick, and Patrick Suppes

Center for Study of Language and Information, Stanford University, CA, USA

Abstract. We have previously shown that classification of single-trial electroencephalographic (EEG) recordings is improved by the use of either a multichannel classifier or the best independent component over a single channel classifier. In this paper, we introduce a classifier that makes explicit use of multiple independent components. Two models are compared. The first ("direct") model uses independent components as time-series inputs, while the second ("indirect") model remixes the components back to the signal space. The direct model resulted in significantly improved classification rates when applied to two experiments using both monopolar and bipolar settings.

1 Introduction

In electroencephalography (EEG), pairs of electrodes are placed on the scalp and the potential difference between them is recorded. There are naturally many different ways to arrange the electrodes on the scalp, of which we have chose the classic 10-20 system [1]. A common electrode reference was used for the monopolar setting, and neighboring electrode pairs were used in a bipolar setting to reduce common-mode noise. In the last section, we summarize the results taking these two different settings into account.

The two experiments considered here were previously reported in [2][3]. In both experiments, sentences about geography were presented to subjects visually, one word at a time for a duration equal to that of matched auditory stimuli (not considered here). Subjects were asked to indicate the truth or falsity of each sentence via keypress. The first experiment had 24 sentences and the second 48. The recordings were done primarily with monopolar settings for the 24-sentence experiment, and with bipolar settings for the 48-sentence experiment. In the 24-sentence experiment, there were 7 subjects with monopolar settings and 3 with bipolar settings. In the 48-sentence experiment, there were 10 subjects with bipolar settings. Each subject saw 10 trials of each sentence.

2 Methods

Independent component analysis (ICA) was designed to estimate the most statistically independent sources from a linear mix of sensor-recorded signals, and

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 541–547, 2006.

assumes that sources are fixed in location and that signal propagation is approximately instantaneous. Based on different approaches to such estimation, algorithms such as FastICA [4], Infomax [5], and SOBI [5] have been proposed. Infomax, which can be implemented as an unsupervised neural network, reduces statistical dependency by maximizing the mutual information between output and input. In addition to the sources $\mathbf{y}$, the algorithm computes the mapping between the sensor recordings and the independent components, called the unmixing matrix $\mathbf{W}$, i.e., $\mathbf{y} = \mathbf{W}\mathbf{x}$. As shown in [6], the update rule for an $N \rightarrow N$ network is $\Delta\mathbf{W} \propto [\mathbf{W}^T]^{-1} + (1 - 2\mathbf{y})\mathbf{x}^T$. We used the EEGLAB implementation of the Infomax algorithm [7] in the current investigation.

Our general methods were as follows. Low-passed filtered trials were downsampled 16 times, then bandpass filtered to remove noise below 1Hz. Trials were split into three sets: training, validation and test. For each sentence, the 10 trials per subject were split 4/3/3 and scaled to between -1 and +1 trial-by-trial. A classification was made on the validation set by selecting the class corresponding to the maximum of the output vector. We minimized the regularized objective function $G_\lambda(\mathbf{w}_i) = \|\mathbf{Z}\mathbf{w}_i - \hat{\mathbf{y}}\|^2 + \lambda^2 \|\mathbf{w}_i\|^2$, with $\mathbf{Z}\mathbf{w}_i$ being the outputs and $\hat{\mathbf{y}}$ the target values.

2.1 Baseline Models

Results for these models were previously published in [3]. By selecting the channel with the best average classification rate, a single-channel classifier (SCC) was defined. The multichannel classifier (MCC) was then defined by choosing the k best channels and concatenating them, yielding longer vectors for the input matrix. In the case of the single ICA-component classifier (ICA-SCC), trials in the training and validation sets were used to estimate the unmixing matrix. The unmixing matrix was then applied to all trials, projecting the trials onto the space of the approximately independent components.

For each permutation, the superset of training and validation data after unmixing was used to compute the matrix, in which each row represented a single trial of the best independent component. The previously reported results of these methods are also included in the results section for the purpose of comparison. We now present the two possible models for implementing the multiple ICA-component classifier (ICA-MCC).

2.2 ICA Direct Model

Procedure for the ICA direct method is:

1. Split the data into two sets and compute unmixing matrix $\mathbf{W}$ based on the first set.
2. Apply $\mathbf{W}$ to unmix all trials in both sets. Each trial can be presented as a matrix $\mathbf{T}$, the multichannel recording, with each row corresponding to a channel $\tilde{\mathbf{T}} = \mathbf{W}\mathbf{T}$.
3. Partition the first unmixed set into two subsets: training and validation.

4. Compute the weight matrix for a regularized linear classifier (SCC) using each component from the training set.
5. Use the weights of the classifier on the validation set to estimate the generalized performance of each component.
6. Repeat steps 3 through 5 three times with different partitioning of the first set to better estimate the components' performance.
7. Sort the components based on their average classification rates on the validation set and then apply MCC classification. Now the best k components are picked and applied to the test set defined in 1. (We fix the regularization parameter $\lambda^2 = 40$ a priori and do not apply validation on k. Instead, we show the results for all k up to the maximum number of components, which is set to be the same as the number of channels.) For each k, a corresponding "indicator" matrix $\mathbf{S}$ is set by having the i^{th} column equal to zero for those channels which are eliminated, and one otherwise. So $\mathbf{S}$ has k non-zero columns. For simplicity, we denote $\mathbf{S}$ with $\mathbf{S}_k$ in subsequent sections.

2.3 ICA Indirect Model

Unlike the direct model, the indirect model involves remixing selected independent components back to the signal space. Given a total number of channels N, and assuming N ICA sources, we eliminated $(N - k_1)$ sources. We then remixed the remaining k_1 sources to form N ICA-cleaned channels. Note that k_1 is a parameter representing the number of ICA sources used. Finally, we applied multichannel classification to these cleaned data, using k_2 channels. It is worth pointing out that this indirect method is closely related to a common artifact cleaning technique often applied to EEG data. In fact, the only difference is that in the common artifact cleaning, e.g. eye-blink removal, the criterion used to decide which sources are eliminated is based on some physical understanding of the nature of the noise. For example, sources resembling eye blinks would be removed based on visual inspection or some computational heuristic. Fig. 1 illustrates the indirect model.

In this case, the decision to eliminate sources depends only on the classification rate for each component on a validation set, and the procedure up to the remixing step is similar to that of the direct model. Assuming a selection matrix $\mathbf{S}_{k_1}$ is determined as in the direct model, the mixing matrix $\mathbf{W^{-1}}$ is used to remix the

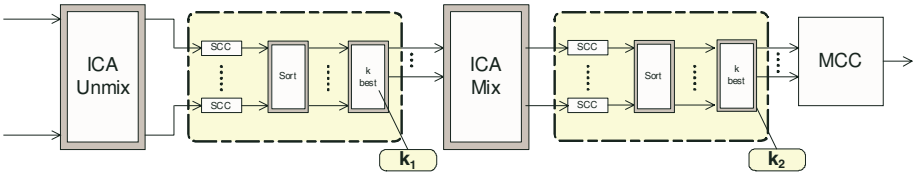


Fig. 1. Illustration (ICA-MCC indirect) of the multichannel classifier based on remixed independent components. The parameter k_1 is the number of ICA sources used and the parameter k_2 is the number of ICA-cleaned channels used.

data back to the signal space. MCC is then used on the ICA-cleaned multichannel data $\mathbf{W}^{-1}\mathbf{S}_{k_1}\mathbf{WT}$, where $\mathbf{T}$ is the original multichannel recording of a trial. We denoted the number of channels used in MCC as k_2.

2.4 Comparison of the Direct and Indirect Models

If we compare the two models in order to choose a scheme for ICA-MCC, we find that the direct model classifiers outperform most of the indirect model classifiers. In Fig. 2, each line of the indirect model corresponds to a specific k_1, and we see that the ICA direct model is consistently better for monopolar subjects. Results are similar for the bipolar setting. It is clear that the direct model is both simpler and more competitive than the indirect model.

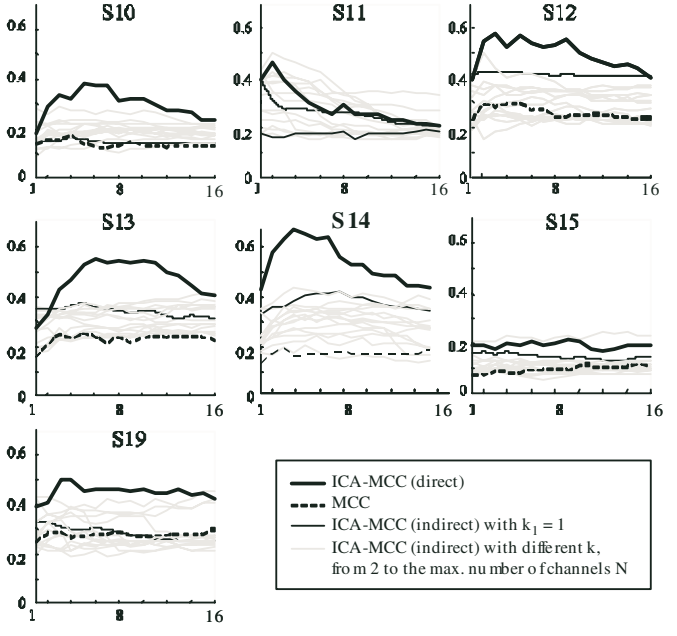


Fig. 2. Comparison of direct and indirect models with different subjects S10-S15 and S19 with monopolar setting. The x-axes are the numbers of channels used and y-axes are the average classification rates achieved on the test sets for 10 permutations.

3 Results

3.1 Scalp Maps

Scalp maps, computed by projecting the appropriate columns of $\mathbf{W}^{-1}$ onto the locations of the electrodes, are commonly used to show the spatial distribution of components. In [3], we showed two scalp maps, one of an obvious eye blink and the other of the best component, for the best monopolar subject S14. In

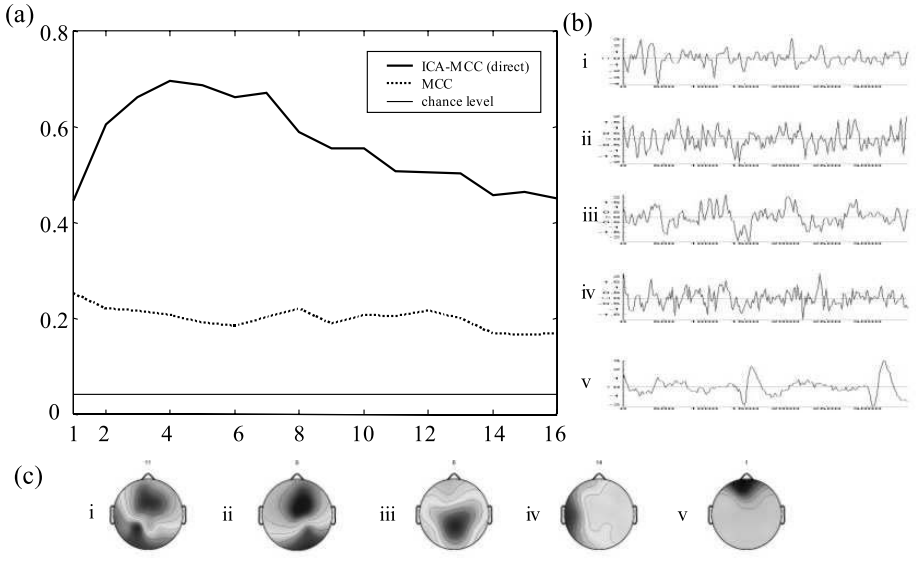


Fig. 3. Classification rates are shown when a different number of channels/components were used for the MCC and ICA-MCC models in (a). The top five components of the best monopolar subject S14 of the ICA-MCC model are shown in (b), and the corresponding scalp maps in (c).

Fig. 3a, we show the classification rates achieved for MCC and ICA-MCC models, with different number of channels or components used. In Fig. 3b, the top five components of ICA-MCC are shown, with (i) the best and (v) the fifth best. In addition, the corresponding scalp maps of these components are shown in Fig. 3c. The eye-blink component which we identified in [3] ranked fifth, while the optimal number of channels k was four, which is indicated in Fig. 3a. Although this parameter k is not validated as rigorously as the other parameters, evidence from a similar experiment suggests that k is robust [8].

From the scalp maps, we can see that the main "sources" come from the temporal regions. We do not attempt to draw any conclusions concerning underlying physiology or the invariance of "sources" across subjects here, as such claims would require techniques we are not currently evaluating, e.g., dipole modelling. However, the scalp maps shown here do provide some evidence against claims that the best components are merely eye-blink artifacts.

3.2 Classification Rates

To summarize our results so far, we expand the table published previously with an additional column corresponding to the new ICA direct model, denoted "ICA-MCC". As the probability distribution of the joint classification results for the 10 permutations cannot be derived without making further assumptions regarding sampling frequency, we conservatively report the results as if they were based on a single permutation. The probability (p-value) of the null hypothesis that

the observed probabilities are at chance level is computed using $P(Y \geqslant k) = 1 - \sum_{j=0}^{k-1} \binom{n}{j} p^j (1-p)^{n-j}$, where n is the number of test trials, k is the number of correct classifications and p the chance probability (note that p is not the same as p-value). The statistical significance of these results is quite remarkable when compared to that of all other methods.

In the case of the monopolar electrodes (Table 1), ICA-MCC outperforms both MCC and ICA-SCC in all cases. The p-values for 6 out of the 7 subjects are less than 10^{-18}, a significance level better than that achieved for the best subject using SCC or MCC. For the bipolar setting (Table 1), ICA-MCC outperforms MCC on 10 of the 13 experimental conditions, with 2 ties and only 1 loss. The results of the best subject (S18) on both experiments are markedly improved. For the 24-sentence experiment, the p-value improves from 10^{-40} to 10^{-55}. For the 48-sentence experiment, from 10^{-65} to 10^{-119}. For the other two subjects of the 24-sentence experiment, the results are changed from 10^{-3} to 10^{-10} and to 10^{-6}. Even the worst subject of the 48-sentence experiment improves from 10^{-1} to 10^{-4}. The achieved classification rates for monopolar and bipolar settings are shown in Fig. 4.

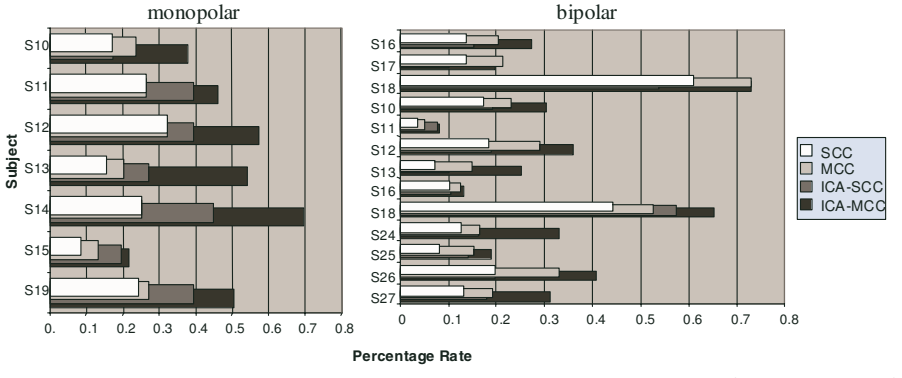


Fig. 4. Percent classification rates of single trials for SCC, MCC, ICA-SCC and ICA-MCC for monopolar (on the left) and bipolar settings (on the right) are shown. All the results shown for the monopolar setting are from the 24-sentence experiment. The results for the bipolar setting are from the 48-sentence experiment, except the first three results (S16,S17,S18) which are from the 24-sentence experiment.

We find that ICA and MCC complement each other well, yielding cleaner sources and enough redundancy to significantly improve classification rates. Regularized-linear methods are clearly effective for single-channel classifications, as they are more effective than the filtered-average-prototype method, and use more efficient computation. Moreover, the family of linear models can be easily extended to multichannel classification (MCC) in a simple manner, yielding significantly improved results for subjects with bipolar settings. In addition, ICA can be used for multichannel classification with a linear combination of the sensor recordings. The improvement on monopolar electrodes is particularly effective

Table 1. Significance levels for the monopolar setting are shown on the left and that for the bipolar setting are on the right. All the results shown for the monopolar setting are from the 24-sentence experiment. The results for the bipolar setting are from the 48-sentence experiment, except the first three results (subjects S16, S17, S18) which are from the 24-sentence experiment.

monopolar

	$p<10^{-n}$			
	SCC	MCC	ICA-SCC	ICA-MCC
	values of -n			
S10	-4	-8	-4	-18
S11	-9	-9	-20	-26
S12	-14	-14	-20	-38
S13	-3	-6	-10	-32
S14	-9	-9	-24	-51
S15	-1	-3	-6	-7
S19	-8	-10	-20	-30

bipolar

	$p<10^{-n}$			
	SCC	MCC	ICA-SCC	ICA-MCC
	values of -n			
S16	-3	-6	-3	-10
S17	-3	-7	-2	-6
S18	-40	-55	-32	-55
S10	-14	-24	-17	-36
S11	-1	-1	-4	-4
S12	-16	-34	-17	-47
S13	-3	-11	-6	-27
S16	-6	-9	-6	-9
S18	-65	-86	-97	-119
S24	-9	-13	-13	-42
S25	-4	-12	-11	-17
S26	-18	-42	-18	-57
S27	-9	-17	-16	-39

using the best component derived with ICA. Finally, combining ICA sources with the multichannel classifier (ICA-MCC) improves all results, with an average improvement of more than 100% for both monopolar and bipolar subjects. The good statistical fits achieved using the ICA-MCC model suggest that it would be useful to develop a more elaborate model that takes into account the actual physical locations of the sensors in the two-dimensional surface covering the scalp.

References

1. Jasper, H.H.: The Ten-twenty Electrode Placement of The International Federation. Electroencephalography and Clinical Neurophysiology **10** (1958) 371–375
2. Suppes, P., Han, B., Epelboim, J., Lu, Z.L.: Invariance Between Subjects of Brain Wave Representations of Language. Proceedings of the National Academy of Sciences **96** (1999) 12953–12958
3. Wong, D.K., Perreau Guimaraes, M., Uy, E.T., Suppes, P.: Classification of Individual Trials Based on The Best Independent Component of EEG-recorded Sentences. Neurocomputing **61** (2004) 479–484
4. Hyvarinen, A.: Independent Component Analysis by Minimization of Mutual Information. Laboratory of Computer and Information Science, Helsinki University of Technology (1997)
5. Belouchrani, A., Abed-Meraim, K., Cardoso, J.F., Moulines, E.: A Blind Source Separation Technique Using Second-order Statistics. IEEE Transactions on Signal Processing **45(2)** (1997) 434–444
6. Bell, A.J., Sejnowski, T.J.: An Information-maximization Approach to Blind Separation And Blind Deconvolution. Neural Computation **7** (1995) 1129–1159
7. Delorme, A., Makeig, S.: EEGLAB: An Open Source Toolbox for Analysis of Single-trial EEG Dynamics. Journal of Neuroscience Methods **134** (2004) 9–21
8. Wong, D.K.: Multichannel Classification of Brain-wave Representations of Language by Perceptron-based Models and Independent Component Analysis. (Ph.D. Dissertation), Stanford University, California, USA (2004)

Application of SVM Framework for Classification of Single Trial EEG

Xiang Liao, Yu Yin, Chaoyi Li, and Dezhong Yao

Center of NeuroInformatics, School of Life Science and Technology,
University of Electronic Science and Technology of China,
Chengdu 610054, P.R. China
dyao@uestc.edu.cn

Abstract. A brain-computer interface (BCI) system requires effective online processing of electroencephalogram (EEG) signals for real-time classification of continuous brain activity. In this paper, based on support vector machines (SVM), we present a framework for single trial online classification of imaginary left and right hand movements. For classification of motor imagery, the time-frequency information is extracted from two frequency bands (μ and β rhythms) of EEG data with Morlet wavelets, and the SVM framework is used for accumulation of the discrimination evidence over time to infer user's unknown motor intention. This algorithm improved the single trial online classification accuracy as well as stability, and achieved a low classification error rate of 10%.

1 Introduction

The online analysis of single trial electroencephalogram (EEG) measurements is a challenge for signal processing and machine learning [1]. Once this complex and non-stationary signal can be reliably processed, we can make use of the brain activities for establishing a communication channel from a human to an output device such as a computer application, like a brain-computer interface (BCI) system.

It is well known that a so-called idle rhythms that are attenuated when engagement with the respective limbs activity takes place (actual, or just imagined), and it can be measured in EEG as a brain rhythm around 10 Hz (μ rhythm), and usually associated with 20 Hz (β rhythm). These two rhythms are both focused over somatosensory or motor cortex, so they can be distinguished from each other by location, frequency or relationship to concurrent motor activity. As the attenuation effect is due to loss of synchrony in the neural populations, it is labeled event-related desynchronization (ERD). In opposite, an increased rhythmic activity is called event-related synchronization (ERS) occurred after movement and with relaxation [2].

In this work, we utilize these well known neurophysiological phenomena (ERD/ERS), and extracted the rhythmic information of the EEG raw data for analysis in time-frequency domain by means of Morlet wavelets. Though some classification methods, e.g. probabilistic models [3], have been successfully used for online analysis in BCI, the support vector machines (SVM) was known as a powerful classifier that may yield very good results using only simple and redundant features. Therefore, we

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 548 – 553, 2006.

use a SVM framework for the discrimination evidence accumulation over time, which in order to improve the single trial online classification accuracy and stability, then a continuous classification output of an unknown single trial will be derived for online analysis (e.g. as a part of feedback systems).

2 Methods

2.1 Preliminary Analysis

The data is provided by Graz BCI group in BCI Competition 2003 [4]. This data set is single trial EEG recorded during left and right hand imaginary movements.

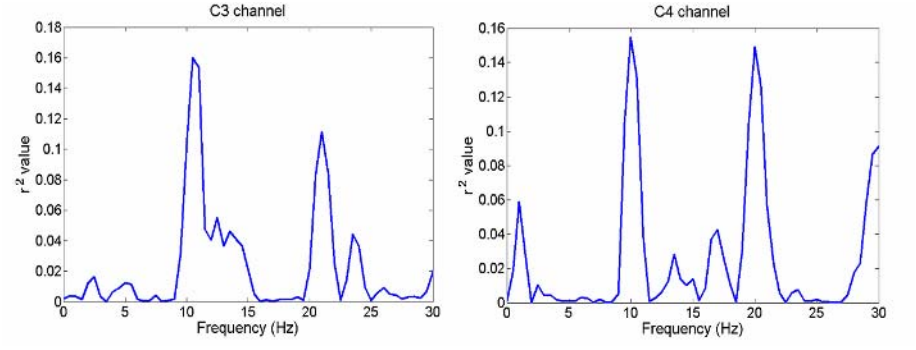


Fig. 1. r^2 values across the spectrum in C3 and C4 channels for discrimination of the two classes, the larger r^2 values indicate the more discriminative power bands

For channels C3 and C4, we estimated the power spectrums over all trials in the training data set individually for the left and right classes by means of Welch method, then the individual channel r^2 values [5] corresponding to the discrimination of these two classes were calculated (Fig.1). The r^2 values reflect the proportion of the variance of the power spectral values accounted for by the class information. Here we kept two discriminative frequency bands, which corresponding to peaks of r^2 values, roughly in 10~11 Hz (μ rhythm) and 20~22 Hz (β rhythm).

2.2 Feature Extraction

In our study, we use wavelet analysis for extraction of the time-frequency information [6]. The continuous wavelet transform (CWT) $W(s,\tau)$ of a signal $z(t)$ is computed according to the formula:

$$W(s,\tau) = \frac{1}{\sqrt{s}} \int_{-\infty}^{\infty} z(t)\psi^*(\frac{t-\tau}{s})dt \tag{1}$$

where τ denotes the time shift, s is the scale, t is the time variable, * means complex conjugation and ψ is the wavelet function. Compare with a real wavelet function, a

complex wavelet function will return information about both amplitude and phase that is better adapted for capturing oscillatory behavior. Hence, we use the complex Morlet wavelets that are well localized in the frequency domain

$$\psi_{\tau,s}(t) = \frac{1}{\sqrt{s}}\pi^{-1/4} e^{(i\omega_0 \frac{t-\tau}{s})} e^{-\frac{1}{2}(\frac{t-\tau}{s})^2} \tag{2}$$

where ω_0 is the eigenfrequency. The wavelet power spectrum of a signal $z(t)$ is the square norm of its convolution with the scaled and shifted wavelet

$$\left|W(s,\tau)\right|^2 = \left|\int z(t)\psi^*_{\tau,s}(t)dt\right|^2 \tag{3}$$

where $\left|W(s,\tau)\right|^2$ is the wavelet power spectrum in scale s at time τ, $\left|W(s_i,\tau)\right|^2$ denotes the wavelet power spectrum over *i*th scale. For online processing permits only causal filtering and the Morlet wavelets are unbounded time symmetric filters centered on the time-point under study [3], here we shift the wavelets backward in the time domain and calculate the convolution of the signal with the causal wavelets.

To examine fluctuations in power over a range of scales (a frequency band), one can define the scale-averaged wavelet power as the weighted sum of the wavelet power spectrum over scales *s*1 to *s*2. For two frequency bands 10~11Hz and 20~22Hz, we calculated the wavelet power over a range of scales individually, then results in two scale-averaged wavelet power in individual channel

$$\overline{W}(\tau) = \frac{\delta j \delta t}{C_\delta} \sum_{j=j_1}^{j_2} \frac{\left|W(s_j,\tau)\right|^2}{s_j} \tag{4}$$

where δj is the factor for scale averaging, δt is the time step in a time series, and C_δ is a reconstruction factor for each wavelet function.

The resulting scale-averaged wavelet power for the two frequency bands (μ and β rhythms) at the two channels (C3 and C4) are then concatenated in order to construct a feature vector at time point t

$$x(t) = (\overline{W}_\mu^{C3}(t), \overline{W}_\mu^{C4}(t), \overline{W}_\beta^{C3}(t), \overline{W}_\beta^{C4}(t))^T \tag{5}$$

These features represent activities over ipsilateral and contralateral motor cortex area.

2.3 SVM Framework

SVM has been successfully applied for classification in various domains, its basic idea is to map data into a high dimensional space and find a separating hyperplane with the maximal margin [7]. Given training vectors x_i of l examples and the corresponding class labels y_i, $(x_1, y_1),\ldots,(x_l, y_l) \in R^N \times \{-1,1\}$. For any testing case x, the non-linear discriminant function is

$$f(x) = \sum_i^{N_S} y_i \alpha_i K(x, x_i) + b \tag{6}$$

where N_s is the number of resulting support vectors, α_i is the positive Lagrangian multipliers, $K(x, x_i)$ is kernel function and b is the bias term. Here we used the Gaussian kernel for SVM realizations

$$K(x, x_i) = \exp(-\gamma \|x - x_i\|^2) \tag{7}$$

where γ is the Gaussian kernel parameter. For optimal hyperplane passing through the origin, there is an alternative bound on the actual risk of support vector machines

$$E[P(error)] \leq \frac{E[SV]}{l} \tag{8}$$

where E [P(error)] is the expectation of the actual risk over all choices of training set of size $l-1$, and E[SV] is the expectation of the number of support vectors (SV) over all choices of training sets of size l [7].

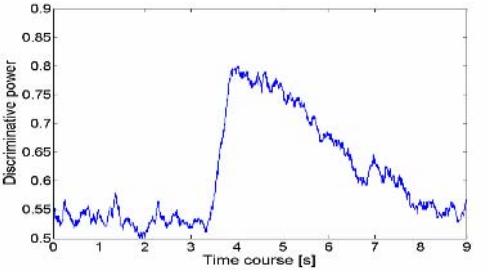


Fig. 2. Time course of discriminative power, the values indicate the estimated accuracy

According to (8), the discriminative power w_t at time point t can be defined as

$$w_t = 1 - \frac{E(SV)}{l} \tag{9}$$

Therefore, w_t can be considered as an estimation of the test accuracy at time point t. Fig.2 is the time course of the w_t on training data set. As indicated by the maximum of w_t, the most discriminative information occurred around 4s.

We regard the values of the discriminant function $f(x)$ in (6) as a time-varying signed distance function (TSD), the absolute magnitude of the value indicates the confidence of the classification. In order to improve the online classification performance, we use a SVM temporal combination framework to incorporate knowledge from preceding time points at a certain time as an evidence accumulation process. The SVM framework is defined as combining the discriminant function $f(x)$ with corresponding discriminative power w_t at each time point

$$f_{t2}(x) = \frac{\sum_{t=t1}^{t2} {w_t}^n f_t(x)}{\sum_{t=t1}^{t2} {w_t}^n} \tag{10}$$

where $f_t(x)$ denotes the time-varying signed distance function at time point t, w_t^n denotes an integer power of w_t, so we can change the value of n for adjusting the effect of discriminative power, here taken to be 2 as an empirical value.

3 Results

As a model selection procedure, the values of the two SVM parameters (the regularization parameter C and Gaussian kernel parameter γ) were estimated via 5-fold cross-validation (CV) with the whole training set.

For comparing different methods, mutual information (MI) [8] was used to reflect the confidence of the classification results. So the resulting time courses for both the classification error and the mutual information on the test data are presented in Fig. 3.

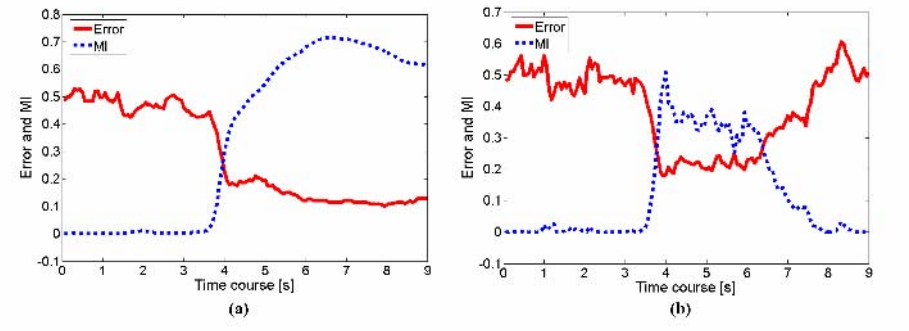


Fig. 3. Time courses of the classification error (solid line) and mutual information (dot line) for (a) classification with the SVM framework and (b) classification without the SVM framework

In Fig. 3(a), the classification is made by chance during the first 4 s. But after 4 s, a noticeable improvement can be observed in both the raising MI and the decreasing classification error. While the discriminative power begin to gradually decrease after 5 s (Fig.2), which indicates fading discriminative ability of the classifier, the full model still gains useful discrimination information via the evidence accumulation process. Finally, the maximum mutual information of 0.71 bits occurs at 6.5 s that reflecting the peak decision confidence and the minimum error of 10% is achieved at 7.8 s. In Fig.3 (b), the classification error was kept around 20% only from 4s to 6s and restarted to increase after that time, and the MI reflect the similar results. Hence the classification accuracy and stability were effectively improved by the framework.

4 Conclusion

In this work, we proposed a method of SVM-based framework for single trial online classification of imaginary hand movements, which enables us to achieve high discrimination accuracies and stability. Therefore, our algorithm seems promising for

a practical realization within an online classification scenario. More generally, this SVM framework is also applicable to other kinds of online classification tasks that discriminate between two classes of such signals, just by adapting the feature extraction procedure. EEG signals that recorded for a brain-computer interface system are generally very noisy, non-stationary, contaminated with artifacts (e.g. EMG and EOG) and usually different from subject to subject that can strongly distort the classifier performance, result in bad generalization. For a continuous adaptation of these characteristics of EEG signal, a specially adapted SVM algorithm applied with this framework would probably against such distortions and lead to a significant improvement in the performance of the brain state classifier, which will be investigated in future work. Our future research will also focus on the combination of this SVM framework with some other robust signal processing methods for single trial EEG analysis, which in order to improve both the classification accuracy and the generalization ability.

Acknowledgement

The authors would like to acknowledge the Graz BCI group, Austria for sharing their data set. This work was supported in party by the 973 Project No.2003CB716106 and NSFC No. 90208003, 60571019.

References

1. Blankertz, B., Curio, G., Müller, K.-R.: Classifying Single Trial EEG: Toward Brain Computer Interfacing. Advances in Neural Inf. Proc. Systems, 14 (2002) 157–164
2. Pfurtscheller, G., Silva, F.H.L.da: Event-Related EEG/MEG Synchronization and Desynchronization: Basic Principles. Clin.Neurophys 110 (1999) 1842-1857
3. Lemm, S., Schäfer, C., Curio, G.: BCI Competition 2003—Data Set III: Probabilistic Modeling of Sensorimotor μ Rhythms for Classification of Imaginary Hand Movements. IEEE Trans. Biomed. Eng. 51 (6) (2004) 1077–1080
4. Blankertz, B., Müller, K.-R., Curio, G., Vaughan, T.M., Schalk, G., Wolpaw, J.R., Schloegl, A., Neuper, C., Pfurtscheller, G., Hinterberger, T., Schröder, M., Birbaumer, N.: The BCI Competition 2003: Progress and Perspectives in Detection and Discrimination of EEG Single Trials. IEEE Trans. Biomed. Eng.51 (6) (2004) 1044–1051
5. Wolpaw, J.R., Birbaumer, N., McFarland, D.J., Pfurtscheller, G., Vaughan, T.M.: Brain-Computer Interfaces for Communication and Control. Clin.Neurophys 113 (2002) 767–791
6. Torrence, C., Compo, G.: A Practical Guide to Wavelet Analysis. Bull. Amer. Meterol. 79 (1) (1998) 61–78
7. Burges, C. J. C.: A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining Knowledge Discovery 2 (1998) 121–167
8. Schlögl, A., Neuper, C., Pfurtscheller, G.: Estimating The Mutual Information of An EEG-based Brain-Computer Interface. Biomed. Technik. 47 (1-2) (2002) 3–8

Normal and Hypoxia EEG Recognition Based on a Chaotic Olfactory Model

Meng Hu[1], Jiaojie Li[2], Guang Li[3], Xiaowei Tang[1], and Walter J. Freeman[4]

[1] Department of Physics, Zhejiang University, Hangzhou 310027, China
[2] Hangzhou Sanitarium of PLA Airforce, Hangzhou 310013, China
[3] National Laboratory of Industrial Control Technology,
Institute of Advanced Process Control, Zhejiang University, Hangzhou 310027, China
guangli@cbeis.zju.edu.cn
[4] Division of Neurobiology, University of California at Berkeley,
LSA 142, Berkeley, CA, 94720-3200, USA

Abstract. The KIII model of the chaotic dynamics of the olfactory system was designed to simulate pattern classification required for odor perception. It was evaluated by simulating the patterns of action potentials and EEG waveforms observed in electrophysiological experiments. It differs from conventional artificial neural networks in relying on a landscape of chaotic attractors for its memory system and on a high-dimensional trajectory in state space for virtually instantaneous access to any low-dimensional attractor. Here we adapted this novel neural network as a diagnostic tool to classify normal and hypoxic EEGs.

1 Introduction

Biological neural systems are complex but rapid and reliable in pattern classification. We follow the architecture of the olfactory system to construct a high dimensional chaotic network, the KIII model, in which the interactions of globally connected nodes are shaped by reinforcement learning to support a global landscape of high-dimensional chaotic attractors. Each low-dimensional local basin of attraction corresponds to a learned class of stimulus patterns. Convergence to an attractor constitutes abstraction and generalization from an example to the class. KIII model has performed well on several complex pattern recognition tasks [1], [2], [3].

Here we present a new application of the KIII network for recognition of normal and hypoxic EEGs based on the feature vectors of 30-60 Hz sub-band wavelet packet tree coefficients constructed using wavelet packet decomposition.

2 KIII Model Description

Biologically, the central olfactory neural system is composed of olfactory bulb (OB), anterior nucleus (AON) and prepyriform cortex (PC). In accordance with the anatomic architecture, KIII network is a multi-layer neural network model, which is composed of several K0, KI, KII units [4]. Among the models, every node is described by a second order differential equation. The parameters in the KIII network are optimized to fulfill some criteria that were deduced in electrophysiological experiments [5].

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 554 – 559, 2006.

In the KIII network, Gaussian noise is introduced to simulate the peripheral and central biological noise source, respectively; the peripheral noise is rectified to simulate the excitatory action of input axons. The additive noise eliminates numerical instability of the KIII model, and makes the system trajectory stable and robust under statistical measures. Because of this kind of stochastic chaos, the KIII network can approximate the real biological intelligence for pattern recognition [6].

3 Application to Normal and Hypoxic EEG Recognition

3.1 Data Acquisition

A mixture of nitrogen and oxygen at normal atmosphere pressure was used to simulate different altitudes in the atmosphere by adjusting oxygen partial pressure: It was provided to subjects via a pilot mask. In the first day, when the subjects stayed at normal atmosphere, they were tested for auditory digit span and serial addition/subtraction tests while the EEGs were recorded. In the second day, after the subjects stayed at environment simulating 3500 m altitude for 25 minutes, they repeated the aforementioned test procedure. The experiments were carried out during the same time period each day. Five healthy male volunteers around 22 years old were taken as subjects. Immediately after the behavioral evaluations 1.5 seconds EEGs were recorded for analysis under both normal oxygen partial pressure and 3500 m altitude. EEG data were taken from 30 Channels including: FP1, FP2, F7, F3, FZ, F4, F8, FT7, FC3, FCZ, FC4, FT8, T3, C3, CZ, C4, T4, TP7, CP3, CPZ, CP4, TP8, T5, P3, PZ, P4, T6, O1, OZ and O2 (10/20 system). The reference was (A1+A2)/2 (A1 = left mastoid, A2 = right mastoid). The EEG amplifier used was NuAmps Digital Amplifier (Model 7181) purchased from Neuroscan Compumedics Limited, Texas, USA. Sampling rate was 250 S/s. All values are in μVolt.

3.2 Evaluation of the Severity of the Effects of Hypoxia by NE Testing

NE is a sensitive and reliable tool for early detection of adverse effects of the environmental hazards on central nervous system. In the normal and simulating 3500m altitude experiments, auditory digit span and serial addition and subtraction were utilized to evaluate the degree of hypoxia. The result of the test is shown in Table 1.

Table 1. Performance of NES under normal and hypoxia states

Subject	Auditory Digit Span Scores		Serial Addition And Subtraction Scores	
	Normal	Hypoxia	Normal	Hypoxia
1	30	29	28	21
2	28	24	31	16
3	25	21	20	20
4	32	23	24	15
5	19	9	18	12

T-tests were performed on the NE under normal and hypoxia conditions. As a result, the NE scores of normal and hypoxia were different observably ($p<0.05$). Furthermore, the scores under the hypoxia condition were lower distinctly, which means that subjects' behavior capability became weaker under the hypoxia condition.

3.3 Feature Vector Extraction

By wavelet packet decomposition, the original waveform can be reconstructed from a set of analysis coefficients that capture all of the time (or space) and frequency information in the waveform [7]. In our analysis, we use the COIF5 wavelet. The number of levels of decomposition is chosen as two and wavelet packet tree coefficients of a 30-60Hz sub-band are abstracted. The feature vector is a 30-dimensions vector due to 30 EEG channels. For each channel, the square of the wavelet packet tree coefficients are summed up as one dimension of the feature vector. According to the topology of the EEG channel, feature vectors can be transformed as a feature topography. A typical feature topography sample of comparing normal and hypoxic EEGs collected from the same subject is illustrated in Fig. 1.

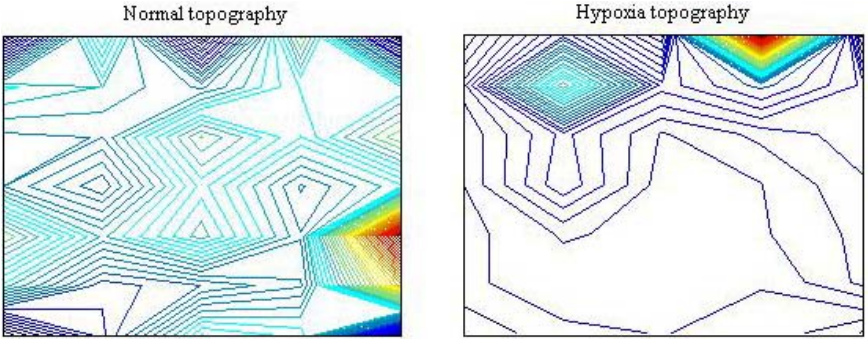


Fig. 1. A feature vector topography of the normal and hypoxia EEG

3.4 Learning Rule

There are two main learning processes: Hebbian associative learning and habituation. Hebbian reinforcement learning is used for establishing the memory basins of certain patterns, while habituation is used to reduce the impact of environment noise or those non-informative signals input to the KIII network. The output of the KIII network at the mitral level (M) is taken as the activity measure of the system. The activity of the *i*th channel is represented by SD_{ai}, which is the mean standard deviation of the output of the *i*th mitral node (Mi) over the period of the presentation of input patterns, as Eq.(1). The response period with input patterns is divided into equal segments, and the standard deviation of the *i*th segment is calculated as SD_{aik}, SD_{ai} is the mean value of these *S* segments. SD_a^m is the mean activity measure over the whole OB layer with *n* nodes (Eq.(2)).

$$SD_{ai} = \frac{1}{S}\sum_{k=1}^{s} SD_{aik} \quad . \tag{1}$$

$$SD_a^m = \frac{1}{n}\sum_{k=1}^{n} SD_{ai} \quad . \tag{2}$$

The modified Hebbian rule holds that each pair of M nodes that are co-activated by the stimulus have their lateral connections $W(mml)_{ij}$ strengthened. Here $W(mml)_{ij}$ stands for the connection weights both from *Mi* to *Mj* and from *Mi* to *Mj*. Those nodes whose activities are larger than the mean activity of the OB layer are considered activated; those whose activity levels are less than the mean are considered not to be activated. Also, to avoid the saturation of the weight space, a bias coefficient K is defined in the modified Hebbian learning rule, as in Eq.(3). $W(mml)_{ij}$ is multiplied by a coefficient r (r>1) to represent the Hebbian reinforcement.

$$\begin{aligned} &IF && SD_{ai} > (1+K)SD_a^m \; and \quad SD_{aj} > (1+K)SD_a^m \\ &THEN && W(mml)_{ij} = W(mml)^{high} \; and \quad W(mml)_{ji} = W(mml)^{high} \quad . \\ &OR && W(mml)_{ij} = r \times W(mml)_{ij} \; and \quad W(mml)_{ji} = r \times W(mml)_{ji} \quad . \end{aligned} \tag{3}$$

Two algorithms to increase the connection weight are presented, algorithm 1 is used to set the value to a fixed high value $W(mml)^{high}$ as in previous references and algorithm 2 is a new algorithm that will multiply an increasing rate to the original value. For the habituation learning, a continuous habituation strategy is used, which means that habituation occurs cumulatively at each time step [1]. At the end of a training session for two types of learning, the connection weights are fixed to perform pattern classification tests. During training, several samples for each class are given. The activity measure vector for every trail is calculated, and the mean activity of those trails, which belong to one class, is defined as the cluster center of that class. Inputs of different classes that the system is trained to discriminate form multiple clusters, each with its center of gravity. When a test pattern is given, the Euclidean distances from the corresponding point to those training pattern cluster centers are calculated, and the minimum distance to a center determines the classification. On the other hand, the concept of classification threshold was introduced into the classification rule. If the difference between the minimum Euclidean distance and the secondary minimum distance is less than the threshold value, it is regarded as a recognition failure.

3.5 Classification of Normal and Hypoxia EEG

We use the KIII model to distinguish hypoxic from normal EEGs. The KIII model learns the desired patterns --- the normal and hypoxic EEG patterns for three times in turn. The test data set contains 80 samples of normal and hypoxic EEG for individuals by 5 different subjects. We chose 40 samples in the odd position for training and used all the 80 samples for classification, and then we chose 40 samples in the even position for training and used all the 80 samples for classification. The final correction

rate is from the mean of twice correction rate. In this application, a 30-channel KIII network is used with system parameters as the reference [5].

The experimental results are shown in Fig. 2. Effectively, the mean of classification rate for test data set is equal to 92%. Hypoxic EEGs can be distinguished from normal EEG by the KIII network. In other words, a new pattern in EEG, which is different from normal one, comes into being, induced by hypoxia. The conclusion of EEG classification is consonant with NE results.

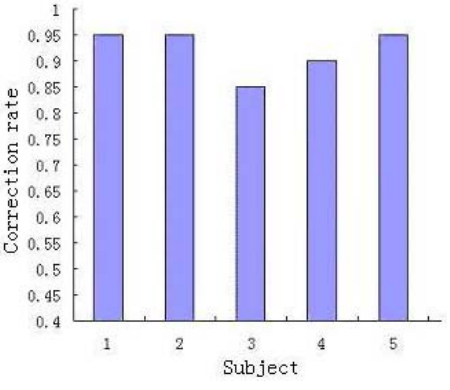


Fig. 2. Rate of correct classification of normal and hypoxic EEGs by the KIII network

4 Discussion

The lowest correction rate (85%) was observed from the subject No.3. When we compared this with the NE test results, we found that the auditory digit span scores of the subject No.3 changed less than other subjects under normal and hypoxic conditions while the scores of serial addition and subtraction remain unchanged. The results of EEG analysis and NE conformed to indicate that the effect of hypoxia on subject No.3 was less than on other subjects. To some extend, rate of correct EEG pattern classification using KIII network represents the degree of hypoxia. It provides the possibility to measure the effect of hypoxia quantitatively.

Derived directly from the biological neural system, the KIII network is more complicated yet more effective in simulating the biological neural system in comparison with conventional ANN. The KIII model has good capability for pattern recognition as a form of the biological intelligence. It needs much fewer learning trials when solving problems of pattern recognition. In our study, it is extremely time consuming to use a digital computer to solve the numerous differential equations within KIII model. This problem restricts the application of the KIII network in real time. The implementation of the KIII in analog VLSI [8] is surely a promising research for building more intelligent and powerful artificial neural network. By providing feature vectors for classification, the EEG might be made to serve as a quantitative indicator of hypoxia in real time, which might significantly improve the safety of those who work in high altitude.

Acknowledgements

This research is supported by the National Natural Science Foundation of China (No. 60421002) and the National Basic Research Program of China (973 Program, No. 2004CB720302).

References

1. Kozma, R., Freeman, W. J.: Chaotic Resonance - Methods and Applications for Robust Classification of Noisy and Variable Patterns. Int. J. Bifur. Chaos. 11(6) (2001) 1607–1629
2. Li, G., Lou, Z., Wang, L., Li, X. and Freeman, W. J.: Application of Chaotic Neural Model Based on Olfactory System on Pattern Recognitions. In: Wang, L., Chen, K., Ong Y.S. (eds.): Advances in Natural Computation. Lecture Notes in Computer Science, Vol. 3610. Springer-Verlag, Berlin Heidelberg New York (2005) 378 – 381
3. Quarder, S., Claussnitzer, U., Otto, M.: Using Singular-Value Decompositions to Classify Spatial Patterns Generated by a Nonlinear Dynamic Model of the Olfactory System. Chemometrics and Intelligent Laboratory Systems. 59 (2001) 45–51
4. Chang, H. J., Freeman, W. J.: Optimization of Olfactory Model in Software to Give 1□f Power Spectra Reveals Numerical Instabilities in Solutions Governed by Aperiodic (Chaotic) Attractors. Neural Networks. 11 (1998) 449–466
5. Chang, H. J., Freeman, W. J., Burke, B. C.: Biologically Modeled Noise Stabilizing Neurodynamics for Pattern Recognition. Int. J. Bifur. and Chaos 8(2) (1998) 321–345
6. Freeman, W. J., Chang, H. J., Burke, B. C., Rose, P. A., Badler, J.: Taming Chaos: Stabilization of Aperiodic Attractors by Noise. IEEE Trans. Circuits and Syst. 44 (1997) 989–996
7. Quinquis, A.: A Few Practical Applications of Wavelet Packets. Digital Signal Processing. 8 (1998) 49–60
8. Principe, J. C., Tavares, V. G., Harris, J. G., Freeman, W. J.: Design and Implementation of a Biologically Realistic Olfactory Cortex in Analog VLSI. Proceedings of the IEEE. 89(7) (2001) 1030–1051

Nonlinear Dynamics of EEG Signal Based on Coupled Network Lattice Model

Minfen Shen[1], Guoliang Chang[2], Shuwang Wang[2], and Patch J. Beadle[3]

[1] College of Information Engineering, Guangdong University of Technology, Guangzhou, China
mfshen57@vip.163.com
[2] Key Lab. of Image Processing, Shantou University, Guangdong, China
[3] School of System Engineering, Portsmouth University, Portsmouth, U.K.

Abstract. EEG signals were expressed as the typical non-stationary signal. More and more evidences were found that both EEG and ERP signals are also chaotic signal from the nonlinear dynamics system. A novel model based on the time-varying coupled map lattice model is proposed for investigating the nonlinear dynamics of EEG under specified cognitive tasks. Moreover, the time-variant largest Lyapunov exponent (LLE) is defined for the purpose of defining quantitative parameters to reveal the global characters of system and extract new information involved in the system. Both simulations and real ERP signals were examined in terms of LLE parameter for studying the signal's dynamic structure. Several experimental results show that the brain chaos changes with time under different attention tasks of the information processing. The influence of the LLE with the different attention tasks occurs in P2 period.

1 Introduction

EEG signal has been proved to be complicated signal produced by typical nonlinear dynamics process [1-2]. It is the analysis of nonlinear dynamics became an important tool that put the observed quantities into the whole variable space of a system, and help to find the inherent dynamics characteristics, which is widely used in feature extraction, pattern recognition, dynamic detection and so on [3-5]. For many real physics systems, including nervous system, they are spatial-temporal nonlinear system which reveals the nonlinearity not only in time dimension but also in space dimension [6]. But the spatial effects were always ignored in the traditional nonlinear dynamics analysis [7-8]. In this paper, we propose a coupled map lattice (CML) model to describe the spatial-temporal nonlinear dynamic system, and some measurements are defined to effectively describe the nonlinear dynamics characters of signals. Furthermore, since the changes of physiochemistry, the oxygen saturation of blood and the conductibility in brain happen momentarily, these changes will significantly influence the system. Thus, the time-variant factors should be considered fro our purpose.

To investigate the inherent nonlinear characteristics of EEG signals, a new algorithm via time-variant coupled map lattice model is proposed for studying the nonlinear dynamics of EEG signals. A new model combining the CML model with neural networks is also developed, which has universal approximation capability. The

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 560 – 565, 2006.

dynamic parameter of EEG signals under different brain functional states was estimated. Several experimental results are compared and explained.

2 Basic Scheme Based on CML

CML is regarded as a dynamic system with discrete time and space and continuous state. CML has the following form [5]:

$$x_{n+1}(i,j) = f\left(\sum_{m=-K}^{K} \sum_{n=-K}^{K} \varepsilon(i+m, j+n) x_n(i+m, j+n) - \varepsilon(i,j) x_n(i,j) \right) \tag{1}$$

where f denotes the local nonlinear map and (i, j) is the coordinate of lattice. K represents the number of lattice points. ε denotes the spatial coupling coefficient which measures the degree of interrelationship between current lattice and associated lattice given as

$$\sum_{m=-K}^{K} \sum_{n=-K}^{K} \varepsilon(i+m, j+n) - \varepsilon(i,j) = 1 \tag{2}$$

For the different local map, both Logistic-CML and RBF-CML models are established as the following relationship:

$$x_{n+1}(i,j) = \mu x_n^{'}(i,j)\left(1 - x_n^{'}(i,j)\right) \tag{3}$$

$$x_{n+1}(i,j) = \sum_{l=1}^{L} W(l) \exp\left(\frac{-\left\| x_n^{'}(i,j) - a(l) \right\|^2}{s} \right) \tag{4}$$

where $x_n^{'}(i,j) = \sum_{m=-K}^{K} \sum_{n=-K}^{K} \varepsilon(m,n) x_n(i+m, j+n) - \varepsilon(i,j) x_n(i,j)$. μ is the map coefficient, while L denotes the number of base functions. s is a parameter indicating the width of base function and a represents the center point of base function. Both s and L are selected according to the experience. a is determined by averagely selecting the value between maximum and minimum value of measured signals. To obtain the model, the problem of the following optimizing procedure needs to be considered

$$\min_{\mu} \left\| x(i,j) - \hat{x}(i,j) \right\| \tag{5}$$

$$\min_{W} \left\| x(i,j) - \hat{x}(i,j) \right\| \tag{6}$$

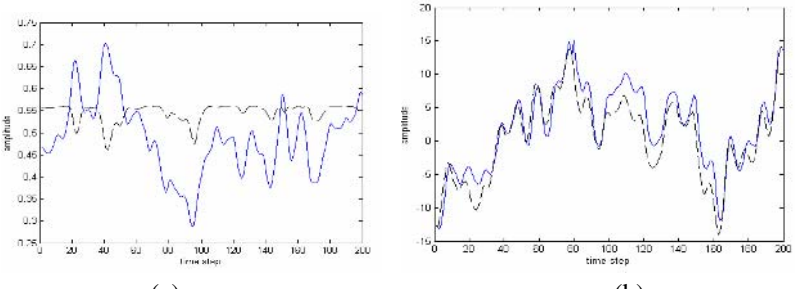
where $x(i,j)=(x_1(i,j),\ldots,x_n(i,j))$ represents the real signal and $\hat{x}(i,j)=(\hat{x}_1(i,j),\ldots,\hat{x}_n(i,j))$ is the predicted signal. Thus, both μ and W can be determined. To build a time-variant model, usually the map relationship between μ and measured signals was found [9]. However, we cannot find a one-to-one function to fit the curve of μ. As we know, theoretically, RBF can represent arbitrary function and has the universal approximation capability. So RBF is employed to capture the changes of map coefficient. Thus, the time-variant model with universal approximation capability can be defined:

$$x_{n+1}(i,j)=\sum_{l=1}^{L} w(l)\exp\left(\frac{-\|\mu(n,i,j)-a(l)\|^2}{s}\right)x_n'(i,j)\left(1-x_n'(i,j)\right) \tag{7}$$

where $\mu(n,i,j)$ is the map coefficient in time n at coordinate (i,j).

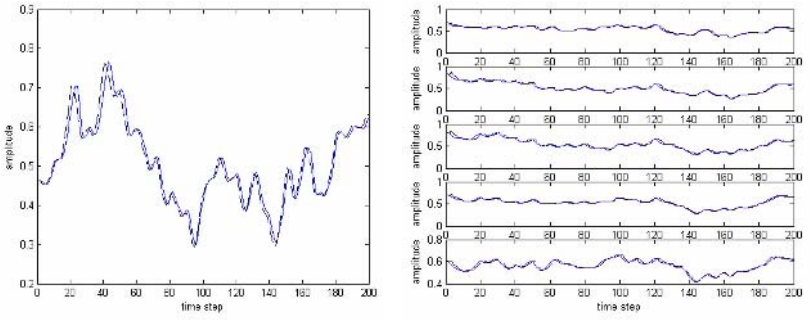
3 Experimental Results

A segment of real spontaneous EEG signal with 1000 data points was used to evaluate the performance of the proposed model. The first 800 data points were used to train the model and the last 200 data were used to test. The scale of the grid is designed as 5×5 and the terminal condition was assumed to be periodic. The correlation coefficient was adopted to measure the degree of likelihood between the measured data and the predicted data. Fig.1 shows the fitness of 2-D time-invariant model with Logistic-CML model and RBF-CML model, respectively. In addition, the mean correlation coefficients between the real data and the predictive data in every lattice for both 2-D time-invariant Logistic-CML model and the RBF-CML model are estimated as 0.17131 and 0.91123, respectively.



(a) (b)

Fig. 1. The fitness for the 2-D time-invariant model. The blue lines represent the real data and the black lines represent the predictive data. (a) The fitness in one lattice of 2-D time-invariant Logistic-CML model. (b) The fitness in one lattice of 2-D time-invariant RBF-CML model.



(a) (b)

Fig. 2. The fitness of the spontaneous EEG data in one lattice of 2-D time-variant Logistic-CML model in which the blue lines represent the real data and the black lines denote the predictive data. (a) The fitness result in one lattice. (b) The fitness result for every lattice of the model.

Based on the experimental results, it can be seen that both 2-D time-invariant Logistic-CML and RBF-CML model cannot give the optimal resolution for the time-varying systems. To deal with the time-varying mapping relationship, the 2-D time-variant Logistic-CML is proposed to predict a segment of EEG data, the results are shown in Fig. 2. The mean correlation coefficient between the real data and the predictive data in each lattice is also estimated. The value is up to 0.96243.

The results indicated that the 2-D time-variant Logistic-CML model can effectively fit the real system in every lattice. Thus the new model is able to represent the physical system and effectively predict the EEG data.

4 ERP Data and Analysis

The dynamic nonlinearity of the ERP's under specified cognitive tasks based on the defined time-varying LEE spectrum is investigated in this section. Ten male students (age 22-25, dextral) participated in the ERP experiment. The CAD-Well 10-20 ERP System was used to collect the EEG at sampling rate of 1000Hz. Electrodes on earlobes were selected as the reference electrodes. Each EEG data collection lasted 1400ms in which the first 400ms data was recorded before the stimuli presented.

In order to study the nonlinear dynamics of the ERPs under different cognitive tasks, the 2-D time-variant Logistic-CML model was used to estimate the LLE parameters [9]. A grid with scale 5×5 adjacent leads was designed to calculate the LLE from 200ms before the stimulus to 800ms after the stimulus. The boundary condition was supposed periodic with (0, :)=(5, :) and (:,0)=(:,5). The initial condition was random and the first 2000 steps were regarded as transition. The time-variant LLE of the spontaneous EEG and the ERP signals were estimated in every step. The corresponding time-variant LLE spectra were shown in Fig. 3.

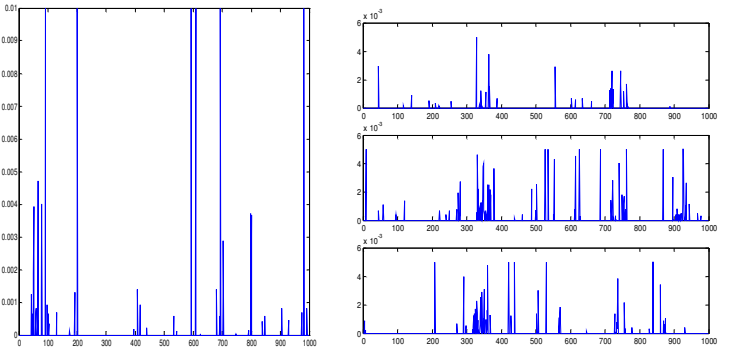


Fig. 3. The time-variant LLE of the signals from 1 to 1000 steps. (a) The time-variant LLE of the spontaneous EEG, (b) From top to bottom: the time-variant LLE of the ERP under three cues with small, middle and large scale of the target.

5 Results and Discussion

As a nonlinear dynamic parameter, LLE is one of important and direct parameter to test if a system is chaos. Based on the experimental result of the time-variant LLE spectra of the spontaneous EEG, the appearance of time steps with chaos character is irregular since the so-called spontaneous EEG is not really spontaneous. The mental activities during the information processing and some uncontroled external factors may influence the EEG. For the time-variant LLE spectra of the ERPs, it is found that there were more time steps showing the chaos of the system, behind 100ms after stimulus and especially between 100-200ms after stimulus under three kinds of cues. Moreover, the appearances of these time steps had their regular patterns. Obviously, the cognitive task significantly strengthened the chaos character of the system. The main influences of the chaos character happened behind 100ms after stimulus, especially in P2 period of the responses of the signals.

It is found that the middle scope cue most significantly influence the chaos degree of the system, which indicated that there was not simple linear relationship between the scopes size of the cues and the influence to system by comparing the three time-variant LLE spectra of the ERP under different scope cues. The result also indicates that there exists the optimal scope size for attention test. This optimal scope size became an important factor for studying the visual attention information process.

6 Conclusions

A novel method for investigating the complexity of the dynamic brain system was presented. The model based on the time-variant coupled map lattice was developed and the time-variant LLE spectrum was also defined as the quantitative parameters to reveal the time-varying characters of the system and discover new information involved in the EEG. Both real spontaneous EEG and ERP signals under different

fixed location cue were examined with the LLE spectra. The results indicated that the brain chaos patterns were time-varying under different cognitive tasks of attention. The main influence of the LLE spectrum corresponding to the scopes size of the cues mainly occurs in P2 period (120-260ms). No simple linearly relationship was found between the scopes size of the cues and the complexity of the signal. The chaos character of the brain enhanced and become regular during the period of processing information under different attention tasks.

Acknowledgements

The research was supported by the Natural Science Foundation of China (60271023 and 60571066) and the Natural Science Foundation of Guangdong Government (5008317).

References

1. Michel, O. and Flandrin, P: Application of Methods Based on High Order statistics for Chaotic Time Series Analysis. Signal Processing 53 (1996) 133-148
2. Rapp, P. E., *et al*.: Dynamics of Spontaneous Neural Activity in the Simian Motor Cortex: The Dimension of Chaotic Neurons. Phys. Letter A 110 (1985) 335–338
3. Dudkin, A. O. and Sbitnev, V. I.: Coupled Map Lattice Simulation of Epileptogenesis in Hippocampal Slice. Biol. Cybern. 78 (1998) 479-486
4. Leung, H., Hennessey, G. and Drosopoulos, A.: Signal Detection Using the Radial Basis Function Coupled Map Lattice. IEEE Transaction on Neur. Networks 11(2000)1133-1151
5. Oketani, N and Ushio, T.: Chaotic Synchronization of Globally Coupled Maps with an Application in Communication. Int. J. Bifurc. Chaos 6 (1996) 2145-2152
6. Chen, P. L. and Chung, W. P.: An Approach to Quantifying the Multi-Channel EEG Spatial-Temporal Feature. Biometrical Journal 42 (2000) 901-916
7. Lachaux, J. P.: Spatial Extension of Brain Activity Fools the Single-Channel Reconstruction of EEG Dynamics. Human Brain Mapping 5 (1997) 26-47
8. Lorenz, E. N.: Dimension of Weather and Climate Attractors. Nature 353 (1991) 241–244
9. Hiroshi, S.: Fluctuation of Mean Lyapunov Exponent for a Coupled Map Lattice Model. Physica. A 5 (2000) 124-130

ICA-Based EEG Spatio-temporal Dipole Source Localization: A Model Study

Ling Zou[1], Shan-An Zhu[2], and Bin He[3]

[1] Department of Computer Science and Technology,
Jiangsu Polytechnic University, 213016, China
zoulingme@yahoo.com.cn
[2] College of Electrical Engineering, Zhejiang University, Hangzhou, China
zsa@zju.edu.cn
[3] Dept. of Biomedical Engineering, University of Minnesota,
Minneapolis, MN 55455, USA

Abstract. In this paper, we examine the performance of an Independent Component Analysis (ICA) based dipole localization approach to localize multiple source dipoles under noisy environment. Uncorrelated noise of up to 40% was added to scalp EEG signals. The performance of the ICA-based algorithm is compared with the conventional localization procedure using Simplex method. The present simulation results indicate the robustness of the ICA-based approach in localizing multiple dipoles of independent sources.

1 Introduction

Solving EEG inverse problem is of significance and importance for both clinical and research applications. The equivalent current dipole model has been used to represent well localized neuronal sources [1]-[8]. The inverse problem is then defined as the estimation of the location and moment parameters of one or more dipoles whose modeled potentials best fit the actual measurements in a least-squares sense [2]. Mathematically, it is a nonlinear optimization problem and is not an easy task when the number of dipole sources is large and when measurement noise exists.

The objective of the present study was to examine the feasibility of improving multiple dipole localization when substantial measurement noise exists; by applying the Independence Component Analysis (ICA) approach [9]. Computer simulation studies were conducted to evaluate the ICA-based dipole localization approach as compared with conventional optimization methods.

2 Methods

In spatio-temporal source modeling (STSM), the spatio-temporal source of the signals is taken into account as a whole. Thus, instead of considering each time point separately, STSM combines the potential recordings at all time points to form a spatio-temporal data matrix V. This matrix has column entries that correspond to measurements at a single time point across the electrode array and row entries that correspond to the time series observed at a single electrode. Assuming simultaneous recordings at m sensors for k time instances from n current dipole sources, we have

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 566 – 572, 2006.

$$V = GQ + N . \quad (1)$$

where V is the measured scalp electric potential matrix, G the gain matrix, Q the moment matrix of the dipole sources and N the noise matrix.

ICA is an effective technique for finding statistically independent sources and independent basis vectors spanning the signal subspaces. It has been previously applied to EEG dipole localization [6]. In the present study, the infomax ICA algorithm was used [10]. After removing noise from the simulated EEG data, ICA was applied to separate the independent sources. The single-dipole-localization procedure was then used to localize each of the ICA-separated dipole sources.

An inhomogeneous concentric three-sphere volume conductor model with an outer radius of 10 cm was used to approximately represent the head volume conductor model [11]. 129 scalp electrodes, uniformly distributed over the upper hemisphere, were used in the present simulation.

Three cases of fixed-location and fixed-orientation current dipoles with time varying strengths were assumed to generate the spatio-temporal potential data matrix V for an interval of k=160, 160, 500 time points respectively. The sampling frequency was 200Hz for both case 1 and case 2 and was 1000Hz for case 3. Different noise levels were added in these cases to evaluate the performance of the algorithms.

There were two separated dipoles in Case 1, which had the eccentricity of 0.55, 0.75 respectively, and the distance between them was around 4cm. The orientations of the two dipoles were radial, which were fixed to be (0.17, 0, 0.216), (0.35, 0.025, 0.1325), and the dipole strength were set to be sinusoidal functions with dipole 1 being of 10Hz and dipole 2 of 40 Hz.

In Case 2, the dipoles were close to each other with the same eccentricity of 0.75, and the distance between them was about 1cm. The orientations of the two dipoles were both tangent and parallel, which were fixed to be (-0.1325, 0, 0.35), and the dipole strength were set to be sinusoidal functions with dipole 1 being of 40Hz and dipole 2 of 35 Hz.

There were three dipoles in Case 3, which had the eccentricity of 0.55, 0.75 and 0.75 respectively. The orientations of the three dipoles were fixed to be (-0.216, 0.3, 0.17), (0.35, 0.025, 0.1325), (0.35, -0.025, 0.1325); and the dipole strength were set to be sinusoidal functions with dipole 1 being of 10Hz, dipole 2 of 40 Hz and dipole 3 of 35Hz.

Dipole parameters are given as follows:

Case1: (0.34, 0, 0.432, 0.17, 0, 0.216); (0.7, 0.05, 0.265, 0.35, 0.025, 0.1325)
Case2: (0.7, 0.05, 0.265, -0.1325, 0, and 0.35); (0.7,-0.05, 0.265,-0.1325, 0, 0.35)
Case3: (0.34, 0, 0.432,-0.216, 0.3, and 0.17); (0.7, 0.05, 0.265, 0.35, 0.025, 0.1325); (0.7,-0.05, 0.265, 0.35,-0.025, 0.1325)

Given a suitable source and a head model, the inverse procedure for STSM requires minimizing the following least squares cost function

$$E_{STSM} = \|V - GQ\|_F^2 . \quad (2)$$

where $\|\bullet\|_F$ indicates the Frobenius norm (For matrix A, $\|A\|_F = \sqrt{\sum_{i=1}^{n} (A^T A)_{ii}}$). If we were to use this cost function directly in an iterative minimization algorithm, it would search for 3n location parameters and 3nk moment parameters, for an overall 3n (k+1) parameters. For realistic problems of practical size, the total number of parameters and nonlinear nature of the cost function makes this minimization difficult.

The computational complexity of the inverse procedure can be simplified by separating the linear and nonlinear dipole parameters because V is linear function of parameter Q. First, we assume that the locations are known, and then a solution for matrix Q that minimizes E_{STSM} is

$$Q = G^* V \quad . \tag{3}$$

where G^* is Moore-Penrose pseudo-inverse. The cost function for the spatio-temporal model then becomes

$$E_{STSM} = \|V - G(G^* V)\|_F^2 \cdot \tag{4}$$

Now the cost function depends only on the gain matrix G, which is a function of only 3n nonlinear location parameters. The number of parameters is reduced to 9 in the case of three dipoles. After finding the value of L using some nonlinear optimization technique, 3nk moment parameters in Q can be estimated using Equation (3).

For each electrode potential map V^i, a single dipole can be localized by using the Quasi-Newton method to minimize Equation (3). The dipole orientation is calculated as the linear least square solution. By considering the activation map of each dipole separately, we search for only one dipole and the search space is dramatically reduced. So we can search for the nonlinear least square solution without additional computationally complex nonlinear optimization techniques.

3 Results

Tables 1 and 2 showed the dipole localization errors obtained by ICA and Simplex methods, corresponding to dipole source configuration #1 (Case 1). The values shown in table 1 and 2 were the mean values averaged over 10 trials of Gaussian white noise. Table 1 and 2 clearly indicated that the ICA-based localization algorithm can substantially decrease the localization error as compared with the conventional Simplex dipole localization procedure. For example, with 10% noise level, the conventional Simplex dipole localization procedure resulted in about 2.3 cm localization errors for each of the 2 dipoles while the ICA-based localization algorithm resulted in about 0.5-2mm localization error. With 40% noise level, the Simplex localization procedure resulted in about 3-5.1cm for each of the 2 dipoles, while the ICA-based procedure resulted in about 3.5-13.5 mm for each of the 2 dipoles. Table 1 and 2 suggested that for 2 well separated dipole sources within the brain (4 cm

apart and at different eccentricity), the ICA-based approach can effectively reduce the localization error when the uncorrelated noise was up to 40%.

Table 1. Simulation results by ICA-based procedure in case 1

NL (%)		CC	LE (mean ± std[mm])	RE (mean)
0	Dipole1	0.9999	0.40 ± 0.29	0.0035
	Dipole2	0.9999	0.22 ± 0.10	0.0027
5	Dipole1	-0.9968	1.26 ± 0.47	0.0482
	Dipole2	-0.9986	0.32 ± 0.09	0.0291
10	Dipole1	0.9873	2.04 ± 0.42	0.0933
	Dipole2	0.9938	0.56 ± 0.06	0.0591
20	Dipole1	0.9411	5.35 ± 0.29	0.2158
	Dipole2	-0.9784	1.39 ± 0.08	0.1252
30	Dipole1	0.8629	5.30 ± 0.21	0.4132
	Dipole2	-0.9526	3.04 ± 0.14	0.1967
40	Dipole1	0.7847	13.20 ± 1.25	0.5150
	Dipole2	0.9214	3.74 ± 0.02	0.2714

Table 2. Simulation results by Simplex procedure in case 1

NL (%)		LE (mean ± std[cm])	RE (mean)
0	Dipole1	2.35 ± 0.65	0.0128
	Dipole2	0.95 ± 0.33	
5	Dipole1	2.50 ± 0.35	0.2115
	Dipole2	1.96 ± 0.86	
10	Dipole1	2.44 ± 0.67	0.4023
	Dipole2	2.18 ± 0.94	
20	Dipole1	3.64 ± 1.37	0.6591
	Dipole2	2.04 ± 0.54	
30	Dipole1	5.99 ± 1.14	0.7774
	Dipole2	3.17 ± 2.08	
40	Dipole1	5.03 ± 1.83	0.8703
	Dipole2	3.31 ± 1.37	

Table 3 and 4 showed the simulation results for Case 2 when 2 source dipoles were located at the same eccentricity and with only 1 cm distance apart. For such relatively closely spaced dipole sources, the ICA-based procedure again resulted in low localization error of less than 0.7-21 mm, while the conventional Simplex dipole localization procedure led to about 1.1-3.7 cm localization error for each of the 2 dipoles when noise level was within 15%. Compared with case 1, antinoise performance was worse in case 2.

Table 3. Simulation results by ICA-based procedure in case 2

NL (%)		CC	LE (mean ± std[mm])	RE (mean)
0	Dipole1	0.9960	0.71 ± 0.20	0.0031
	Dipole2	0.9977	0.74 ± 0.14	0.0024
5	Dipole1	0.9544	2.07 ± 0.16	0.0323
	Dipole2	-0.9586	0.80 ± 0.07	0.0213
10	Dipole1	0.7207	9.03 ± 0.48	0.1635
	Dipole2	0.8390	2.85 ± 0.04	0.0386
15	Dipole1	0.2764	20.84 ± 3.50	0.5912
	Dipole2	0.7230	5.28 ± 0.05	0.0587

Table 4. Simulation results by Simplex procedure in case 2

NL (%)		LE (mean ± std[cm])	RE (mean)
0	Dipole1	2.38 ± 0.77	0.0157
	Dipole2	1.11 ± 0.79	
5	Dipole1	3.29 ± 1.91	0.2664
	Dipole2	1.64 ± 0.87	
10	Dipole1	2.68 ± 1.55	0.4591
	Dipole2	2.09 ± 0.73	
15	Dipole1	2.93 ± 2.09	0.5791
	Dipole2	3.64 ± 3.04	

Table 5. Simulation results by ICA-based procedure in case 3

NL (%)		CC	LE (mean ± std[mm])	RE (mean)
0	Dipole1	0.9999	0.43 ± 0.43	0.0036
	Dipole2	0.9999	0.13 ± 0.12	0.0016
	Dipole3	0.9999	0.15 ± 0.12	0.0019
5	Dipole1	-0.9965	0.68 ± 0.20	0.0318
	Dipole2	0.9530	0.89 ± 0.09	0.0356
	Dipole3	-0.9489	0.49 ± 0.09	0.0347
10	Dipole1	-0.9864	1.03 ± 0.11	0.0683
	Dipole2	0.8608	2.63 ± 0.07	0.0934
	Dipoke3	-0.8489	2.15 ± 0.09	0.0899
15	Dipole1	0.9742	1.08 ± 0.40	0.1150
	Dipole2	-0.4003	18.82 ± 0.68	0.5705
	Dipole3	-0.7282	4.01 ± 0.08	0.0772

When the number of dipoles was larger than 2, it was hard to achieve reasonable localization by Simplex method. On the other hand, the ICA-based approach was still effective. Table 5 and 6 showed the mean dipole localization error by using ICA and

Table 6. Simulation results by Simplex procedure in case 3

NL (%)		LE (mean ± std[cm])	RE (mean)
0	Dipole1	3.80 ± 2.81	0.0266
	Dipole2	3.21 ± 2.34	
	Dipole3	3.65 ± 2.90	
5	Dipole1	3.95 ± 1.98	0.2893
	Dipole2	3.28 ± 2.00	
	Dipole3	3.58 ± 2.60	
10	Dipole1	3.68 ± 0.76	0.5467
	Dipole2	4.51 ± 2.70	
	Dipole3	4.67 ± 2.29	
15	Dipole1	4.49 ± 2.31	0.7096
	Dipole2	3.94 ± 2.97	
	Dipole3	6.19 ± 2.78	

Simplex. Note that the estimation errors were less than 2cm when noise level was within 15% by using ICA-based procedure, while the localization errors had reached above 6cm for Simplex method.

From above tables, we can see that the indexes, i.e., correlative coefficient (CC) decreased while localization errors (LE) and relative errors (RE) increased with the noise level (NL) increasing. The low relative errors showed that the difference between the measured and dipole-generated potentials was little, which showed an inversely-proportional relationship with the noise level.

4 Discussions

In the present study, we have conducted computer simulations to examine the feasibility of localizing multiple dipole sources from the noise-contaminated scalp EEG by means of ICA approach. Such ICA-based approach has been previously reported in a computer simulation study for 3 spatially well separated dipoles with 2% additive noise added [6]. However, no data were available on the performance of such ICA-based dipole localization approach when there exist high noise levels as in clinical and experimental settings. The present computer simulation study suggests that the ICA-based approach is effective and efficient for multiple dipole localization with substantial uncorrelated additive noise of up to 40%. Also note that the ICA-based dipole localization approach is similarly effective for source configuration of more than 2 dipoles (shown here at 3 dipoles), while the conventional localization procedure using Simplex method would experience substantial difficulty in localizing the multiple dipoles.

Acknowledgments

This work was supported in part by NSF BES-0218736 and NIH R01 EB00178.

References

1. Scherg, M., von Cramon, D.: Two Bilateral Sources of the AEP as Identified by a Spatio-temporal Dipole Model. Electroenceph. Clin. Neurophysiol. 62 (1985) 32–44
2. He, B., Musha, T., Okamoto, Y., Homma, S., Nakajima, Y., Sato, T.: Electric Dipole Tracing in the Brain by Means of the Boundary Element Method and Its Accuracy. IEEE Trans. Biomed. Eng. 34 (1987) 406–414
3. Salu, Y., Cohen, L.G., Rose, D., et al.: An Improved Method for Localizing Electric Brain Dipoles. IEEE Trans. Biomed. Eng. 37 (7) (1990) 699–705
4. Mosher, J.C., Lewis, P.S., Leahy, R.M.: Multiple Dipole Modeling and Localization from Spatio-temporal MEG Data. IEEE Trans. Biomed. Eng. 39 (6) (1992) 541–557
5. Cuffin, B.N.: A Method for Localizing EEG Sources in Realistic Head Models. IEEE Trans. Biomed. Eng. 42 (1995) 68–71
6. Zhukov, L., Weinstein, D., Johnson, C.: Independent Component Analysis for EEG Source Localization. IEEE Eng. Med. Biol. Mag. 19 (3) (2000) 87–96
7. Kosugi, Y., Uemoto, N., Hayashi, Y., He, B.: Estimation of Intra-cranial Neural Activities by Means of Regularized Neural-network-based Inversion Techniques. Neurological Research 23 (2001) 435–446
8. He, B., Lian, J.: Spatio-temporal Functional Neuroimaging of Brain Electric Activity. Critical Review of Biomedical Engineering 30 (2002) 283–306
9. Comon, P.: Independent Component Analysis, A New Concept? Signal Processing 36 (1994) 287–314
10. 10.Lee, T.-W., Girolami, M., Sejnowski, T.J.: Independent Component Analysis Using an Extended Infomax Algorithm for Mixed Sub-Gaussian and Super-Gaussian Sources. Neural Computation 11 (2) (1999) 409–433
11. Wang, Y., He, B.: A Computer Simulation Study of Cortical Imaging from Scalp Potentials. IEEE Trans. Biomed. Eng. 45 (6) (1998) 724–735

Networking Property During Epileptic Seizure with Multi-channel EEG Recordings

Huihua Wu, Xiaoli Li, and Xinping Guan

Institute of Electrical Engineering, Yanshan University, Qinhuangdao 066004, China
xlli@ysu.edu.cn

Abstract. EEG recordings are widely used in epilepsy research. We intend to address a question whether small world network property exists in neural networks when epileptic seizures occur. In this paper, we introduce a bispectrum analysis to calculate the interaction between two EEG recordings; then, a suitable threshold is chosen to convert the interaction of the six channels at five frequency bands to a sparse graph (node: n=30, edge: k=4-7). Through analyzing a real EEG recording, it is found the clustering coefficient is similar to that of regular graph; however the path length is less than that of regular graph. Thus a primary suggestion can be made that neural networks possess small world network characteristic. During epileptic seizures, the small world property of neural network is more significant.

1 Introduction

An epileptic seizure is a successive sudden burst of neural activity in the brain. Epileptic seizures detection or mechanism analysis is performed by EEG recordings [1], [2]. Since the small world network concept has been proposed by Stanley Milgram (1967), many studies have shown that small world network property exists in many fields such as, movie actor collaboration network, the World Wide Web network, power system and neural networks [3], [4].

In this paper, we intend to address a question whether small world network property exists in neural network when epileptic seizures occur. First of all, we convert it to a sparse network based on bispectrum analysis of multi-channel EEG data. The structural properties of the network are quantified by the path length and clustering coefficient by graph theory. A real case of epileptic seizure is analyzed, and the result shows that small world network property exists in neural network, and it is more significant during the epileptic seizure. The clustering coefficient of the neural network is similar to that of regular graph, however the path length is less than that of regular graph. It is a possible mechanism of epileptic seizure that small world property of functional connectivity support synchronization of local neural network and information transmission of distant neural network.

2 Method

2.1 Phase Coupling

Phase coupling is prevalent in neural signal processing [5], [6]. It depicts the interaction of two independent signals. In general, higher order spectrum (HOS) is

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 573 – 578, 2006.

used to get phase coupling information in different fields, communication, radar, sonar, speech, biomedical, geophysical and imaging systems [7]-[9]. In this paper, we use a bispectrum analysis to extract the phase coupling information of two EEG signals [10], [11]. The natural estimate of the bispectrum is the FFT of the third-order cumulate sequence. Bispectrum, in discrete form, transforms one-dimensional time domain data into a two-dimensional frequency domain cross-come matrix. It is defined as:

$$\hat{S}_{xyz}(f_1,f_2)=\sum_{k=-N-1}^{N-1}\sum_{I=-N-1}^{N-1}\hat{C}_{xyz}(k,I)w_M(k,I)\exp[-j(2\pi f_1 k+2\pi f_2 I)] \quad (1)$$

Where the scaled-parameter window, $w_M(t,s)=w(t/M,s/M)$. In order to get a consistent estimation, we segment the data into K epochs of length $L = N/K$, compute the bispectrum of each epoch and average them, then perform the frequency smoothing by using the frequency domain filter, $W_M(f_1, f_2)$, the FFT of $w_M(t,s)$. More details can be found in [12].

2.2 Small World Network

Small world network is between the regular and the random network, in which most of the nodes are connected to their nearest neighbors, but a few of them can spread over a long range. The most popular manifestation is "six degrees of separation" concept, proposed by the social psychologist Stanley Milgram (1967), which means there is a path of acquaintance with a typical of about six between most pairs of people in the United State [13].

The pioneering article of Watts and Strogatz started an avalanche research on the property of small world network [14]. To interpolate between regular and random networks, we have to consider the following random rewiring procedure. First, start with a ring lattice with N nodes in which every node is connected to its first k neighbors (considering $N>>k>>Ln(N)>>1$, where $k>>Ln(N)$ guarantees that a random graph could be connected). Second, rewire each edge of the lattice with probability P such that self –connections and duplicating edges are excluded. By varying P one can closely monitor the transition between order P=0 and randomness P=1 [3]. There are two characteristic parameters to describe the small world network: clustering coefficient (CC) and path length (PL). It is known that small world network is highly clustered, like regular lattices, yet has low characteristic path lengths, like random graphs [15].

3 Result

3.1 EEG Recordings

The EEG data is recorded during an invasive pre-surgical epilepsy monitoring. The epileptic focus is located in neocortical brain structures. The patient is female, 32 years old. In order to obtain a high signal-to-noise ratio, fewer artifacts, and to record directly from focal areas, intracranial grid-, strip-, and depth-electrodes were utilized. The EEG data were acquired using a Neurofile NT digital video EEG system with 128 channels, 256 Hz sampling rate, a 16 bit analogue-to-digital converter, the details

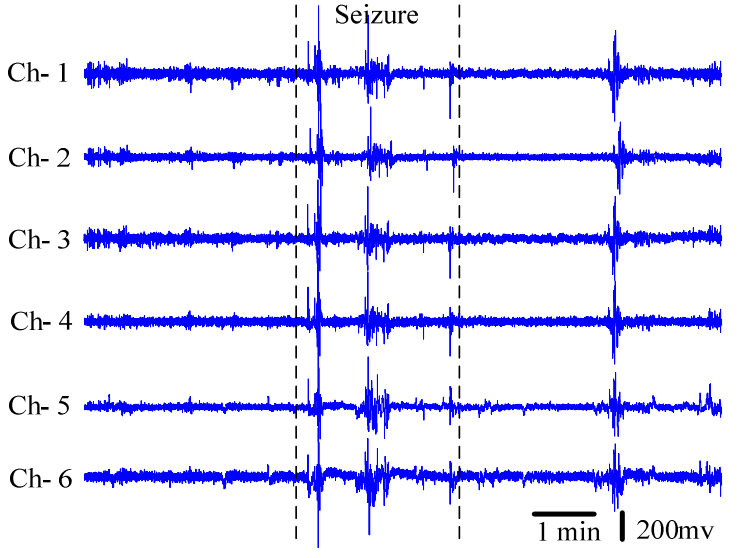


Fig. 1. Multi-channel EEG recordings, the length is about 10 minutes from 3.5 minutes before epileptic seizure, sample rate=256. The patient is a female, 32 years old.

can be found in [16], [17]. In this paper, the six-channels EEG recording is about 10 minutes, the epileptic seizure occurs at 3.5 minutes, as shown in Fig. 1.

3.2 Result

The bispectrum is computed, using a Hanning window of 4 seconds with a 50% overlapping, to describe the relationship between EEG recordings. Then, the average of bispectrum at the five frequency bands (1-4 Hz, 4-8 Hz, 8-16 Hz, 16-32 Hz and

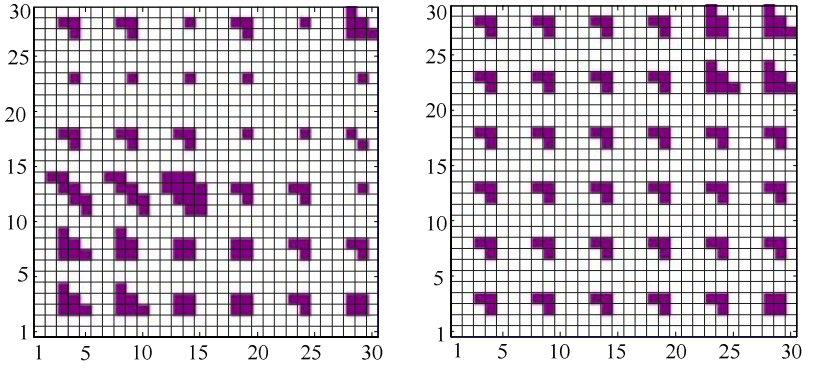


Fig. 2. The 30*30 matrix connect graph of six EEG recording at the five frequency bands. The *left* graph's CC is 0.31577, PL is 4.5978 (preseizure) and the *right* graph's CC is 0.55126, PL is 3.4476 (seizure). The red lattice represents the node's value is 1 and the white one is 0 (n=30, k=4).

32-64 Hz) is computed. Setting an edge k, a spare network can be obtained. Fig. 2 shows two typical graphs at the pre-seizure and seizure.

Above method is applied to treat with the EEG recordings in Fig. 1. Fig. 3 shows the clustering coefficient and path length over time at the edge of 4. At the same time,

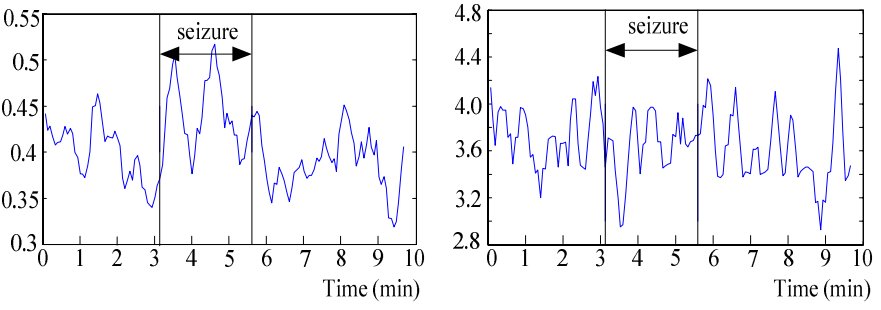


Fig. 3. The *left* graph is the clustering coefficient (CC) of each segment over time (n=30, k=4). The CC is much higher than that of normal state. The *right* is the average path length (PL) of each segment.

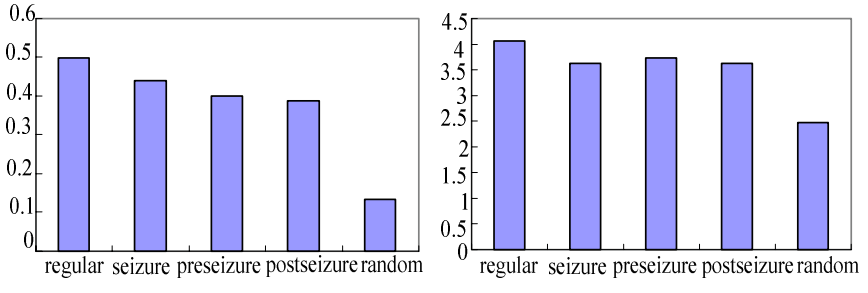


Fig. 4. The *left* graph shows the comparison of CC of the regular and random graph (n=30, k=4). The *right* one shows the comparison of PL of the regular and random graph.

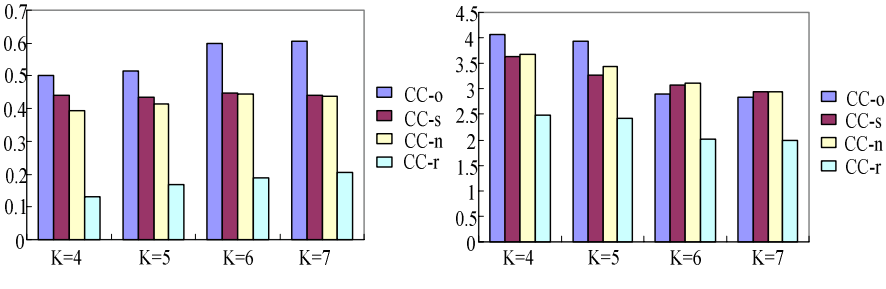


Fig. 5. *Left,* the clustering coefficient of epileptic seizure, normal state, regular and random network (n=30, k=4-7). *Right,* the path length of epileptic seizure, normal state, regular and random network (n=30, k=4-7).

the regular and random graph with the same nodes and edges are analyzed. It is found that that the average path length is low and similar to that of random graph, however, the clustering coefficient is high and during epileptic seizure it increases, as shown in Fig. 3 and Fig. 4. Also, we found that the clustering coefficient and path length of graph change as changing the edges. At k=4 and 5, the clustering coefficient of the graph is high and the path length of the neural network is low, it is similar to small world network property. However, k=6 and 7, the path length of the graph of neural network is higher than one of the regular graph, the graph is different from the small world network, as shown in Fig. 5.

4 Conclusion

In this paper, we introduce a bispectrum analysis to extract phase coupling information among EEG recording, and then construct a sparse but connected graph. By computing the graph's clustering and path length, it is found that small world network property exists in neural networks, when the epileptic seizures occur; small world network property of neural network is more significant. However, small world network property of the neural network only appears just at the edge of k=4 and 5, but k=6, 7.

Acknowledgments

The support of National Science Foundation of China (60575012) is gratefully acknowledged. We would like to thank the Epilepsy Center of the University Hospital of Freiburg, Germany to provide EEG data for analysis.

References

1. Iasemidis, L.D., Pardalos, P.M., Principe, J.C.: Adaptive Epileptic Seizure Prediction System. IEEE Transactions on Biomedical Engineering 50 (2003) 616-627
2. Hoeve, M.J., Jones, R.D., Carroll, G.J., Goelz, H.: Automated Detection of Epileptic Seizures in the EEG. Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Istanbul, Turkey 3 (2001) 943-946
3. Albert, R., Barabasi, A.L.: Statistical Mechanics of Complex Networks. Review of Modern Physics 74 (2002) 47-97
4. Albert, R., Barabasi, A.L.: Emergence of Scaling in Random Networks. Science 286 (1999) 509-512
5. Pikovsky, A.S., Rosenblum, M., Kurths, J.: Synchronization, a Universal Concept in Nonlinear Sciences. Cambridge University Press, Cambridge (2001)
6. Pikovsky, A.S., Rosenblum, M.G., Osipov, G.V., Kurths, J.: Phase Synchronization of Chaotic Oscillators by External Driving. Physica D 104 (1997) 219-238;
7. Helme, B.I., Nikias, C.L.: Improved Spectrum Performance via a Data Adaptive Weighted Burg Technique. IEEE Transactions on Signal Processing 33 (1985) 903–910
8. Nikias, C.L., Raghuveer, M.R.: Multidimensional Parametric Spectral Estimation. Signal Processing 9 (1985) 280–290

9. Shen, C., Lemmin, U.: Ultasonic Intensity Power Spectrum Estimation by Using the Bispectral Reconstrucion Technique. Signal Processing 61 (1997) 39–44
10. Sigl, J.C., Chamoun, N.G.: An Introduction of Bispectral Analysis for the Electroencephalogram. Journal of Clinical Monitoring 10 (1994) 392-404
11. Hinich, M.J., Clay, C.S.: The Application of the Discrete Fourier Transform in the Estimation of Power Spectra, Coherence and Bispectra of Geophysical Data. Review of Geophysics 6 (1968) 347-363
12. Swami, A., Mendel, J.M., Nikias, C.L. (Max): Higher-Order Spectral Analysis Toolbox. The MathWorks, Inc (1998)
13. Kochen, M. (ed.): The Small World. Ablex, Norwood, New Jersey (1989)
14. Watts, D.J., Strogatz, S.H.: Collective Dynamics of 'Small-World' Networks. Nature 393(1998) 440–442
15. Olaf, S., Jonathan, D.Z.: The Small World of the Cerebral Cortex. Neuroinformatics, Vol.2 (2004) 145-162
16. Maiwald, T., Winterhalder, M., Aschenbrenner-Scheibe, R., Voss, H.U. Schulze-Bonhage, A., Timmera, J.: Comparison of Three Nonlinear Seizure Prediction Methods by Means of the Seizure Prediction Characteristic. Physica D 194 (2004) 357–368
17. Aschenbrenner-Scheibe, R., Maiwald, T., Winterhalder, M., Voss, H.U., Timmer, J., Schulze-Bonhage, A.: How Well Can Epileptic Seizures Be Predicted? An Evaluation of a Nonlinear Method. Brain 126 (2003) 2616–2626

Time-Frequency Analysis of EEG Based on Event Related Cognitive Task

Xiao-Tong Wen[2], Xiao-Jie Zhao[1,2], and Li Yao[1,2]

[1] Department of Electronics, Beijing Normal University, Beijing, China, 100088
[2] Institute of Cognitive Neuroscience and Learning, Beijing Normal University, Beijing, China, 100088
zhaoxj86@hotmail.com

Abstract. Compared with traditional time analysis and frequency analysis, time-frequency analysis of EEG not only reveals more abundant information but also reflects their dynamic temporal oscillatory activities. Event related potential (ERP) is a kind of cognitive electroencephalogram (EEG) signal, which are directly related with task and average across spontaneous EEG. Effective time-frequency representation method and how to analyze EEG based on the event related cognitive task in time-frequency domain according to their experiment speciality are the hot study at present. In current paper, a new time-frequency method about filter bank and Hilbert transform (FBHT) was introduced, and the application for ERP and ER-EEG signal were explored in order to mine new information effectively during the classic Stroop cognitive task.

1 Introduction

Because of the non-stability of biological signals, it is hard to describe the signal's local dynamic feature in time-frequency domain only by time analysis or frequency analysis. The time-frequency analysis method is applied widely to biological signals processing at present [1]. Electroencephalogram (EEG) is a kind of biological spontaneous potential that records electric temporal signal from the scalp position. Event related potential (ERP) also called evoked potential is a kind of cognitive EEG. The potential evoked by a single stimulus is often so weak that it has to be enhanced by repeating the stimulus many times and averaging across every EEG trials caused by the same stimulus. In other words, the event related EEG (ER-EEG) contains a series of trials each of which related to one repeated stimulus, while the ERP signal is the average result across these trials. ER-EEG directly reflects the electric activities of neuron assembly at different scalp electrode during cognitive task. With the development of modern signal analysis methods, people applied some new methods to the data with some conclusions already obtained by traditional methods and tried to mine new information that couldn't get by traditional methods. Therefore, the time-frequency analysis method was put forward and applied to physiological, pathological and cognitive EEG signals [2][3]. Durka et al analyzed the time-frequency feature of sleeping EEG by matching pursuit algorithm [4]. Hagoort et al studied N400 time-frequency variability of ERP signals by wavelet transform [5]. These studies obtained

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 579 – 585, 2006.

considerable new information that was not available by traditional methods. But the study also affected by the limitation of time-frequency method and the particular characteristics of cognitive task. Therefore, the research on the effective analysis of EEG is not only important but also helpful to reveal the brain neural mechanism.

This paper firstly introduced a filter bank and Hilbert transform (FBHT) time-frequency representation method. The result showed FBHT could reserve time-frequency details more perfectly. Then we analyzed ERP and ER-EEG signals of Stroop task respectively, and discussed the two analysis results. The study results were helpful to mine brain dynamic cognition activities in depth.

2 Time Frequency Representation of ERP

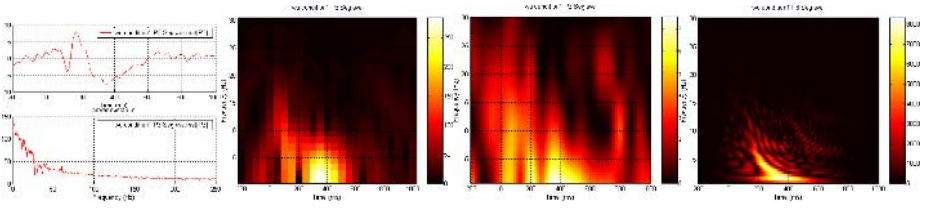
2.1 Data Recording

Classic psychological Stroop task of color words was designed. Stimuli were 4 colorful Chinese characters (red, yellow, blue and green) in different display colors. Subjects were asked to identify the display color of the Chinese character using keystroke. The stimuli with color-meaning congruent word and color-meaning incongruent word were all repeated for 96 times. EEG was recorded from 32 electrodes using ESI 128 channel workshop (NeuroScan, USA) with two referenced electrodes to two mastoids (band-pass 0.05~30 Hz, sampling rate 500 Hz). Trials contaminated by excessive eye-blink or movement artifacts, peak-to-peak deflections over 200μV, were rejected by an automatic procedure. The analysis time course was about 1000 ms post stimulus onset with baseline at 200 ms pre-stimulus [6].

2.2 The Application of Traditional Time-Frequency Tepresentation Methods in ERP

Time frequency analysis refers to representing, analyzing and processing the signal in the time-frequency domain. To perform time frequency analysis, the first thing to do is to represent the signal in the time-frequency domain. There are mainly two kinds of time frequency representations, one is linear such as short term Fourier transform (STFT) and wavelet transform, and the other is nonlinear. The widely used nonlinear representations are bio-linear representations such as Wigner distribution, Choi-Williams distribution etc. In STFT method, the signal is separated into many segments in the time domain and Fourier transform is applied to each of these segments to study its local frequency feature. Time frequency resolution of the result of STFT depends on the segment number, the more the segments are the higher the time resolution is. The frequency resolution is related to the segment length, long segment leads to high frequency resolution. But in the case of a fitted length signal, the segment length and the segment number are incompatible. Therefore, the time-frequency resolution of STFT is often poor. In wavelet packet method, the signal is decomposed according to standard wavelet packet decomposition tree, then, the results are rearranged according to frequency order from low to high to obtain the time frequency representation matrix, which is $n \times m$. The time resolution and frequency resolution are also restricted because $n \times m$ equals the length of the signal. Therefore, wavelet packet method couldn't obtain very high time frequency resolution either. And in the case of

Wigner distribution, the cross-term often generate none zero values where there is actually nothing in the time-frequency domain. We applied these three representative methods to ERP signals in Stroop task and calculated their time-frequency distributions (Fig. 1). The electrode P3 was chosen for the advantage that it could reflect Stroop effect well.



(a) (b) (c) (d)

Fig. 1. ERP signal and its spectrum of P3 (a), its time-frequency energy distribution of STFT (b), wavelet packet (c) and Wigner distribution (d). Time course was about 1000 ms post stimulus onset with baseline at 200 ms pre-stimulus. Frequency range was from 1 to 30 Hz. The result of wavelet packet decomposition was interpolated to facilitate the display.

Because the stimulus only occurred in a comparatively short time range and the ERP signal relied on the task has limited number of data points. The results demanded for high time-frequency resolution. In Fig. 1, although all of the three methods described the variation of the frequency components according to time, the time-frequency resolution of the results of STFT and wavelet packet method remained poor. The Wigner distribution appeared a lot of cross-term interference, which led to the lost of many details. Furthermore, the time-frequency representation by means of Wigner distribution could not reserve time-frequency phase information of the original signal. Accordingly, the several classic time-frequency representation methods mentioned above could not fit ERP signal. To overcome these, we introduced a method based on FBHT to obtain the signal's time-frequency representation, which had high time-frequency resolution and could offer both time-frequency amplitude information and time-frequency phase information.

2.3 Calculating Time-Frequency Representation of ERP by Using FBHT Method

In this method, a filter bank $H_{fc,w}(t)$ contains series of FIR filters with the same bandwidth w was designed. Their corresponding center frequency $fc(i)$ s covered the interesting frequency range with equal frequency intervals. Its frequency transfer function can be expressed as:

$$H_d(\omega)=\begin{cases}1 & \omega_l \le |\omega| \le \omega_h \\ 0 & \text{ot her}\end{cases}, H(\omega)=\frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(\omega)W_R(\omega-\theta)d\theta \qquad (1)$$

Here, $W_R(\omega)$ was the window function's spectrum.

FIR filters with 256 points and a sampling frequency of 500Hz was generated. In order to eliminate the negative effect of side lobe, each FIR filter was windowed by a Blackman-Harries window. The signal *s(t)* was filtered by each filter of the filter bank $H_{fc,w}(t)$ in turn to produce a series of narrow band signals $s_{fc,w}(t)$ centered by frequency *fc(i)*, which covered the same frequency range from *fc(i)-w/2* to *fc(i)+w/2*. $s_{fc(i),w}(t)$ could be regarded as the component of the original signal in the *i*th narrow frequency band. Then an analysis signal sequence $S_a(t)$ was obtained by following this:

$$s_a(t) = s_{fc,w}(t) + i(\tilde{s}_{fc,w}(t)) = A(t)e^{j\phi(t)} . \tag{2}$$

$$\tilde{s}_{fc,w}(t) = \frac{1}{\pi}\int_{-\infty}^{\infty}\frac{s_{fc,w}(\tau)}{t-\tau}d\tau . \tag{3}$$

This analysis signal sequence can be regarded as a function of *fc* and *t*, namely a kind of time-frequency representation of *s(t)*.

$$S(f,t) = A(f,t)e^{j\phi(f,t)} . \tag{4}$$

Here, *A(f,t)* is the amplitude time-frequency distribution and $\phi(f,t)$ is the phase time-frequency distribution. We define the energy distribution as:

$$E(t,f) = \log[A(f,t)^2 + 1] . \tag{5}$$

We applied this method to the ERP signal of Stroop task to calculate its time frequency representation *S(f,t)* and followed equation (5) to obtain its time-frequency energy distribution *E(f,t)* (Fig. 2). Comparing Fig. 2 with Fig. 1, there is a maximal positive wave at about 300 ms in Fig. 1 (a) which should belong to P300 according to its appearance time. The occurrence of P300 is related to selective attention [7]. From Fig. 1 (b), (c), (d) and Fig. 2, it was obviously that there was strong energy distribution at about 300 ms after stimulus onset, which ranged over a certain frequency band. In Fig. 1 (c), Wigner distribution offered less information and a poor time-frequency resolution. Wavelet packet and STFT showed distribution trends similar to FBHT, but a poorer ability to describe the details. Furthermore, the result of wavelet packet decomposition was affected directly by the choice of wavelet base. Consequently, we analyzed ER-EEG in time-frequency domain by FBHT.

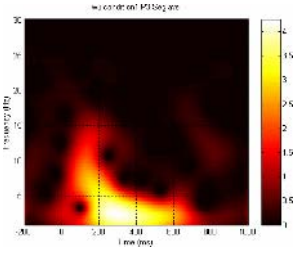


Fig. 2. Time-frequency energy distribution of ERP signal at P3 electrode using FBHT

3 Time Frequency Analysis of ER-EEG

3.1 Traditional Time Domain Analysis of ER-EEG

In traditional time domain analysis of ER-EEG, researchers mainly followed two approaches. One approach was to average ER-EEG signals across trails first, then analyzed the averaged signal namely ERP, in which, the evoked potential after averaging would emerge from the spontaneous potential due to the comparatively fixed amplitude and time intervals from stimuli onset [8]. The other approach was to process each trial first, and then the statistics method was used to analyze the results of all trials [9]. Although the method of averaging across trials could raise the SNR and give prominence to the characters of evoked potentials that were often visualized, some useful information hided under the varying disorder waveform, which often helped to explore the significant characters of the signal, were lost because of averaging. That was why the second kind of approach was widely applied in the analysis of ER-EEG signals.

3.2 Time-Frequency Statistic Analysis Based on ER-EEG

FBHT was used to calculate each trial's time-frequency energy distribution of P3 electrode (Fig. 3). It was seen that most of the trial has strong energy distribution at about 300 ms, moreover, the time-frequency component contained in each trial are much richer than that of ERP. In Fig. 2, most of the energy distributions were below 8 Hz (δband andθband), and over 8 Hz (α band) only between 100 ms and 200 ms after stimulus there existed evident energy distributions. While, in other time-frequency areas, the energy levels remained close to zero. Over 14 Hz (βband), few evident energy distribution appeared. But from the EEG time-frequency energy distribution result of each trial, there were not only strong energy distributions existed below 8 Hz, but also abundant time-frequency components varying in a relatively higher frequency band of 8 Hz to 25 Hz. The results above indicated that the phase randomicity in the comparatively higher frequency band was stronger than that in the comparatively lower frequency band. Consequently, the comparatively higher frequency components trended to be canceled out each other when directly averaging signal across trials, which presented in time-frequency energy distribution of ERP was the missing of the frequency component actually existed.

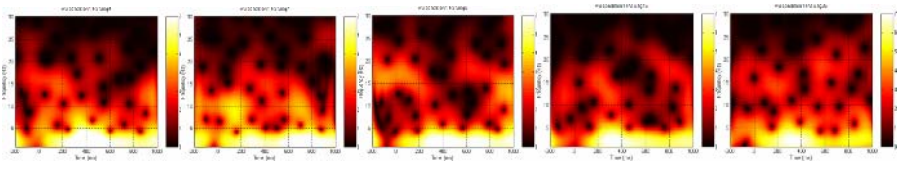


Fig. 3. Time-frequency energy distributions of 8 EEG trials chosen randomly at P3 electrode

Furthermore, we averaged the time-frequency energy distribution of each trial (Fig. 4) and compared the result with Fig. 2. Although the higher frequency components seemed more evident than that in Fig. 2, the energy focused did not locate around 300 ms but extended from 200 ms to 1000 ms. The main cognitive component

P300 was not evident anymore. Therefore, averaging time-frequency energy did not fit for studying ER-EEG. But we could use statistic analysis of all EEG trials to find the useful information that hided by averaged ERP. Considering that FBHT can also extract time-frequency phase information, this approach was used to analyze phase synchrony by performing statistical analysis to each corresponding trials of two electrodes so as to represent the phase synchronization in the time-frequency domain [10] (Fig. 5). It was thought that during the task-dependent period, synchronized oscillatory activities bound multiple neuronal assemblies together to form large-scale networks that supported cognitive functions. From Fig. 5, it also showed strong synchrony at about 300 ms and below 10 Hz between electrode P3 and P4 that indicated there existed coordination between the left and the right at the parietal region related to P300 component.

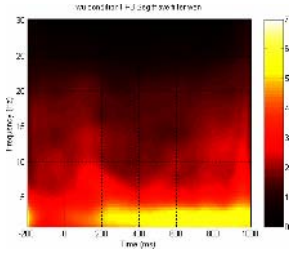


Fig. 4. Averaging the time-frequency energy distribution across all trails at P3 electrode

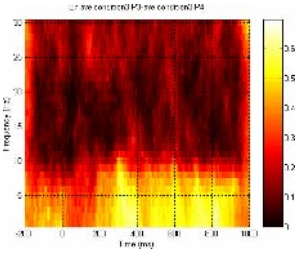


Fig. 5. Time-frequency phase synchrony between electrodes P3 and P4 based on statistic analysis of ER-EEG

4 Discussion

The study on application of signal's time-frequency representation mainly focused on the data self characteristic. An effective time-frequency representation method should meet two requirements: the advantage of method and its adaptation for data studied. FBHT time-frequency representation with comparative higher time frequency resolution and perfect local details was suitable for EEG signals. The unique feature of ER-EEG in experiment design and data acquisition that the stimulus was repeated many times, not only made the EEG caused by each stimulus has the similar component, but also made the components homologous. It benefited task-dependent ERP component extraction as well as the application of different statistics analysis methods. However, the time-frequency method broke the limitation of information separation in traditional methods, it's convenient for studying dynamic feature of the signal. For both ERP and ER-EEG signals analysis during the same cognitive task, we thought the time-frequency representation were useful to mine new information. To study what cognitive component like P300 etc activated and what oscillation activated with the component is mainly based on ERP analysis, while ER-EEG analysis could find more characteristic by statistic method and study the complex changeful EEG signals in depth.

The application of time-frequency analysis to Stroop task has obtained more abundant information such as the oscillatory activation and coordination of ER-EEG in different conditions. This information needs new reasonable cognition meaning studied by the cognitive neuroscience researchers. Meanwhile, we also could study the spatial time-frequency distribution among global brain by applying the time-frequency method to EEG signals at different electrodes. The study on spatial-temporal of ER-EEG could help to explain how the oscillation act and interact, and to reveal the neural coordinative mechanism of cognition processing more deeply.

Acknowledgment. This work is supported by National Natural Science Foundation of China. (60271014, 60472016).

References

1. Bianchi, A.M., Mainardi, L.T., Cerutti, S.: Time-Frequency Analysis of Biomedical Signals. Trans. Inst. Meas. & Control 22(3) (2000) 215-230
2. Blanco, S., Figliola, A., Quian Quiroga, R.: Time-Frequency Analysis of Electroencephalogram Series III: Wavelet Packets and Information Cost Function. Phys. Rev. E 57 (1998) 932-940
3. Williams, W.J.: Recent Advances in Time-Frequency Representations: Some Theoretical Foundation. Time-Frequency and Wavelets in Biomedical Signal. IEEE Press Series in Biomedical Engineering (1997)
4. Durka, P.J., Blinowska, K.J.: A Unified Time Frequency Parameterization of EEGs. IEEE Eng. Med. Biol. Mag. 20(5) (2001) 47-53
5. Hagoort, P., Hald, L., Bastiaansen, M., Petersson, K.M.: Integration of Word Meaning and World Knowledge in Language Comprehension. Science 304(5669) (2004) 438-441
6. Peng, D.L., Guo, T.M., Wei, J.H., Xiao, L.H.: An ERP Study on Processing Stages of Children's Stroop Effect. Sci. Tech. Engng. 4(2) (2004) 84-88
7. Polich, J., Kok, A.: Cognitive and Biological Determinants of P300: An Integrative Review. Biol. Psychol. 41(2) (1995) 103-146
8. Squires, N.K., Ollo, C.: Comparison of Endogenous Event-Related Potentials in Attend and Nonattend Conditions Latency Changes with Normal Aging. Clin. Neurophysiol. 110(3) (1999) 564-574
9. Lachaux, J.P., Rodriguez, E., Martinerie, J., Adam, C., Hasboun, D., Varela, F.J.: A Quantitative Study of Gamma-Band Activity in Human Intracranial Recordings Triggered by Visual Stimuli. European Journal of Neuroscience 12(7) (2000) 2608-2622
10. Wen, X.T., Zhao, X.J., Yao, L.: Synchrony of Basic Neuronal Network Based on Event Related EEG. In: Wang J., Liao X., Yi Z. (eds.): International Symposium on Neural Network. Lecture Notes in Computer Science, Vol. 3498. Springer-Verlag, Berlin Heidelberg New York (2005) 725-730

Adaptable Noise Reduction of ECG Signals for Feature Extraction

Hyun Dong Kim, Chul Hong Min, and Tae Seon Kim*

School of Information, Communications and Electronics Engineering
Catholic University of Korea, Bucheon, Korea
{hdkim81, ventura, tkim}@catholic.ac.kr
http://isl.catholic.ac.kr

Abstract. Conventionally, ECG signal is measured on static condition since various types of noise including muscle artifact noise and electrode moving artifact noise are in coupled in dynamic environment. To solve this problem, various noised signals are grouped into six categories by context estimation, and effectively reconfigured noise reduction filter by neural network and genetic algorithm (GA). Neural network based control module effectively select optimal filter block by noise context based clustering at running mode and filtering performance was improved by GA at evolution mode. Experimental results showed that proposed algorithm effectively remove baseline wander noise and muscle noise and feature extraction results showed significant improvement of T duration extraction values.

1 Introduction

The electrocardiogram (ECG) signal is a tracing of electrical activity signal generated by rhythmic contractions of the heart and it can be measured by electrodes placed on the body surface. The clinical experience data showed that ECG signal is very effective to detect heart disease and there are steady efforts of researchers to develop automated ECG signal classification algorithms. Conventionally, ECG signal is measured on static condition since the appearance of heartbeats varies considerably, not only among patients, but also movement, respiration and modifications in the electrical characteristics of the body [1]. Even, several electrical and mechanical noise components are also added to signal and it made difficult to extract key features. In general, measured ECG signal data contains muscle artifact noise, baseline wander noise, electrode moving artifact noise, and 60Hz power line noise. To overcome these difficulties, various noise reduction algorithms have been developed. The power line noise can be easily removed by notch filter since its frequency centered on 60Hz and typical ECG signal has lower frequency than 60Hz. Recent research results showed that wavelet interpolation filer (WIF) is effective to remove muscle noise [2]. For baseline wander noise removal, finite impulse response (FIR) filter and infinite impulse response (IIR) filter, and adaptable filter are widely used. FIR filter or IIR filter showed good characteristic of noise removal within cutoff frequency band but they require predetermined cutoff frequencies. Adaptive filter adaptively cope with change

* Corresponding author.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 586 – 591, 2006.

of signal condition, however it made signal distortion accordance by inappropriate selection of convergent coefficient. For dynamic environment, various types of noise including muscle artifact noise and electrode moving artifact noise are in coupled and it made more difficult to remove them from original ECG signal. In this paper, to solve this problem, various noised signals are grouped into six categories by context estimation, and effectively reconfigured noise reduction filter. For noise estimation and signal clustering, neural networks are applied and genetic algorithms (GAs) are applied to dynamic noise reduction filter reconfiguration.

2 Noise Reduction Method by Context Estimation

Proposed noise reduction algorithm is mainly consists of five modules including noise context estimation module, control module for filter block design at running mode, 3stage filter block module, selection module for filter design at evolution mode, and decision module for fitness calculation at evolution mode as shown in Figure 1.

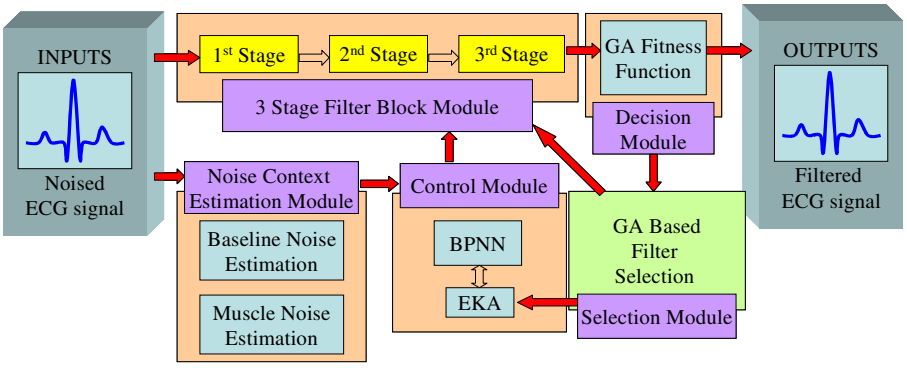


Fig. 1. Conceptual diagram of proposed adaptable noise reduction algorithm

2.1 Noise Context Estimation

Accurate modeling or estimation of ECG signal noise in dynamic environment is impossible since the signal artifacts by respiration, motion activity, and their coupled signal noise are not predictable. For this reason, to find environmental context based optimal noise reduction method, noised signals are grouped into several categories by context estimation. And then baseline wander noise and muscle noise are quantized as environmental context information.

2.1.1 Baseline Wander Noise Context Estimation

Conventionally, 1~50 Hz IIR band pass filter was widely used to remove baseline noise, but it made changes on important features including signal intervals and vertices. In this paper, noised ECG signals are low pass filtered with 2Hz cut-off frequency, and the gradient of filtered signal is calculated at every 100 samples to quantize the baseline wander noise component estimation as shown in Figure 2.

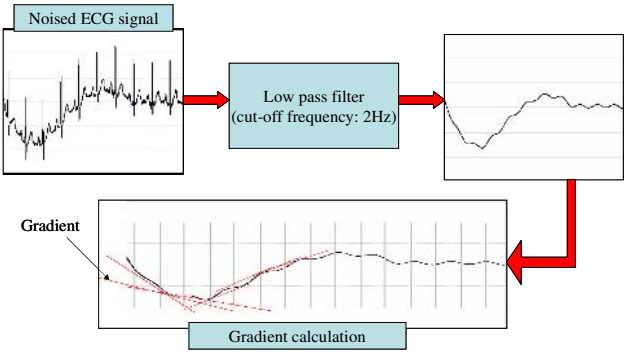


Fig. 2. Low pass filtering and gradient calculation for baseline wander noise component estimation

2.1.2 Muscle Noise Context Estimation

The muscle noise is one of the most difficult noise types to remove since it has non-stationary and non-linear nature. Typically Gaussian model is used to model muscle noise and recent research results showed that impulsive noise model is more accurate to model real-life muscle noise than conventional Gaussian model [3]. In this paper, noised signals are high pass filtered with 50Hz cut-off frequency and then impulse type components are considered as muscle noise components. As shown in Figure 3, the number of vertices above the heuristic determined threshold line is counted and this quantized information is used as muscle noise context information.

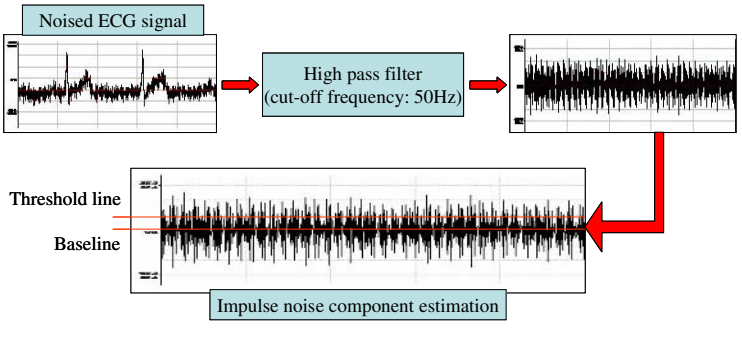


Fig. 3. High pass filtering and impulse noise estimation for muscle noise component estimation

2.2 Neural Networks Based Control Module and Filter Block Module

To find adaptable filter composition, the outputs of noise context estimation module are fed to inputs of control module. The control module is consists of two part, evolutionary knowledge accumulator (EKA) and neural network for filter design. EKA stores optimal filter design results for six-clustered dynamic measurement environments. For network construction, two types of outputs from context estimation module are used as network inputs and six outputs are used to cluster dynamic

measurement environment into six group. The 6 best filter combinations for each clusters are 0.8~45Hz band pass filter, 0.8Hz high pass filter, 50Hz low pass filter, combination of 0.8Hz high pass filter and 50Hz low pass filter, combination of wavelet interpolation filter (WIF) and 0.8~45Hz band pass filter, and combination of WIF, 0.8Hz high pass filter and 50Hz low pass filter. For clustering of environmental noise, three layered feed-forward error back-propagation neural network (BPNN) is applied and network weights were updated after each training vector (by "training by vector" method). Based on the systemic statistical experiment, network learning parameters (learning rate, momentum) and structure (number of layers, number of hidden neurons) were optimized.

2.3 Genetic Algorithm Based Decision Module and Selection Module

For continuous performance update, the every filtered output from 3-stage filter block is evaluated by fitness function at decision module. Fitness function is defined as detection percentage of five ECG signal features such as P-R interval, QRS duration, Q-T interval, R-R interval, and S peak point using So & Chan and Kim's feature extraction method [4]. If fitness value is over 0.8, ECG feature extraction results are considered as acceptable values. Or system assumes that designed filter cannot effectively remove noise, so GAs search new set of filter block for adaptable noise reduction. Based on GAs search results, EKA is updated and neural network for environmental context based clustering task is retrained.

3 Experimental Results

For performance evaluation, clinical records of the MIT-BIH arrhythmia database are used. It was the first generally available set for evaluation and it is widely used in

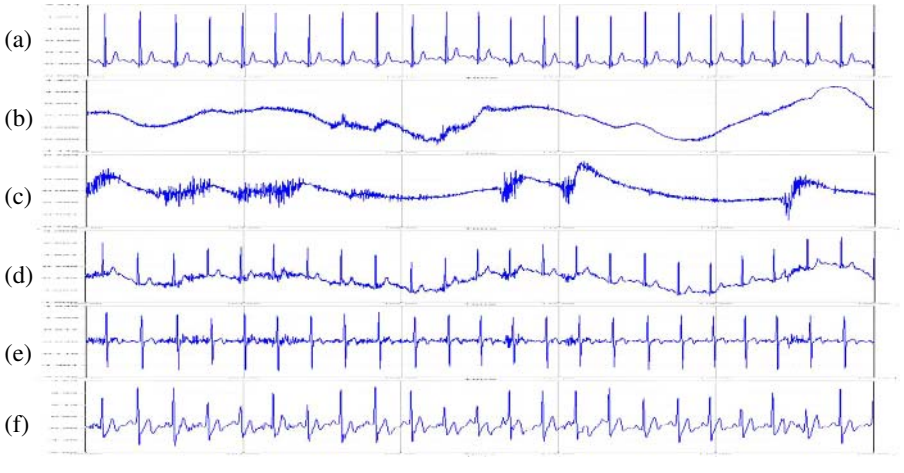


Fig. 4. Comparison of MIT-BIH (#103) ECG signal (a) original ECG signal (b) baseline wander noise (c) muscle artifact noise (d) noise added ECG signal (e) filtered signal by standard filter (f) filtered signal by proposed method

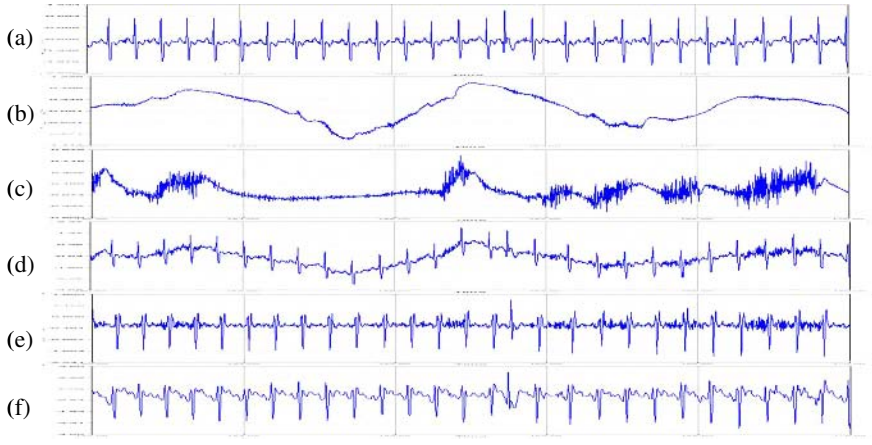


Fig. 5. Comparison of MIT-BIH (#118) ECG signal (a) original ECG signal (b) baseline wander noise (c) muscle artifact noise (d) noise added ECG signal (e) filtered signal by standard filter (f) filtered signal by proposed method

many research literatures [5]. Among MIT-BIH arrhythmia database, record #103 and #118 are selected and baseline wander noise and muscle artifact noise added data are also used from MIT-BIH noise stress test database. All the data are sample at 360Hz. For performance comparison, conventional standard filtered results are compared with proposed algorithm based filtered results. For conventional standard filter design, combination of 0.8Hz 20tab FIR high pass filter and 45Hz 20tab FIR low pass filter are used. Figure 4 and Figure 5 shows the comparison of noise reduction results. As shown in these figures, both methods remove baseline wander noise, but standard filter showed bigger S-T segment distortion because of remaining muscle artifact noise. To show the noise reduction performance, three ECG features (P duration, QRS duration, and T duration) of filtered signal are compared with features of original ECG signal. Feature extraction results of proposed method almost agreed with feature extraction values of original signal. Especially, proposed method showed significant improvement on extraction of T duration values by effective removal of muscle noise.

Table 1. Comparison of feature extraction results (unit: msec)

			P duration	QRS duration	T duration
MIT-BIH #103 (Normal)	Original signal		0.101563	0.0625	0.289062
	Standard filter	Measured value	0.023437	0.09375	0.1875
		Error	-0.078126	+0.03125	-0.101562
	Proposed method	Measured value	0.101563	0.073281	0.261718
		Error	0	+0.010781	-0.027344
MIT-BIH #118 (Left bundle branch block)	Original signal		0.070313	0.132812	0.277343
	Standard filter	Measured value	0.101563	0.109375	0.1875
		Error	+0.03125	-0.023437	-0.089843
	Proposed method	Measured value	0.066406	0.113282	0.278906
		Error	-0.003907	-0.01953	+0.001563

4 Conclusion

In this paper, adaptable ECG signal noise reduction algorithm for measurement in dynamic environment is developed. To remove baseline wander noise and muscle artifact noise, optimal filter set is selected by neural network based clustering using noise context. To show the performance, feature extraction results of proposed algorithm and conventional standard filter are compared and proposed method almost agreed with feature extraction results of original signal. With completion of this work, more accurate measurement of ECG during exercise is expected and it leads development of reliable automatic heart disease diagnosis system.

References

1. Park, K., Lee, K., Lee, B., Jeong, K., Yoon, H.: Design of A Wavelet Interpolation Filter for Elimination of Muscle Artifact in the Stress ECG, J. Biomedical Eng. Res. 21(5) (2000) 495-503
2. Yoon, S., Park, H., Park, J., Bae, E., Kim, Y.: An Implementation of Digital Signal Processing System for The Baseline Elimination, J. KIMICS 5(7) (2001) 1287-1294
3. Pancer, T. P.: A Suppression of an Impulsive Noise in ECG Signal Processing, Proc. 26th Annual Int'l Conf. IEEE EMBS (2004) 596-599
4. Kim, H., Min, C., Kim, T.: Feature Extraction of ECG Signal for Heart Disease Diagnosis, Proc. IEEK & KIEE Conf. Information and Control Systems (2004) 325-327
5. http://ecg.mit.edu

Mining the Independent Source of ERP Components with ICA Decomposition

Jia-Cai Zhang[1], Xiao-Jie Zhao[1], Yi-Jun Liu[2], and Li Yao[1]

[1] College of Information Science and Technology, Beijing Normal University, Beijing, China, 100875
{jiacai.zhang, yaoli}@elec.bnu.edu.cn
[2] McKnight Brain Institute, University of Florida, Gainesville, FL 32610, USA
yijunliu@psychiatry.ufl.edu

Abstract. Independent component analysis (ICA) can blindly separates the input ERP data into a sum of temporally independent and spatially fixed components arising from distinct or overlapping brain regions. In this study, we use ICA to illustrate that the P300 components in two ERPs recorded under various conditions or tasks are both mainly contributed from a few independent sources. ICA decomposition also indicates a new method to compare P300 components between two ERPs induced by two related tasks. Our comparisons are made on those independent sources contributed to the P300 components, rather than on the ERP waveforms directly. This novel approach identifies not only the similar or common independent components in both conditions that bring about a common part in ERP time courses, but also those different components induced by the different parts in ERP waveforms. Our study suggests that the ICA method is a useful tool to study the brain dynamics.

1 Introduction

The electrical activity of active nerve cells in the brain produces currents spreading through the head. These currents also reach the scalp surface and resulting voltage differences on the scalp can be recorded as the electroencephalogram (EEG). Due to the low signal to noise ratio of raw EEG, multi-trials of EEG are averaged together in order to boost the signal of interest and average out the (essentially random) noise. Event-related potentials (ERPs) are averages of EEG trials time-locked to a set of similar experimental events[1]. ERP waveform has some characteristic components such as P300. In order to study the independent sources of P300, we use ICA decomposition to separate the ERP time courses recorded on the scalp surface, which represents the composite of all neural events happening in the brain[2].

For the same subject, different brain currents can be induced spreading through the head under various conditions or tasks (experimental events), which produce different event-related potentials. Comparison of related but different ERPs is a widely used approach in studying brain dynamics[3]. However, the ERP waveform recorded at different scalp sensors is not always independent. It is a sum of temporally independent but spatially fixed components arising from distinct or overlapping brain or extra-brain sources.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 592 – 599, 2006.

In this study, comparisons focus on the P300 components in ERP waveforms. ICA decomposition is used as a preprocessing step, and the independent component sources that contribute largely to P300 are compared. this method can identify several similar or common independent components and several different components as well between two ERPs.

We will introduce two approaches for the application of ICA in ERP. One is to search for the independent sources of P300 components in ERP waveforms and the other is to compare the independent sources of P300 components under different conditions. Our results indicate that ICA may provide an encouraging and effective method to study brain dynamics from ERP.

2 Methods and Material

2.1 Data Recording

The ERP data were collected from the average of ten healthy children (11-12 years old) performing Stroop color-word task [4]. Our experiment has two conditions. In the 1st condition, subjects were presented with a series of color words, whose meaning and color are consistent. In the 2nd condition, the meaning and color are inconsistent. For example, if the word "green" is in green ink, it will be designated as the Consistent condition; if in red ink, it will be designated as the Inconsistent condition.

The ERP data used in this analysis was collected from 30 scalp electrodes, all of which were referred to the mean of the left and right mastoid. The sampling rate was 500 Hz. Every epoch lasts for 1000 milliseconds (500 sample points). The prestimulus lasting for 200 milliseconds (as baseline) has been eliminated.

2.2 ICA and Infomax ICA

ICA refers to a family of related algorithms[5]. ICA has recently been one of the most exciting topics in the fields of neural computation, advanced statistics, and signal processing[6][7][8]. ICA was originally proposed to recover N source signals s from the linearly mixed signals x except for two minor modifications: Amplitude scaling and ordering permutation[9][10][11]. Here A is an unknown n-by-n matrix, s is the independent sources.

$$x = \{x_1(t), x_2(t), \ldots, x_n(t)\} = As = A\{s_1(t), s_2(t), \ldots, s_n(t)\}. \quad (1)$$

Infomax ICA algorithm[12] is a self-organizing learning algorithm that maximizes the information transferred in a network of non-linear units. It is performed in three steps[13]: (1) centering the data on the origin; (2) “sphereing” the data by diagonalizing its covariance matrix; and (3) minimize redundancy between the outputs by minimize the mutual information between the components.

This algorithm does not assume any prior knowledge of the input distributions, and has proved useful in a wide range of biomedical applications[13]. Batch algorithms for ICA have also been used, such as FastICA[14]. The Informax ICA algorithm adopted in this paper is downloaded from EEGLab (The MATLAB toolbox package that performs the Informax ICA algorithm is included in EEGLab, which can be obtained from websites: http://www.sccn.ucsd.edu/eeglab/).

2.3 Pearson Correlation Coefficients

Comparison of original ERP data is often difficult. In order to study the difference in brain activation between different conditions, we can compare the ICA decomposition results. We know $x = As$ and each row of s means expression vectors of activation of independent components, and each column of A means the projection strength of the respective components onto scalp sensors. This projection is also displayed as scalp topographies. We may compare projection (column vectors in A), time courses(row vectors in s), or both. For convenience, the scalp maps from two ICA decompositions are stored in matrices A_1 and A_2 respectively, and independent component time courses are stored in matrices s_1 and s_2.

One of the most often used statistical quantities is Pearson correlation coefficient [15], r_{xy}, which measures the degree of (linear) interrelation between two sampled variables, x and y. x and y are usually measured over time and a typical aim in correlation analysis is to value the evidence for similarity of one time-dependent variable with the other, with the measures of strength and direction. Pearson correlation coefficient can be used to describe the similarity between two independent component sources based on following formula:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}. \tag{2}$$

3 Results and Discussion

3.1 Independent Component of P300

P300[16] (P3) is an obviously identifiable positive-going wave component always gets its peak around 300 milliseconds from stimulus ERP time courses. Here, P300 components reach its peak over the time 270-310 milliseconds under both the conditions. The other observation is all ERP waveforms correlate highly, especially after the stimulus occurs under both conditions.

The ICA decomposition results show that a few Independent components reach their peak along with P300 under each condition. We are interested in the question that whether these independent components contribute most to, or correspond to the independent sources in the P300 components of ERP?

In order to study the independent component sources of P300, only independent components in 4th, 7th, 11th and 16th rows of s_1 are reserved for the 1st condition (their peaks appear around 270-310 milliseconds); all other components are all eliminated by zeroing out the corresponding rows of the activation matrix s, and projecting the remaining components onto the scalp electrodes respectively. We obtained 4 scalp maps restored from them under both conditions (Fig 1). The most left panel is the time course of independent sources(row of s_1), next to the time courses are their projection length onto the scalp maps (column of A_1) and their contribution to scalp map(A_1 X s_1) around the time of p300 occurs. As for the independent component sources of the 2nd condition, we also found four independent component sources, i.e., the 5th, 8th, 9th and 17th components whose peak time are also coincident with P300.

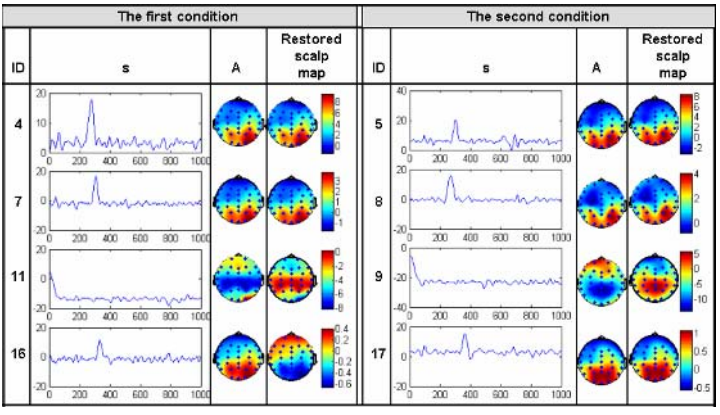


Fig. 1. The independent sources contributed to P300 and their projection on to the scalp

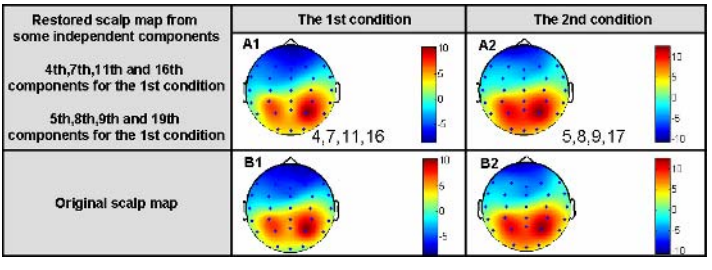


Fig. 2. The scalp map restored from a few independent components

Fig.2 shows that the scalp maps restored from the 4 independent components imitates the scalp map around the time 270-310 milliseconds of the original ERP. Fig.2.(A1) and Fig.2.(A2) are scalp maps reconstructed from 4 independent components of 1st condition. Fig.2.(B1) and Fig.2.(B2) are the averaged scalp maps over the time 270-310 milliseconds. The imitation between A1 and B1 indicates that 4 independent components contribute most to the P300 under the 1st condition. In the 2nd condition, the scalp map reconstructed from the projection of the four independent sources is also similar to the original scalp map around the same time.

The strong similarity indicates that P300 comes from only a few independent sources, and do without the other sources at this time. ICA decomposition reveals that only a few independent components contribute most to P300. This provides certain cues for searching independent origin sources of P300.

3.2 Comparison of ICA Decomposition

Direct comparison of time courses or scalp maps is difficult, because best-matching component pairs may not have the same component ranks or polarities. So we focused on the comparison between the independent sources mining by ICA, which contributing largely to the P300 components in ERP waveforms. The purpose of

comparing the ICA decomposition results is to find those independent components pairs which either make the similarity of P300 or differentiate the P300 components under two conditions. The similarity is measured by Pearson correlation coefficients.

Similar or common component pairs. Pearson Correlation Coefficients reveals that there are several pairs of independent components, whose projections and time courses are both highly correlated; while other independent components don not have similar or common components. Table 1. lists the similar pairs of components. They may be the similar or common sources which bring up the similarity of P300 components of different conditions.

Table 1. Pearson correlation coefficients of both the projection and the time courses of independent sources to P300 between different conditions

Component ID of the 1st Condition	Component ID of the 2nd Condition	Pearson Correlation Coefficients of the Projection	Pearson Correlation Coefficients of the Time Course
4	8	0.9155	0.8506
7	5	0.9868	0.7431

Similar projection vectors may indicate the same spatially fixed component source location or origination, while similar source time courses may indicate the simultaneity of components between different conditions. Both similar projection and time courses denote the spatial and temporal similarities of the components, and also result in similar contribution to the P300 components recorded on the scalp sensors.

In Fig. 3, the first row of the most right panel are the restored averaged scalp maps over the time 270-310 milliseconds, which were restored just from 2 independent sources (4th and 7th components) in the 1st condition, and below is the averaged restored scalp maps over the same time from 2 corresponding similar independent sources (5th and 8th components) in the 2nd condition. These two scalp maps are similar, indicating there are common or similar independent sources to P300 under both conditions.

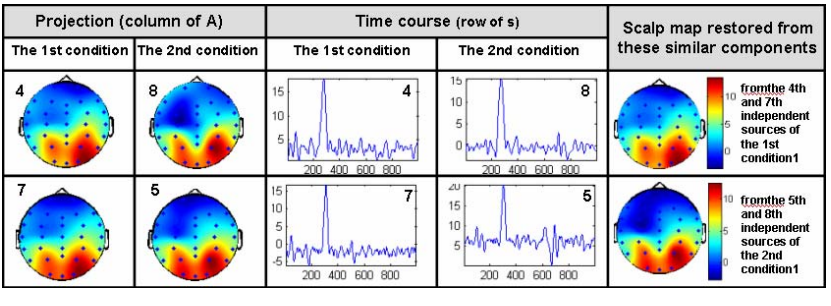


Fig. 3. Compares the similar independent sources to P300 between two conditions

Different component pairs. The difference in some regions of scalp maps Fig.2(B1) and Fig.(B2) tells us that there are some different independent sources contributing to P300, besides the similar or common components, and these different independent sources are different in the scalp maps, especially in the frontal area of the brain. The distinction of these sources may result from disparate activated brain regions (represented by A), or from different time courses of activation (represented by s).

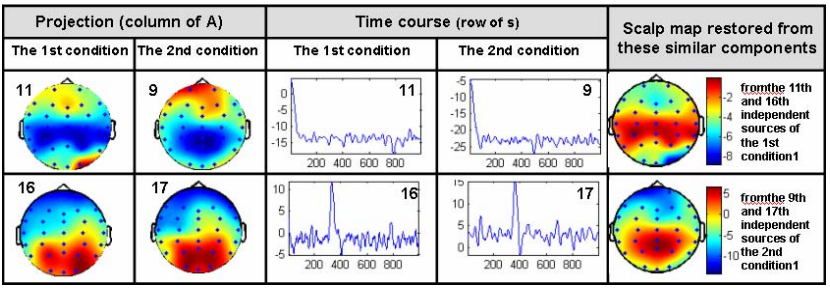


Fig. 4. Compares the different independent sources to P300 between two conditions

The most right panel of Fig.4 are the averaged restored scalp maps of independent sources of 11th and 16th components under the 1st condition, and the corresponding restored scalp maps of independent sources of 9th and 17th components under the 2nd condition. Fig.4 demonstrates the contributions of different independent component pairs can be well distinguished in both regional distribution and strength.

Fig.5 shows 4 differentiations of the scalp map of the 2nd conditions with (subtracted by) that of the 1st condition, (A) is the differentiation of the original scalp maps, (B) is the differentiation of scalp map restored from 5th, 8th, 9th, and 17th independent sources together under the 2nd condition with that restored from 4th, 7th, 11th and 16th sources under the 1st condition, (C) is the differentiation of scalp map restored from the similar or common component pair, (D) is the differentiation of scalp map restored from the different component pair.

It is obvious from the comparison of (A) and (B) in Fig.5 that a few independent components not only contribute most to the P300 components, but also reserve the detail of the P300 components, especially the difference between two conditions. (C) Illustrates that the difference from similar independent component pairs is not so obvious, i.e. they contribute almost the same to P300 components. This provide additional evidence that the 5th and 8th independent sources of the 2nd condition may

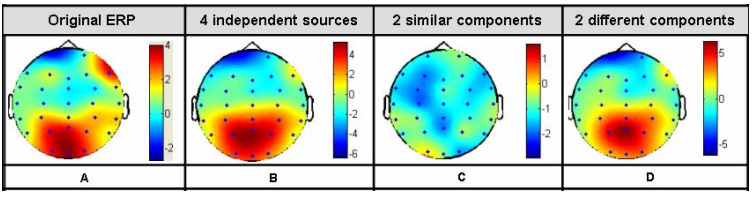


Fig. 5. Comparison of the differentiations between the scalp maps under different conditions

come from the same brain region or from the same psychological response as the 4th and 7th independent sources of the 1st condition. The observation of (D) is similar to (A) confirms that the different independent source pairs not only contribute primarily to the P300 components under each condition, but also make most of the difference.

4 Conclusion

Although the neural mechanisms that generate ERP are not fully known, the ICA algorithm can be applied in the analysis of ERP and our analyses implicate that ERP recorded at multiple scalp sensors are a linear sum of the scalp electrode-recorded activations generated by distinct sources. From the ICA results, we also show that most of the P300 components in ERP waveforms may be generated by a few distinct independent components. Combining the ICA decomposition results and Pearson correlation coefficients, we mine out common or similar independent component pairs and different independent components pairs under different experimental conditions. The former correspond to the similarity of the P300 components in ERP waveforms; the latter nicely explain the difference of the P300 components in ERP waveforms.

Acknowledgment. This study is funded by the National Natural Science Foundation of China. (60271014, 60472016).

References

1. Rosenfeld, J.P.: Event-related Potentials in Detection of Deception, Malingering, and False Memories. In: Murray Kleiner(ed.): Handbook of Polygraphy, N.Y.: Academic Press (2002)
2. Hyvärinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. John Wiley & Sons (2001)
3. Breznitz, Z.: Brain Activity During Performance of Naming Tasks: Comparison Between Dyslexic and Regular Readers. Scientific Studies of Reading 9(1) (2005) 17 – 42
4. MacLeod, C.M.: John Ridley Stroop: Creator of a Landmark Cognitive Task. Can. J. Psychol. 32 (1991) 521 – 524.
5. Makeig, S., Delorme, A., Westerfield, M., Jung, T.P., Townsend, J., Courchesne, E., Sejnowski, TJ: Electroencephalographic Brain Dynamics Following Visual Targets Requiring Manual Responses. PLOS Biology 2(6) (2004) 747 – 762
6. Oja, E.: Blind Source Separation: Neural Net Principles and Applications. In: Harold H.S., Mladen V.W., Barak A. P., Wim S.(eds): Proc. SPIE Defense & Security Symp. (2004) 1 – 14, Orlanda, USA
7. Hyvärinen, A., Oja, E.: Independent Component Analysis: Algorithms and Applications. Neural Networks 13(4-5) (2000) 411 – 430
8. Makeig, S., Debener, S., Onton, J., Delorme A: Mining Event-related Brain Dynamics. Trends in Cognitive Science 8(5) (2004) 204 – 210
9. Calinon, S., Billard, A.: Recognition and Reproduction of Gestures Using a Probabilistic Framework Combining PCA, ICA and HMM. Proc. ICML. ACM International Conference Proceeding Series, Vol. 119. ACM Press N.Y., USA (2005) 105 – 112
10. Liu, Z.Y., Kai, C.C., Xu L.: One-Bit-Matching Conjecture for Independent Component Analysis. Neural Computation 16(2) (2004) 383 – 399

11. Haykin S.: Signal Processing in a Nonlinear, NonGaussian, and Nonstationary World. Lecture Notes in Computer Science, Vol. 3445. Springer-Verlag, GmbH (2005) 43 – 53
12. Lee, T.W., Girolami, M., Sejnowski T.J..: Independent Component Analysis Using an Extended Infomax Algorithm for Mixed Sub-Gaussian and Super-Gaussian Sources. Neural Computation 11(2) (1999) 417 – 441
13. Makeig S., Westerfield M., Jung T.P., Enghoff S., Townsend J., Courchesne E., Sejnowski T.J.: Dynamic Brain Sources of Visual Evoked Responses. Science 295(1) (2002) 690 – 694
14. Maino D., Banday A.J., Baccigalupi C., Perrotta F., Górski K.M.: Astrophysical Component Separation of COBE-DMR 4-yr Eata with FASTICA. Monthly Notice of the Royal Astronomical Society 344(2) (2003) 544 – 552
15. Manfred M.: Estimating Pearson's Correlation Coefficient with Bootstrap Confidence Interval From Serially Dependent Time Series. Mathematical Geology 35(6) (2003) 651 – 665
16. Donchin, E., Coles, M.: Is the P300 Component a Manifestation of Context Updating? Behavioral Brain Sciences 11 (1988) 357-374

Multiple Signal Classification Based on Chaos Optimization Algorithm for MEG Sources Localization*

Jie-Ming Ma[1], Bin Wang[1], Yang Cao[2], and Li-Ming Zhang[1]

[1] Department of Electronics Engineering, Fudan University, Shanghai 200433, China
{042021026, wangbin, yang_cao, lmzhang}@fudan.edu.cn
[2] Department of Physiology and Biophysics, School of Life Science, Fudan University, Shanghai 200433, China

Abstract. How to localize the neural activation sources effectively and precisely from the magnetoencephalographic (MEG) recording is a critical issue for the clinical neurology and the study on brain functions. Multiple signal classification (MUSIC) algorithm and its extension referred to as recursive MUSIC algorithm are widely used to localize multiple dipolar sources from the MEG data. The drawback of these algorithms is that they run very slowly when scanning a three-dimensional head volume globally. In order to solve this problem, a novel MEG source localization method based on chaos optimization algorithm is proposed. This method uses the property of ergodicity of chaos to estimate the rough source location. Then combining with grids in small area, the accurate dipolar source localization is performed. Experimental results show that this method can improve the speed of source localization greatly and its accuracy is satisfactory.

1 Introduction

The spatio-temporal magnetoencephalography (MEG) produced by the brain can be measured by superconducting quantum interference device (SQUID) biomagnetometers which have extremely high sensitivity and low noise. Unlike the electric field, the magnetic field is insensitive to spherically symmetric spatial inhomogeneities in the conductivity profile of the skull. This insensitivity gives MEG the potential to determine the current distribution in the brain more simply and accurately than electroencephalography (EEG).

The multiple signal classification (MUSIC) algorithm [1] and its extension referred to as recursive MUSIC (R-MUSIC) [2] are widely used to localize multiple dipolar sources from the MEG data. In MUSIC and R-MUSIC methods, dipole is scanned through a grid confined to a three-dimensional head. Unfortunately, it is quite time-consuming. For example, if the head is modeled as a sphere that is centered at the origin of the Cartesian coordinate system and has a radius of 9 cm, considering only one quadrant we have to repeat 729000 times to locate one dipole with the precision of one millimeter. To overcome this problem, in this paper, a method based on chaos optimization algorithm (COA) is proposed to locate current dipoles quickly and precisely.

* This research was supported by the grant from the National Natural Science Foundation of China (No. 30370392).

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 600 – 605, 2006.

In section 2, we briefly review the problem formulation and describe the method based on COA. In section 3, we present some simulated results to show the excellent performance of the proposed method. Conclusion is given in section 4.

2 The Proposed Scheme

2.1 Forward Model

A complete MEG model is comprised of the model of primary current sources, head shape, magnetoconductivity, *etc*. [3]-[6]. For simplicity, in this paper, the primary current is modeled as a current dipole and the head is modeled as a concentric sphere. When the sensors are radially oriented, there are no effects of the volume currents, so the model can be further simplified.

In [2], the spatio-temporal independent topographies model is proposed which defines a source as one or more non-rotating dipoles with a single time course. We cluster the p dipoles into K subsets, and we can obtain the model in a matrix form

$$\boldsymbol{B} = \boldsymbol{A}(\boldsymbol{\rho},\boldsymbol{\theta})\boldsymbol{S}^T \ , \tag{1}$$

where $\boldsymbol{\rho} = [\boldsymbol{\rho}_1,\cdots,\boldsymbol{\rho}_K]$ is location parameter, and $\boldsymbol{\theta} = [\boldsymbol{u}_1,\cdots,\boldsymbol{u}_K]$ is orientation parameter, and $\boldsymbol{S}^T = [\boldsymbol{s}_1^T,...,\boldsymbol{s}_K^T]^T$ is intension parameter.

2.2 Cost Function

In [2], signal subspace method is introduced to yield the following cost function

$$\{c_1, c_2, \cdots, c_K\} = \text{subcorr}\left\{\boldsymbol{A}, \hat{\boldsymbol{\Phi}}_S\right\} \ , \tag{2}$$

where we refer to span ($\hat{\boldsymbol{\Phi}}_S$) as the signal subspace. $\{c_1, c_2, \cdots, c_K\}$ are subspace correlations which subject to $1 \geq c_1 \geq c_2 \cdots \geq c_K \geq 0$, and K is the number of independent dipoles.

The k^{th} dipole location can be estimated by

$$\hat{\boldsymbol{\rho}}_k = \arg\max_{\boldsymbol{\rho}} \text{subcorr}\left(\left[\hat{\boldsymbol{A}}_{k-1}\boldsymbol{G}(\boldsymbol{\rho})\right], \hat{\boldsymbol{\Phi}}_s\right)_k \ , \tag{3}$$

where

$$\hat{\boldsymbol{A}}_{k-1} = [\boldsymbol{a}(\hat{\boldsymbol{\rho}}_1,\hat{\boldsymbol{u}}_1),\cdots,\boldsymbol{a}(\hat{\boldsymbol{\rho}}_{k-1},\hat{\boldsymbol{u}}_{k-1})] = [\boldsymbol{G}(\hat{\boldsymbol{\rho}}_1)\hat{\boldsymbol{u}}_1,\cdots,\boldsymbol{G}(\hat{\boldsymbol{\rho}}_{k-1})\hat{\boldsymbol{u}}_{k-1}] \ . \tag{4}$$

$\boldsymbol{G}(\boldsymbol{\rho})$ can be considered to be the gain matrix or relationship between a unit moment source and the column vector of measurement locations.

In [2], the enumerative method is adopted. If we want to obtain high computational precision, we have to design a sufficiently dense grid in the volume of the head. But it is quite time-consuming to calculate at each grid point.

2.3 Method Based on COA

In this paper, COA is adopted to find the global maximum of $\text{subcorr}\left(\left[\hat{\boldsymbol{A}}_{k-1}\boldsymbol{G}(\boldsymbol{\rho})\right], \hat{\boldsymbol{\Phi}}_s\right)_k$.

A chaotic movement can go through every state in a certain area according to its own regularity, and every state is obtained only one time. COA is based on ergodicity, stochastic property, and "regularity" of chaos, which avoid being trapped in local minima during scanning [7]. Here we modify the algorithm to adapt to the localization problem.

Chaos variables are generated by the well-known logistic map

$$u^{t+1} = \mu u^t (1-u^t) \ , \tag{5}$$

where superscript denotes iterative time, and μ is a control parameter. We let $\mu = 4$ to generate chaotic evolution.

The procedures of performing COA are as follows:

Step 1. Initialize u_i^0 , $u_i^0 \in (0,1)$, $i = 1, 2, 3$, which have small differences to generate three chaos variables. NUMBER is specified, and let $f = \text{subcorr}\left(\left[\hat{\boldsymbol{A}}_{k-1}\boldsymbol{G}(\boldsymbol{\rho})\right], \hat{\boldsymbol{\Phi}}_s\right)_k$.

Step 2. u_i^0 is mapped into the variance ranges by the following equation: $x_i^0 = c_i + d_i u_i^0$. And let $x_i^* = x_i^0$, $f^* = f(x_1^*, x_2^*, x_3^*)$, $k = 0$.

Step 3. Let $u_i^{t+1} = 4u_i^t(1-u_i^t)$.

Step 4. Let $x_i^{t+1} = c_i + d_i u_i^{t+1}$.

Step 5. If $f(x_1^{t+1}, x_2^{t+1}, x_3^{t+1}) \geq f^*$, let $x_i^* = x_i^{t+1}$, $i = 1, 2, 3$, and $f^* = f(x_1^{t+1}, x_2^{t+1}, x_3^{t+1})$, and set $k = 0$;otherwise set $k = k + 1$.

Step 6. If $k \geq$ NUMBER, go to Step 7; Otherwise, return to Step 3.

Step 7. End.

How to choose parameter d_i is important. When localizing MEG sources with the sphere head model of radius of 9 cm, if we let d_i be 9, it will take long time to localize the sources in deep locations because the NUMBER should be set large. In order to improve localization speed, we let d_i be a variable ranging from 1 to 9 with the interval 1. For each d_i, we calculate { f^*, x_1^*, x_2^*, x_3^* }, and then $\left\{x_1^*, x_2^*, x_3^*\right\}$ which generates the maximal f^* is the best solution.

We assume that $\{x_1, y_1, z_1\}$ is the result of the rough localization. Then we refine it by combining with grids in small areas. That is to say, we calculate at each voxel in the region from $x = x_1 - h$ cm to $x = x_1 + h$ cm, from $y = y_1 - h$ cm to $y = y_1 + h$ cm, and from $z = z_1 - h$ cm to $z = z_1 + h$ cm, at 0.1cm intervals, where parameter h controls the region of the second search. Combining COA and grids, we can not only control the precision easily but also speed up localization greatly.

3 Experimental Results

Simulated signals are used to evaluate the performance of the proposed method. We implement the program using Matlab 6.5 on a PC with a Pentium(R) 4 1.7G CPU. And a standard arrangement of 37 radial sensors is used here. The array has one sensor at θ=0, a ring of six sensors at $\theta = \pi/8$, $\varphi = k\pi/3$, $k = 0, \ldots, 5$, a ring of twelve sensors at $\theta = \pi/4$, $\varphi = k\pi/6$, $k = 0, \ldots, 11$, and a ring of eighteen sensors at $\theta = 3\pi/8$, $\varphi = k\pi/9$, $k = 0, \ldots, 17$. They are distributed on the upper region of a 9 cm single-shell sphere as shown in Fig. 1.

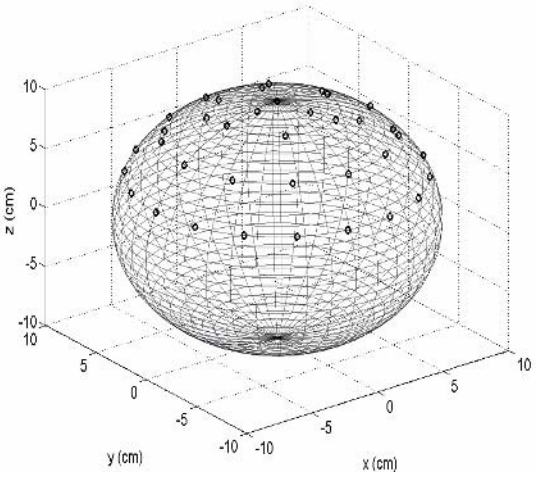


Fig. 1. The location distribution of 37 sensors

Here, a 3-dipole case is considered. For simplicity, we fix the orientation of each source and assign each an independent time series as follows:

$$s_1(t) = \sin 10\pi t\ , \quad s_2(t) = \sin 12\pi t\ , \quad s_3(t) = \sin 14\pi t\ .$$

Using the model described in section 2.1, we produce a 37×500 spatio-temporal data set, and then add white Gaussian noise on each sensor channel for an SNR of 15 dB. This simulated noisy data set is used as the measured data to localize the sources. In COA, u_i^0 is generated randomly, and we let c_i be 0.1, NUMBER be 2000, and h be 0.5.

For comparison, we localize sources by R-MUSIC and the method proposed in this paper, respectively.

Before localization, the number of dipoles should be determined. We perform an eigenvalue decomposition of $\boldsymbol{R}_F$ and plot these eigenvalues. The first ten of them are shown in Fig. 2. The 4th eigenvalue is almost zero, and it gives us a clear indication that the number of elemental dipoles is 3.

Table 1 displays the results of a 1 mm grid R-MUSIC scan. The results show that the accuracy is satisfactory but it takes 31555 seconds to finish the localization. Table 2 is the results of the rough localization by COA. The estimated locations are close to the true locations with the largest error of 0.2489 cm. The execution time is only 1515.5 seconds, about 1/20 of R-MUSIC's. Then, we locate dipoles by combining COA with grids. Table 3 shows that the total execution time is 1552.6 seconds. In comparison with Table 1, our method has the same results as R-MUSIC's, but the speed is improved greatly. The execution time is about 1/20 of R-MUSIC's.

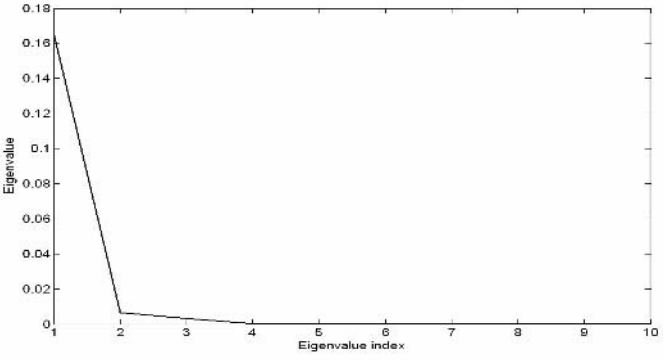


Fig. 2. The eigenvalues of the simulated data. Only the first 10 of 37 eigenvalues are plotted. That the 4th eigenvalue is almost zero indicates that the number of elemental dipoles is 3.

Table 1. Localization results by R-MUSIC algorithm

	True Location (cm)			Estimated Location (cm)			Time (s)
	x	y	z	x	y	z	
Dipole1	6.2421	3.0060	4.0000	6.2	3.0	4.0	
Dipole2	2.4354	2.4354	8.3149	2.4	2.4	8.3	31555
Dipole3	5.4641	1.4641	5.6569	5.5	1.5	5.7	

Table 2. Rough localization results using COA

	True Location (cm)			Estimated Location (cm)			Error (cm)			Time (s)
	x	y	z	x	y	z	x	y	z	
Dipole1	6.2421	3.0060	4.0000	6.1051	3.0647	3.8097	0.1370	0.0587	0.1903	
Dipole2	2.4354	2.4354	8.3149	2.4900	2.5107	8.4021	0.0546	0.0753	0.0872	1515.5
Dipole3	5.4641	1.4641	5.6569	5.3275	1.2152	5.5896	0.1366	0.2489	0.0673	

Table 3. Localization results using COA combining with grids

	True Location (cm)			Estimated Location (cm)			Time (s)
	x	y	z	x	y	z	
Dipole1	6.2421	3.0060	4.0000	6.2	3.0	4.0	
Dipole2	2.4354	2.4354	8.3149	2.4	2.4	8.3	1552.6
Dipole3	5.4641	1.4641	5.6569	5.5	1.5	5.7	

There are several reasons for our method to have such good performances. Firstly, COA is not like some stochastic optimization algorithms that escape from local maxima by accepting some bad solutions according to a certain probability. It searches on the regularity of chaotic motions so that it can more easily escape from local maxima. Thus, COA has high search efficiency. Secondly, in order to improve the computation speed, we lower the goal and estimate the rough source locations as the arguments which are close to the global maximum of the cost function. Therefore, the NUMBER can be set small and the second carrier wave can be omitted. The accurate dipolar source localization can be performed by combining with grids in small areas. In addition, the simple iterative formula of COA also saves computational time.

4 Conclusion

In this paper, we proposed a MEG source localization method based on COA. The experimental results from the simulation show that the source localization operation can be speeded up greatly. Further, combining with grids in small areas, we can obtain more accurate results. The localization of MEG sources based on COA precisely and quickly will contribute to its further applications.

References

1. Mosher, J.C., Lewis, P.S., Leahy, R.M.: Multiple Dipole Modeling and Localization from Spatio-Temporal MEG Data. IEEE Trans. Biomedical Engineering **39**(6) (1992) 541-557
2. Mosher, J.C., Leahy, R.M.: Recursive MUSIC: a Framework for EEG and MEG Source Localization. IEEE Trans. Biomedical Engineering **45**(11) (1998) 1342-1354
3. Mosher, J.C., Leahy, R.M., Lewis, P.S.: EEG and MEG: Forward Solutions for Inverse Methods. IEEE Trans. Biomedical Engineering **46**(3) (1999) 245-259
4. Cuffin, B.N.: Effects of Head Shape on EEG's and MEG's. IEEE Trans. Biomedical Engineering **37**(1) (1990) 44-52
5. Crouzeix, A., Yvert, B., Bertrand, O., Pernier, J.: An Evaluation of Dipole Reconstruction Accuracy with Spherical and Realistic Head Models in MEG. Clinical Neurophysiology **110**(12) (1999) 2176-2188
6. Schimpf, P.H., Ramon, C., Haueisen, J.: Dipole Models for the EEG and MEG. IEEE Trans. Biomedical Engineering **49**(5) (2002) 409-418
7. Li, B., Jiang, W.S.: Chaos Optimization Method and Its Application. Control Theory and Applications **14**(4) (1997) 613-615

Automatic Segmentation of Putamen from Brain MRI

Yihui Liu[1,3], Bai Li[1], Dave Elliman[1], Paul Simon Morgan[2], and Dorothee Auer[2]

[1] School of Computer Science & IT, University of Nottingham,
Jubilee Campus, Nottingham, UK NG8 1BB
yxl@cs.nott.ac.uk
[2] Academic Radiology,University of Nottingham,
Queen's Medical Centre, Nottingham, UK NG7 2UH
[3] School of Computer Science, Shandong University of Light Industry,
Jinan, Shandong, China 250100
yihui_liu@hotmail.com

Abstract. In this paper we present an automatic segmentation of the Putamen shape from brain MRI based on wavelets and a neural network. Firstly we detect the Putamen region slice by slice using 1D wavelet feature extraction. Then fuzzy c-means technology is combined with edge detection to segment the objects inside the Putamen region. Finally features are extracted from the segmented objects and fed into a neural network classifier in order to identify the Putamen shape. Experiment shows the segmentation results to be accurate and efficient.

1 Introduction

Nuclear magnetic resonance is recognized as an important tool for clinical diagnosis. The main task in MR imaging is to extract useful information for diagnostic purposes from the images. Traditionally, this task was accomplished by radiologists; however increasingly their skills are enhanced by image processing as the quality and speed of computer algorithms are rapidly improving.

At present several methods are used for tissue classification using multi-spectral MR images; discriminant analysis [1,2], fuzzy methods [3], neural networks [4,5], knowledge-based techniques [6], shape-based methods [7], multivariate principal component analysis methods [8] and statistical pattern recognition [9,10].

Amini et al. [11] presented an automated method to segment the thalamus from magnetic resonance images (MRI) based on a discrete dynamic contour model. Internal forces deforming the model are calculated from local geometry. Because the thalamus has low contrast and discontinuous edges using MRI, external forces are improved based on fuzzy c-means (FCM) unsupervised clustering, the Prewitt edge-finding filter, and morphological operators.

In this research we use wavelet analysis and neural network to detect the region including the Putamen area. The automatic detection of the Putamen region is carried out slice by slice. After locating the Putamen region, the fuzzy c-means cluster technology is combined with Canny edge detection to segment the objects inside the Putamen region. Finally the neural network is used to classify the Putamen shape based on simple moment features extracted from the binary objects inside Putamen region.

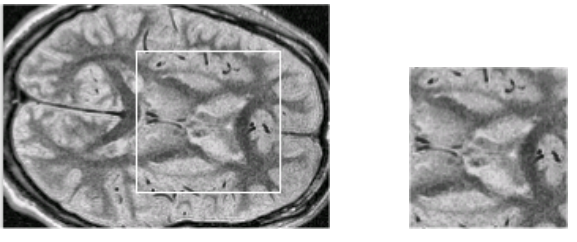
J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 606 – 613, 2006.

2 Detection of the Putamen Region Slice by Slice

2.1 Feature Extraction Based on Wavelets

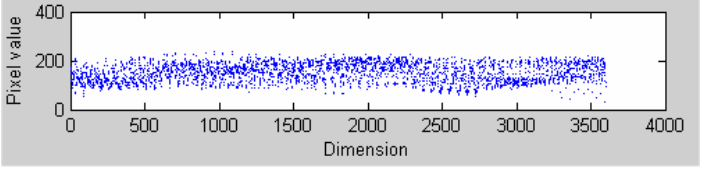
The 1D wavelet decomposition process is described below. Starting from the data s, the first step produces two sets of coefficients; approximation coefficients cA_1, and detail coefficients cD_1. These coefficients are computed by convolving s with the low-pass filter for approximation coefficients, and with the high-pass filter for detail coefficients. The coefficients are down-sampled by keeping the even indexed elements. Then the approximation coefficients cA_1 are split into two parts by using the same algorithm and are replaced by cA_2 and cD_2, and so on. This decomposition process is repeated until the required level is reached.

We use an MRI proton density image to segment the Putamen shape. Figure 1(a) shows the slice including Putamen and Figure 1(b) shows the sub-region including the Putamen.

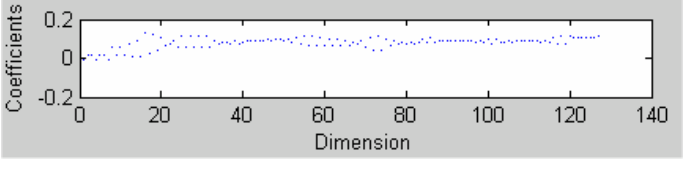


(a) (b)

Fig. 1. (a) The proton-density image (b) The region including Putamen



(a) 1D representation for Putamen region



(b) Extracted wavelet features

Fig. 2. (a) The 1D representation for Putamen region (b) The extracted wavelet features

The sub-region including Putamen is called the Putamen region. Firstly this region is re-sized to 60 by 60 pixels. Its 1D signal representation then has 3600 dimensions. After a 5 level wavelet 1D decomposition, the 5th level low frequency coefficients is obtained to form the feature space. If the dimension number of 1D data is l_{data}, the number of coefficients at the 5th level is l_{coef}, where $l_{coef} = l_{data} / 2^5$. In this research a bi-orthogonal wavelet is used for wavelet analysis. Considering the effect of border distortion, features of 127 dimensions are extracted from data of 3600 dimensions. Figure 2 shows the 1D representation of Putamen region and extracted feature vector.

2.2 Classification of Putamen Region Using a Neural Network

A backpropagation neural network was chosen to classify Putamen and non-Putamen regions. The basic backpropagation algorithm adjusts the weights in the steepest descent direction. This direction is the negative of the gradient and the performance improves most rapidly alone this direction. However, this does not necessarily result in the fastest convergence. For performance reasons we chose the scaled conjugate gradient algorithm to train the network. A search is performed along conjugate directions, which generally results in faster convergence than the steepest descent approach. For the training algorithms, a learning rate is used to determine the magnitude of the weight update. In most of the conjugate gradient algorithms, the step size is adjusted at each iteration, and this involves a search that is computationally expensive. The scaled conjugate gradient algorithm (SCG), developed by Moller [12], avoids a time consuming search by using a step size scaling mechanism and shows super-linear convergence on most problems.

To train the neural network, seven slices including the Putamen region from MRI brain were used for 60x60 sub-windows. During training, a sub-window of the Putamen region is manually selected and two sub-windows of slightly different size taken and scaled to the 60x60 size for training (Figure 1). The networks are also trained with non-Putamen images. Unlike sub-window of the Putamen region, it is impossible to define a non-Putamen model. A training set of non-Putamen regions was first collected randomly. After training, the network is used to detect the Putamen area on images. If the neural network fails to classify a non-Putamen region, then this sub-window is added to the non-Putamen image training set. This process is repeated many times on other training images until the correct detection results are achieved. Where the Putamen region has a different size, the sub-sampling of the each slice has to be repeated at different scales. Each slice is used to generate an image sequence and the sizes of two consecutive images in the sequence differ by a factor of 0.9 and 5x5 grid points are selected to detect the Putamen and non Putamen regions. The number of images in the sequence depends on the original slice size. The detection of Putamen region is applied to all the images in the sequence, and a sub-window is identified as a candidate Putamen region if it generates a neural network output activation value above 0.85. Otherwise the sub-window is identified as a non-Putamen image. If two sub-windows in consecutive images of the sequence are both identified as candidate Putamen region and the sub-windows overlap each other by more than 90%, then the sub-windows are confirmed as Putamen regions. Some results are shown in Figure 3.

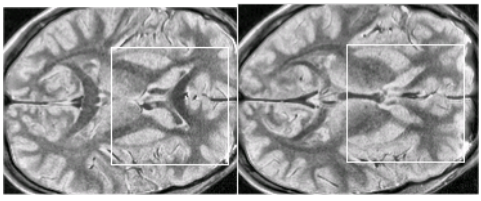


Fig. 3. The detection results of the Putamen region

The backpropagation network used two hidden layer. The input layer had 127 units corresponding to the wavelet feature vector. The first hidden layer has 90 neurons and the second hidden layer has 30 neurons. The output layer has two neurons. The log-sigmoid transfer function is used for this multilayer neural network.

3 Object Segmentation Inside the Putamen Region

We performed a contrast-limited adaptive histogram equalization to enhance the contrast of Putamen region. Unlike histogram equalization, this operated on small data regions rather than on the entire image. The contrast transform function is calculated for each of these tile regions individually. The number of tile rows and columns is 8. A uniform distribution is used as the basis for creating the contrast transform function. After enhancing the image, a Gaussian low-pass filter of size [3 3] with standard deviation σ (0.5) was used to smooth the image. The results of contrast-limited adaptive histogram equalization and de-noising of Putamen region are shown in Figure 4(b). The segmentation of image data aims to partition the image space into different cluster regions with similar intensity values. Most medical images present overlapping grey-scale intensities for different tissues. The image data is classified into two clusters using the fuzzy c-means method [13]. The fuzzy clustering results

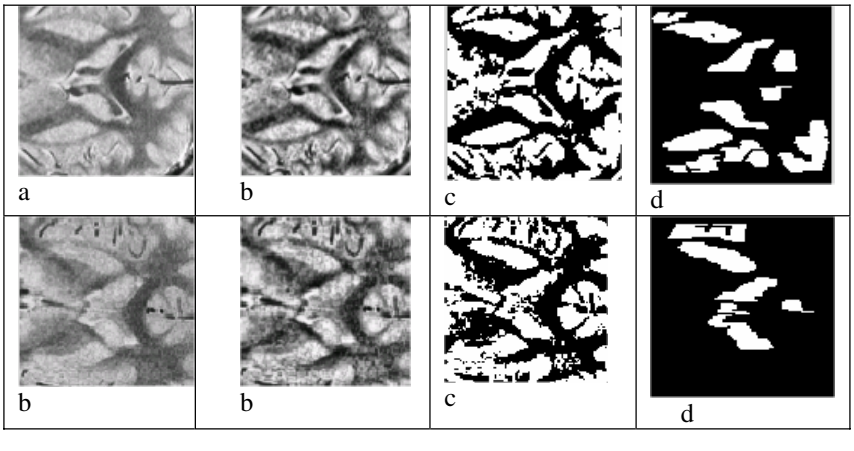


Fig. 4. (a) The original image (b) The contrast-limited adaptive histogram equalization & Gaussian lowpass filter (c) The classification of two clusters (d) A morphological operation

are shown in Figure 4(c). The morphological operations of dilation and erosion were used in combination to implement image processing operations for binary objects in order to segment objects inside the Putamen region. Figure 4(d) shows the results of these morphological operations.

For sample 2, after the fuzzy c-means clustering operation and morphological operation, the Putamen shape is not segmented clearly. Canny edge detection is performed to segment the connected part of the Putamen [14]. The operation is performed as below:

$$I_{new} = I_{ori} \& \sim E_{canny}$$

where E_{canny} is the result of Canny detection, I_{ori} is the result of the fuzzy c-means operation and I_{new} is new result.

The results are shown in Figure 5.

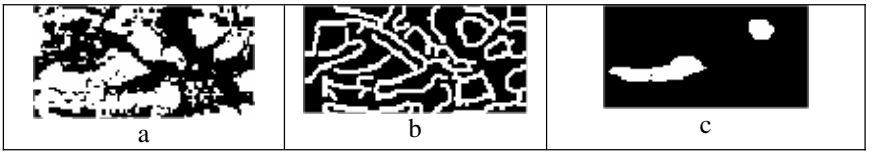


Fig. 5. (a) The result of fuzzy c-means operation (b) The result of Canny edge detection (c) The result of morphological operation for I_{new}

4 Classification of the Putamen Shape

After the segmentation of Putamen region, the identification of Putamen is needed. A backpropagation neural network based on scaled conjugate gradient network is selected to perform the classification of Putamen shape. Features are extracted from the objects in the binary image of Putamen region.

Two moment-based features were selected for classification. For the objects in the selected region, we define the description as below:

$$I(i, j) = 1, (i, j) \in C ,$$

where C is the set of points (i, j) inside the objects of interest.

The shape of the objects can be described through the moments. We define the geometric moments as follows:

$$m_{pq} = \sum_i \sum_j i^p j^q \qquad (i, j) \in C$$

The central moments, which are invariant to translation are defined below:

$$u_{pq} = \sum_i \sum_j (i - \bar{x})^p (j - \bar{y})^q, \quad (i, j) \in C$$

where $\bar{x} = \dfrac{m_{10}}{m_{00}}$, $\bar{y} = \dfrac{m_{01}}{m_{00}}$.

Two useful features that are related to these moments provide useful discriminant information. One is 'Orientation', which is the angle between the x-axis and the major axis of the ellipse that has the same second-moments as the region. The orientation θ is defined as below:

$$\theta = \frac{1}{2}\tan^{-1}\left[\frac{2u_{11}}{u_{20}-u_{02}}\right].$$

The second is 'Eccentricity', which is the eccentricity of the ellipse that has the same second-moments as the region. The value is between 0 and 1. An ellipse whose eccentricity is 0 is a circle, while an ellipse whose eccentricity is 1 is a line. The eccentricity is defined as below:

$$\xi = \frac{(u_{20}-u_{02})^2 - 4u_{11}^2}{(u_{20}+u_{02})^2}$$

Figure 6 shows the ellipse description of objects inside Putamen region.

Fig. 6. The ellipse description of objects inside Putamen region

Another feature extracting from the objects is the 'Solidity', which is obtained by dividing the actual number of pixels in the region by the number of pixels in the convex hull. The convex hull of an arbitrary set is the smallest convex set containing this set of pixels. The length (in pixels) of the major axis of the ellipse and the length (in pixels) of the minor axis of the ellipse are also selected for feature vectors.

The input layer has 5 neurons, which represent the 5 features of the binary object. The hidden layer has 40 neurons and the output layer has 2 neurons to decide if the binary object is one of Putamen shape.

5 Discussion

Two brain samples were used for tests. Some slice results are shown in Figure 7. The results were obtained efficiently and are correct. The proposed Putamen segmentation method based on wavelets and a neural network has the advantage of directly and simultaneously illustrating the brain Putamen structure in MRI brain images. This is especially useful to the radiographer where the Putamen structure is not easily distinguished due to a small intensity variation. The proposed method can be combined with other image segmentation techniques at various resolutions in order to increase the accuracy of quantification of the Putamen region in the brain.

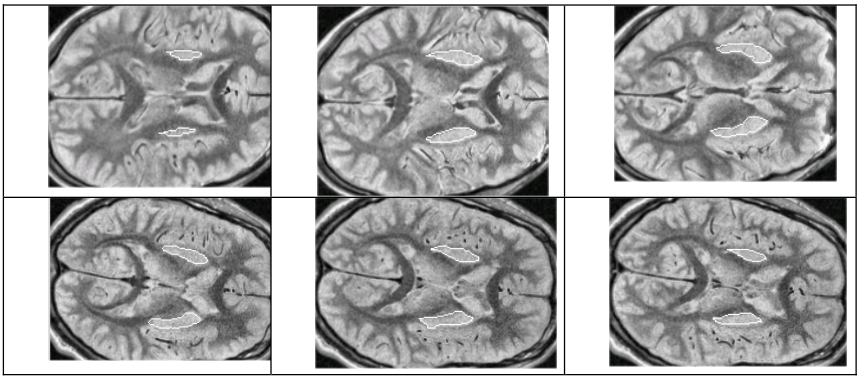


Fig. 7. The segmentation results of the Putamen for two brain cases

References

1. DeCarli, C., Murphy, D.G.M., McIntosh, A.R., Teichberg, D., Shapiro, M.B., Horwitz, B.: Discriminant Analysis of MRI Measures as A Method to Determine the Presence of Dementia of the Alzheimer Type. Psychiatry Res. 57(2) (1995) 119–130
2. Zavaljevski, A., Dhawan, A.P., Gaskil, M., Ball, W., Johnson, J.D.: Multi-Level Adaptive Segmentation of Multi-Parameter MR Brain Images. Comput Med. Imag. Graphics 24(2) (2000) 87–98
3. Suckling, J., Sesson, T., Greenwood, K., Bullmore, E.T.: A Modified Fuzzy Clustering: Algorithm for Operator Independent Brain Tissue Classification of Dual Echo MR Images. Magn. Reson. Imag. 17(7) (1999) 1065–1076
4. Alirezaire, J., Jernigan, M.E., Nahmias, C.: Automatic Segmentation of Cerebral MR Images Using Artificial Neural Networks. IEEE Trans. Nucl. Sci. 45(4) (1998) 2174–2182
5. Reddick, W.E., Glass, J.O., Cook, E.N., Elkin, T.D., Deaton, R.J.: Automated Segmentation and Classification of Multispectral Magnetic Resonance Images of Brain Using Artificial Neural Networks. IEEE Trans. Med. Imag. 16(6) (1997) 911–918
6. Clark, M.C., Hall, L.C., Goldgof, D.B., Velthuizen, R., Murtagh, F.R., Silbiger, S.: Automatic Tumor Segmentation Using Knowledge-Based Techniques. IEEE Trans. Med. Imag. 17(2) (1998) 187–201
7. Duta, N., Sonka, M.: Segmentation and Interpretation of MR Brain Images: An Improved Active Shape Model. IEEE Trans. Med. Imag. 17(6) (1998) 1049–1062
8. Antalek, B., Hornak, J.P., Windig, W.: Multivariate Image Analysis of Magnetic Resonance Images with the Direct Exponential Curve Resolution Algorithm (DECRA): Part 2. Application to Human Brain Images. Journal of Magnetic Resonance. 132(2) (1998) 307–315
9. Andersen, A.H., Zhang, Z., Avison, M.J., Gash, D.M.: Automated Segmentation of Multispectral Brain MR Images. Journal of Neuroscience Methods. 122(1) (2002) 13–23
10. Bezdek, J.C., Hall, L.O., Clarke, L.P.: Review of MR Image Segmentation Techniques Using Pattern Recognition. Med Phys. 20(4) (1993) 1033–1048
11. Amini, L., Soltanian-Zadeh, H., Lucas, C., Gity, M.: Automatic Segmentation of Thalamus from Brain MRI Integrating Fuzzy Clustering and Dynamic Contours. IEEE Transactions on Biomedical Engineering. 51(5) (2004) 800 – 811

12. Moller, M. F.: A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. Neural Networks. 6 (4) (1993) 525-533
13. Styner, M., Brechbuhler, C., Szckely, G., Gerig, G.: Parametric Estimate of Intensity Inhomogeneities Applied to MRI. IEEE Trans. Med. Imaging. 19(3) (2000) 153–165
14. Canny, J.: A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 8(6) (1986) 679-698

A Neural Network Model for the Estimation of Time-to-Collision

Ling Wang[1], Hongjin Sun[2], and Dezhong Yao[1]

[1] Center of NeuroInformatics, School of Life Science and Technology, University of Electronic Science and Technology of China, Chengdu, 610054, China
dyao@uestc.edu.cn
[2] Department of Psychology, Neuroscience and Behaviour, McMaster University, Hamilton, Ontario, L8S 4K1, Canada

Abstract. Artificial Neural Networks (ANNs) which are derived from Biological Neural Networks (BNNs) are enhanced by many advanced mathematical techniques and have become powerful tools for solving complicated engineering problems. Integrating BNNs with mature ANNs is a very effective method of solving intricate biological problems and explaining neurophysiological data. In this paper we propose a neural network model that explains how the brain processes visual information about impending collisions with an object - in particular, how time-to-collision information is caculated in the brain. The model performs extremely well as a result of incorporating physiological data with the methods involved in the development of ANNs. By implementing this novel compuational neural network model, the results of the simulation demonstrate that this integrative approach is a very useful and efficient way to deal with complicated problems in neural computation.

1 Introduction

Artificial neural networks (ANNs) that are derived from biological neural networks (BNNs), are often generated through the integration of many advanced techniques, e.g., mathematic methods, computer science, artificial intelligence etc. ANNs have become very useful tools for solving many complex engineering problems. This is partly because ANNs have many highly desirable and efficient properties and capabilities including: nonlinearity, input-output mapping, adaptivity, evidential response, contextual information, fault tolerance, VISI implementability, uniformity of analysis and design, and neurobiological analogy.

The biological brain can be considered as a highly complex, nonlinear, and parallel computer (information-processing system). The brain has the capability to organize its structural constituents, known as neurons, as a way of constructing elaborate neural networks that perform certain computations (e.g., pattern recognition, perception and motor control, etc) at speeds many times faster than the fastest digital computer in existence today. Neural networks play a central role in an animals survival by virtue of the flexible manner in which they are able to interact with a complex and dynamic

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 614 – 619, 2006.

environment. While neurophysiological data provides information regarding the communication between individual neurons, computational modeling approaches provide important insights in the communication that occurs within an entire network of neurons.

One important application of neural networks is in the field of computational vision. The visual system provides us with a representation of the environment around us and more importantly guides us in effectively interacting with the environment. One relevant area of vision research relates to the neural computations involved in the processing of visual motion in 3-dimentional space. For example, behaviorally, it has been demonstrated that the TTC (Time-to-Collision - that is, the time elapsed before a looming object on collision course will reach the eye) is of profound importance to an animals survival by allowing them to avoid impending danger and escaping as soon as possible [1]. But how the brain processes the visual information to obtain TTC is less known. Based on the results of well-grounded artificial neural network techniques and some electrophysiological data on birds [2,3], we propose a novel computational neural network model to simulate the brain mechanisms involved in the information processing of visual motion-in-depth, including the estimation of TTC.

2 TTC-Neural Network

2.1 Physiological Basis of the Visual Processing of TTC

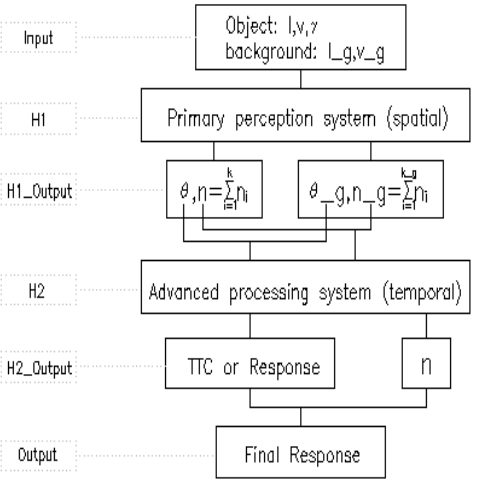
For an animal, it is critical to be able to respond quickly and accurately to a looming object on a collision course. When an object moves towards the eyes, its retinal image size increases, and when it moves away, its retinal image size decreases. Dynamically changing the retinal image size, even of a stationary object, can produce a sensation of motion-in-depth [4,5]. Lee [5] proposed that the expansion of the retinal image of an approaching object could trigger a behavioral response and that the precise timing of the response is controlled by the optical variable called τ which happens to be equal to TTC. If the object is approaching at a constant speed, τ is equal to the inverse of the relative rate of expansion of the looming object. This can be expressed as $\tau = sin\theta/\theta' \approx \theta/\theta'$, where θ is the visual angle subtended by the looming object and θ' is the derivative of θ over time. While, there are many behavioural studies suggesting that τ can be a very useful source of information for various aspects of visual-motor control in humans and animals [1], very few studies have illustrated how the brain might actually compute such information.

It has been discovered that particular neurons in the pigeon brain selectively respond to displays of an object moving on a collision course directly towards the eye [2]. More importantly, the onset of the responses always occurs at a constant time before the object would reach the bird. This remained true for various object movement velocities and various object sizes, suggesting that these neurons encode τ by responding to a threshold value of τ [1, 2, 6]. Although it has been demonstrated empirically that such neurons can perform the computation, it is not clear how the brain BNN is organized to process this visual information to obtain TTC information.

2.2 The Framework of the Integrated Network

On the basis of both the ANN techniques and the electrophysiological findings related to collision detections in pigeons [1,2], we have previously reported some simple and convenient neural network models to simulate this biological neural network for the estimation of TTC [7,8]. However, one limitation of these earlier models is that the activation in first hidden layer could not be easily explained in physiological terms. In the current paper, we proposed an improved spatio-temporal integrated computational neural network model. To build such a model, we assumed that, as a result of learning, the neural network structure is established using optimal parameters, which have also been observed in physiological findings [9].

The framework of this integrated computational model is summarized in the Flow chart 1. This figure illustrates the entire flow of visual information from the point at which information about the approaching object is initially received, to the final acquisition of TTC information. The retinal image of the object serves as the input for the entire informational process. Subsequently, the primary perceptual system in H1 receives information about the visual angle *è* and as a result, sums the responses from the total of *n* neurons which constitutes the output of H1 and in turn serves as the inputs of H2. Next, based on the changing visual angle over time, the advanced temporal processing system in H2 detects the looming object, represented by the number of responding neurons *n*, and determines whether H2_Output should be produced regarding TTC. Finally the H2_Output TTC is multiplied by the responding neuron number *n* of the looming object to generate the final TTC information output.



Flow chart 1. Visual information process

3 Implementation of the Network

3.1 Implementation of Spatial Neural Network

The first level of the computational model is a spatial neural network (SNN), starting from the input (retinal image of the object) to the H1_Output. Its main role is to perceive visual angle θ and to work out the total responding neurons number n from the initial retinal image. The SNN's structure and information processing mechanisms are proposed based on a simple perceptron neural network and some physiological findings. The physiological and anatomical basis of this structure may be located in the earlier stage of the bird's tectofugal pathway from the retina to the optic tectum.

Fig.1 provides a visual interpretation of this SNN structure (Fig.1a) and information process (Fig.1b).

We assume that the retinal image of the external boundary of the object would trigger the responses from all of the corresponding neurons whose receptive fields overlap with that part of the space. As a result, the spatial separation of the farthest two neurons would provide information about the visual angle, i.e. $\theta=|R_l-R_r|$, and the number of neurons whose receptive fields correspond with the outer edge of the retinal image would be summed together to n. Therefore, if the same object subtends at a different offset angle ($\ddot{a}$ in Fig.1b), then H1_Output would indicate the same visual angle θ but different responding neuron number n. This could occur, for example, if the object is on a direct collision at either $\ddot{a}=0$ (indicated by the right orbit in Fig. 1b), or at $\ddot{a}=5°$(indicated by the left orbit shown in Fig. 1b).

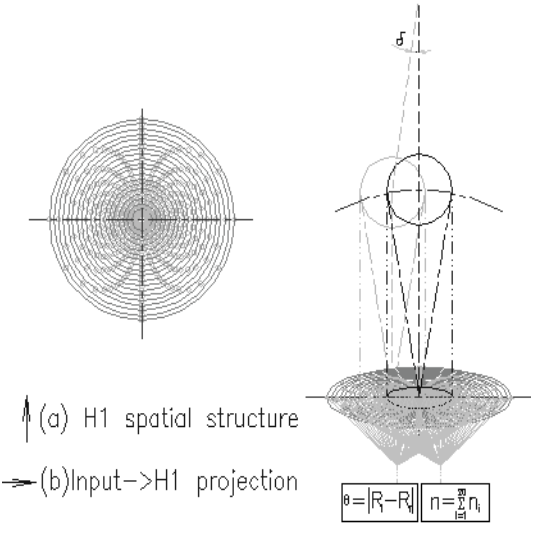


Fig. 1. SNN structure and information process

It is conceivable that the more meaningful variables related to object motion on a collision course would be the object's size and velocity. However, from the retinal point of view, the first order input is the retinal image size and the change of retinal image size and position. Consequently, we make use of the retinal image size as the input of this SNN, which should be evaluated at the retinal level before feeding into the neural network simulating processing in the higher brain areas.

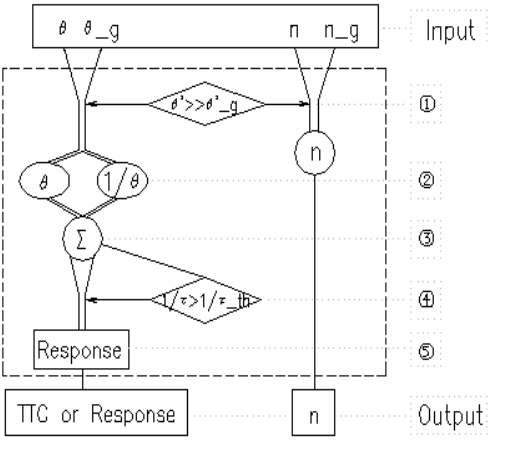


Fig. 2. TNN structure(a) and information process(b)

3.2 Implementation of TNN

The second level of our model is a temporal neural network (TNN) starting from the H1_Output and leading to the final output of the entire network (TTC), shown in Fig.2 and also seen in [7, 8]. The processing at this level could be achieved through biological struc-tures such as the optic tectum, the nucleus rotundus and even the core of the extrostriatum which represents the later stage of the tectofugal pathway in birds.

We employed a back-propagation neural network (BPNN) model. Here, we adopted the traditional "logsig" as a transfer function in step ② in Fig.2, making afferent sequential $\grave{e}_i$ in different input ranges in *logsig* function by means of weights of opposite signs and biases with big differences. This accomplishes the task of simulating $\grave{e}_i$ and $1/\grave{e}_i$.

In order to obtain information about TTC, we use the theoretical value of Tau computed through the equation $1/\tau \approx \theta'/\theta$, to act as the teacher. The final error curve can be derived from the difference between the theoretical value of Tau and the real output TTC of the network. The real output of the model is a serial time sequence, which makes the output unit suitable for a threshold unit. If given a proper threshold, this will allow us to directly predict an impending collision. We also take advantage of the adaptive learning rate and the auxiliary momentum item, by adopting a gradient descend algorithm to improve the algorithm and promote the NN performance.

Table 1. The final optimal parameters of TNN

Layer i to j	W_{ij}	b_{ij}
1 to 2	-1.37,0.9	4.01,0.82
2 to 3	0.17,0.65	-0.5

We assume that BPNN has some optimal parameters to respond to similar inputs quickly. We present a quick searching method in light of minimal square error and best neural network performance for obtaining the optimal parameters. It is conceivable that this optimization reflects millions of years of evolution through competition and adaptation to the surrounding environment. The final parameters are shown in Table 1 where W_{ij} is the connective weight from the *i*th layer to *j*th layer and b_{ij} is the corresponding bias.

4 Results

After training the neural network becomes adaptive to the inputs. Given similar inputs, it could quickly acquire accurate TTC information. Fig. 3 shows that the difference between the theoretical value and the network output in terms of the minimum mean square error is in the e-6 range. The neurons in the middle stages can also have biological implications [7, 8].

In addition, the integrated spatio-temporal neural net-work can also be applied to different collision scenarios, such as identifying the dis-tinction between different forms of motion-in-depth (e.g. self-motion vs. looming[3]), different movement directions (i.e., at different offset angles[6]), and different object sizes and velocities[2]. The physiological implica-tions and their consistency with the experimental animal results have been discussed elsewhere [7, 8].

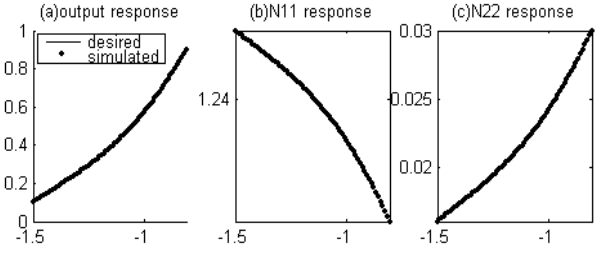


Fig. 3. The final and middle layer results. The MSE of a, b, and c are 1.14e-005, 2.69e-005 and 1.48e-006 respectively.

5 Conclusion and Discussions

Our integrated spatio-temporal neural network is a simple and convenient computational model. Each level can be organized by using an adaptive integration of the NN algorithm and biological data. The properties of the model are consistent with response patterns observed in animals and also match the minimum consumption principle. This neural network not only allows one to obtain accurate TTC information but also has clear physiological significance. The network makes use of neurons from the entire tectofugal pathway, which had been shown to have characteristic responses to looming objects. After all, it is perhaps conceivable that this may recruit the entire visual information stream starting from the initial retinal image to gain high level information about TTC.

The fact that this model performs extremely well indicates that combining physiological findings with an ANN is a very fruitful and powerful approach. Together, the well-designed structure and nearly perfect performance of the model, it not only provides a good approximation of brain mechanisms, but could also provide important insights into machine vision and robotics.

Acknowledgement

This work was supported in partly by the 973 Project No.2003CB716106 and NSFC No. 30525030, 60571019.

References

1. Hecht, H., Savelsbergh, G.J.P. (ed.): Time-to-contact, Advances in Psychology Series, Amsterdam: Elsevier – North Holland (2004)
2. Sun, H.J., Frost, B.J.: Computation of Different Optical Variables of Looming Objects in Pigeon Nucleus Rotundus Neurons, Nature Neurosci. 1 (1998) 296-303
3. Frost, B. J., Sun, H.J.: The Biological Basis of Time to Collision Computation, In: Hecht, H., Savelsbergh, G.J.P. (ed.): Time-to-contact, Advances in Psychology Series, Amsterdam: Elsevier – North Holland (2004) 13-37
4. Gibson, J.J.: The Ecological Approach to Visual Perception, Houghton Mifflin, Boston (1979)
5. Lee, D.N.: A Theory of Visual Control of Braking Based on Information about Time-to-collision, Perception 5 (1976), 437-459
6. Wang, Y.C., Frost, B.J.: Time to Collision Is Signalled by Neurons in the Nucleus Rotundus of Pigeons. Nature 356 (1992), 236-238
7. Wang, L., Yao, D.Z., Sun, H.J.: A Simple Computational Model for the Estimation of Time-to-collision. In: Zhang, Y.T., Xu, L.X., Roux, C., Zhuang, T.G., Tamera, T., Galiana, H.L. (eds.): Proceedings of the 27th IEEE EMBS annual Conference (2005)
8. Yao, D.Z., Wang, L.: Visual Information Processing in Direct Collision Course —A Simple Computational Model. In: He, J.P., Gao, S.K., Lin, J.R. (eds.): Proceedings of International Conference on Neural Interface and Control (2005) 131-134
9. Guo, A.K.: Biological Neural Network, Acta Biophysica Sinica 12 (1991) 615-622

Classification of Movement-Related Potentials for Brain-Computer Interface: A Reinforcement Training Approach

Zongtan Zhou, Yang Liu, and Dewen Hu

Department of Automatic Control,
College of Mechatronics and Automation,
National University of Defense Technology,
Changsha, Hunan, 410073, P.R. China
dwhu@nudt.edu.cn

Abstract. This paper presents a new data driven approach to enhance linear classifier design, where the classifier obtained through theoretical model, is optimized through reinforcement training, to fit the data better as well as to improve its generalization ability. Applied to motor imagery experiment data in EEG based Brain-computer interface (BCI) applications, this method achieved a rather lower mean squared error of 0.59, and by which our group got a second place in the BCI competition III (dataset IVc).

1 Introduction

Brain-computer interfaces (BCIs) detect electrical signals that produced by brain activity, and change these signals into outputs that convey the user's intent to the outside world [1]. The primary goal of BCI research is to enable these users, who may be completely paralyzed, to express their wishes, operate word-processing programs, or even control multidimensional movements of a robotic arm or a neuroprosthesis [1, 2]. It has been shown that the imagination of limb movement can modify brain electrical activity, and different electroencephalograph (EEG) patterns can be obtained under specific motor imagery tasks. Preparation for a real movement or imagination of the movement activate is accompanied by movement related potentials (MRPs) and event-related desynchronization (ERD), i.e. an amplitude decrease of the pericentral μ-and β-rhythms [3], thus can be recognized by means of linear or nonlinear classifiers.

Before classification, proper processing is required to extract the most discriminative information [4]. Common spatial pattern (CSP) is a supervised spatial filter which project the raw multichannel EEG to a few directions that maximizing the energy difference between two categories [5, 6]. It has been proved to be a successful method for extracting MRP features in BCI applications [7, 8]. Wavelet transform (WT) is another popular approach by which subtle time-frequency structure of EEG waves can be analyzed, and more effective compared to simple spectral analysis [9, 10].

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 620–628, 2006.

In our scheme put forward in this paper, these two methods are combined to perform the preprocessing, then feature vectors are constructed by correlation matrix, and an initial linear classifier is given intuitively. Afterwards, weights of this classifier are optimized again on original feature through reinforcement training to get better fitting and generalizing power. Reinforcement learning, as remarked in machine learning theory, is a means to solve the problem that how an agent choose the best action to achieve its goal according to the reward of the environment [11]. In this paper the concept of reinforcement is introduced and utilized to contribute for achieving better classifier design, where the original classification result is treated as rewards, i.e. the samples which are misclassified or bad classified are thus guided to better positions assigned. Due to the fact that the number of samples is equal to the dimension of feature space, overtraining must be taken into account. Leave-one-out testing is employed as the criteria for evaluating the performance of the new classifier.

Description about reinforcement training approach based on correlation matrix is presented in Section 2. Experiment on dataset from BCI competition III and classification results is presented in Section 3, and some discussions will be given in Section 4.

2 Reinforcement Training Based on Correlation Matrix

2.1 Reinforcement Training of Linear Discriminant Analysis

Suppose now we have a set of n samples $\mathbf{y}_1, \ldots, \mathbf{y}_n$, some labelled ω_1 and some labelled ω_2, in a feature space of d dimensions, and we want to use these samples to determine the weights $\mathbf{a}$ in a linear discriminant function $g(\mathbf{a}) = \mathbf{a}^T\mathbf{y}$. The initial weight $\mathbf{a}^0$ is determined by a theoretic model or some *a priori* acknowledge. Due to the disaccord between the actual data and the theoretic hypothesis, the initial decision surface is always not the best one. So we want to change it slightly away from the initial position to fit the data better as well as to get more generalization power.

At the initial position,

$$\begin{pmatrix} y_{10} & y_{11} & \cdots & y_{1d} \\ y_{20} & y_{21} & \cdots & y_{2d} \\ \vdots & \vdots & & \vdots \\ y_{n0} & y_{n1} & \cdots & y_{nd} \end{pmatrix} \begin{pmatrix} a_0^0 \\ a_1^0 \\ \vdots \\ a_d^0 \end{pmatrix} = \begin{pmatrix} b_1^0 \\ b_2^0 \\ \vdots \\ b_n^0 \end{pmatrix} \quad or \quad \mathbf{Y}\mathbf{a}^0 = \mathbf{b}^0 \tag{1}$$

Where $\mathbf{Y}$ is a $n - by - \hat{d}$ ($\hat{d} = d + 1$) augmented feature matrix whose i-th row is the vector $[1,\ \mathbf{y}_i^T]$ and $\mathbf{b}^0$ is the output of the classifier, which represents the approximate posterior probabilities. We can assign better distribution of $\mathbf{b}^0$ by changing $\mathbf{a}^0$ to $\mathbf{a}$ while constraining $\mathbf{a}$ to move within a neighborhood of $\mathbf{a}^0$ so as to restrain over-fitting. That is to solve

$$\begin{aligned} &\min \quad g(\mathbf{a}) = \|\mathbf{Y}\mathbf{a} - \mathbf{b}\|^2 \\ &st. \quad \|\mathbf{a} - \mathbf{a}^0\|^2 < \gamma \end{aligned} \tag{2}$$

Where $\gamma > 0$ is a neighborhood of $\mathbf{a}^0$. Let $\Delta\mathbf{a} = \mathbf{a} - \mathbf{a}^0, \Delta\mathbf{b} = \mathbf{b} - \mathbf{b}^0$. Through introducing a penalty factor β, it transforms to an unconstrained optimization problem

$$\min \quad g(\Delta\mathbf{a}) = \|\mathbf{Y}\Delta\mathbf{a} - \Delta\mathbf{b}\|^2 + \beta\|\Delta\mathbf{a}\|^2 \tag{3}$$

Let $\frac{\partial g}{\partial \Delta\mathbf{a}} = 0$ we have

$$\Delta\mathbf{a} = (\mathbf{Y}^T\mathbf{Y} + \beta\mathbf{I})^{-1}\mathbf{Y}^T\Delta\mathbf{b} \tag{4}$$

Then new weight vector is $\mathbf{a} = \mathbf{a}^0 + \Delta\mathbf{a}$.

1) Assignment of $\Delta\mathbf{b}$. For i-th sample, Δb_i is computed using

$$\Delta b_i = b_i - b_i^0 = k(l_i - b_i^0), \quad i = 1, 2, \ldots, n \tag{5}$$

Where l_i is the target line toward which the i-th sample moves. The range of moving can be controlled by the coefficient k. For convenience, samples belong to the same class have the same l, which is defined as the sample which is furthest away from the origin. Fig. 1 illustrates the distribution of b before and after moving. We can see that the distributions of two classes separate smoothly while retaining their configurations.

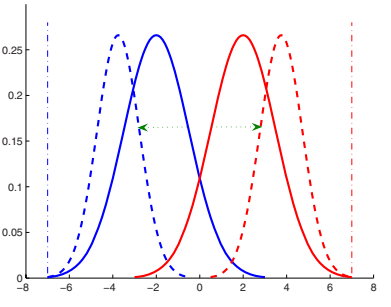


Fig. 1. Assignment of $\Delta\mathbf{b}$

2) Selection of k and β. There is also a danger of overfitting as we adjust the distribution of $\mathbf{b}$ a little subjectively to improve the representing ability of the classifier on training set. To overcome this problem, the penalty factor β and coefficient k can be selected using leave-one-out (LOO) method. That is, for each time, leave out one different sample for test, and perform reinforcement training on the rest. β and k that minimize the total number of errors is then selected.

2.2 Beginning with Correlation Matrix

Here we provide a novel approach for the initial classification. Denote $\mathbf{v_i} = [v_{i1}, v_{i2}, \ldots, v_{iK}]^T \in \mathbf{R}^K$ as the value of i-th sample (it can be written as a vector

whatever the true dimension is). Also we have n training samples $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$, labelled ω_1 or ω_2. Then for an arbitrary sample $\mathbf{v}_j$, n dimension feature $\mathbf{y}_j = [y_{j1}, y_{j2}, \ldots, y_{jn}]^T$ can be constructed, where

$$y_{ji} = \sum_k v_{jk} v_{ik}, \quad i = 1, 2, \ldots, n \tag{6}$$

is correlation between signal v_j and v_i. From another point of view, y_{ji} can be seen as the inner product between $\mathbf{v}_j$ and $\mathbf{v}_i$. If we further constrain that $\|\mathbf{v}_j\| = \|\mathbf{v}_i\| = 1$, it is easy to know that $-1 \leq y_{ji} \leq 1$. Intuitively, the initial weight $\mathbf{a}^0$ of linear discriminant function is given as

$$a_i^0 = \begin{cases} 1/n_1 & if\ \mathbf{v}_i \in \omega_1 \\ -1/n_2 & if\ \mathbf{v}_i \in \omega_2 \end{cases}, \quad i = 1, 2, \ldots, n \tag{7}$$

Where n_1 and n_2 are the numbers of samples of two classes. So far the initial classification which has the same form as (1) has been constructed, and reinforcement training can be performed on it now.

3 Experiments and Results

These approaches are applied in the classification of BCI data for motor imagery experiments. The datasets are provided by Fraunhofer FIRST, Intelligent Data Analysis Group, and Campus Benjamin Franklin of the Charité - University Medicine Berlin, Department of Neurology, Neurophysics Group, for BCI competition III, dataset IVc.

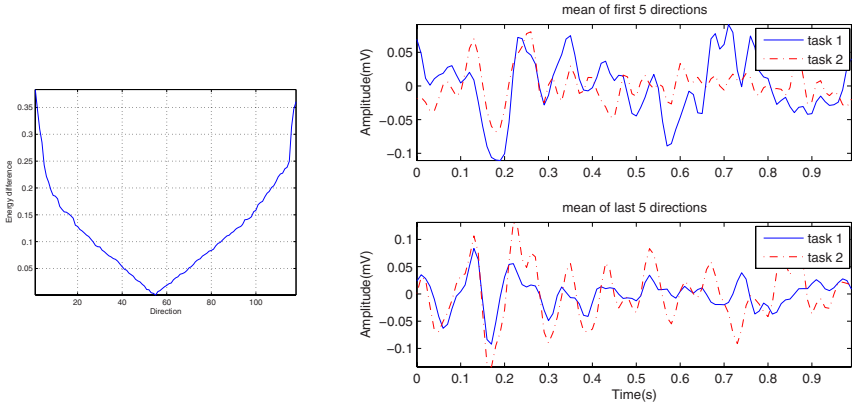
3.1 Experiment Design and Data Acquisition

118 channels of EEG potential were recorded from one healthy subject. Visual cues (letter presentation) indicated for 3.5 seconds which of the following 3 motor imageries the subject should perform: left hand (task 1), right foot (task 2), tongue. The presentations of target cues were intermitted by periods of random length, 1.75 to 2.25 seconds, in which the subject could relax. The test data was recorded more than 3 hours after the training data. The test experiment was similar to the training sessions, but the motor imagery had to be performed for 1 second only, compared to 3.5 seconds in the training sessions. The other difference was that the class tongue was replaced by the class relax (task 3). The data consist of 210 labelled and 420 unlabelled trials for training and test respectively. The performance criteria is the mean squared error (MSE) with respect to the target vector that is -1 for class left, 1 for foot, and 0 for relax, averaged across all trials of the test set.

Original EEG potential data is herein pre-processed by applying band-pass temporal filter followed by CSP spatial filtering to reduce the dimensions, then integrated time-frequency features are constructed through continuous wavelet transformation, to feed to our classifier.

1) Applying Band-Pass Temporal Filter. Due to that the testing data is only 1s length, we get data of 1s length after the position of every cue as trials for both training and testing data. Then we filter signal using a 2.5-25Hz band-pass filter according to channels by the "forward and reverse" method to avoid phase shift.

2) Common Spatial Pattern Filtering. Then we train CSP as a spatial filter to reduce the dimensions. Fig. 2a shows the energy differences between two tasks at every projective direction. Considering the balance between classes, we take the first 5 plus the last 5 directions, thus the 118 channels are reduced to 10 virtual channels. Averaged time courses spatial filtered by CSP of training data are shown in Fig. 2b.



(a) (b)

Fig. 2. CSP processing to reduce data dimensions. a) Absolute energy differences between two tasks at every CSP projective directions. Before 55^{th} direction, energy of task 1 is greater than task 2, and contrariwise after 55^{th} direction. b) Averaged time courses spatial projected by CSP of training data. Top: mean value of first 5 directions, where the energy of task 1 is greater than task 2. Bottom: mean value of last 5 directions, where the energy of task 2 is greater than task 1.

3) Continuous Wavelet Transformation. Continuous Wavelet Transformation (CWT) is used to further transform the signals to time-frequency domain. Signal of every virtual channel after CSP is filtered by the Morlet wavelet which is scaled and temporally shifted, i.e.

$$\Psi_{\tau,s}(t) = \frac{1}{\sqrt{s}}\pi^{-\frac{1}{4}}e^{(i\omega_0\frac{t-\tau}{s})}e^{-\frac{1}{2}(\frac{t-\tau}{s})^2} \tag{8}$$

Next, we get the amplitude of the CWT coefficients, and perform maximum filter on the time-frequency domain to avoid the stripe effect. Fig. 3 shows the correspondent CWT features of the competition dataset.

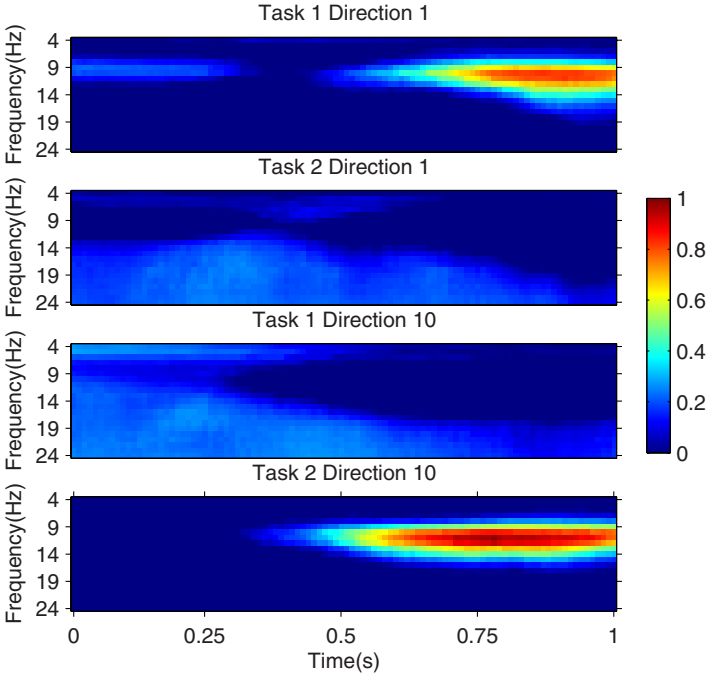


Fig. 3. Averaged CWT coefficients of training signals after CSP (the first and last directions)

3.2 Reinforcement Training from Correlation Matrix

Now the reinforcement training based on correlation matrix is applied on the CWT coefficients of 10 virtual CSP channels. The procedures are: First, correlation matrix for the CWT coefficients is computed on training data using (6) and an initial linear classifier is given as (7); Second, best parameters k and β are selected on the training data; Finally, weights of the classifier is optimized using (4).

The correlation matrix on the training data is shown in Fig. 4, and it can be seen that correlations between features from different category is obviously smaller(referred by the blue-black sub-square). The data is arranged according to category, i.e. the first 105 trails are belong to task 1, and the last 105 trails are belong to task 2. It is clear to see that positive correlations exist among the same task, and negative correlations among different task. Initial features scatter and the feature scatter after reinforcement training of training data are illustrated in Fig. 5, it be seen that the feature separability after reinforcement is getting better.

Then by using LOO method, parameters $k = 5$ and $\beta = 30$ are selected to adjust weights of the classifier.

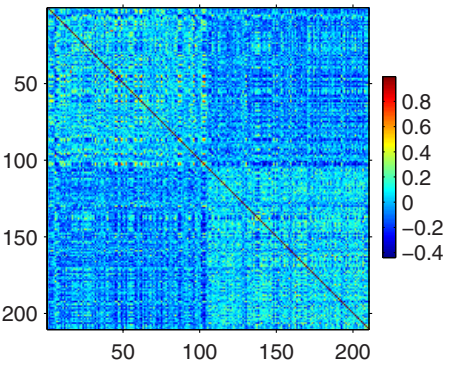
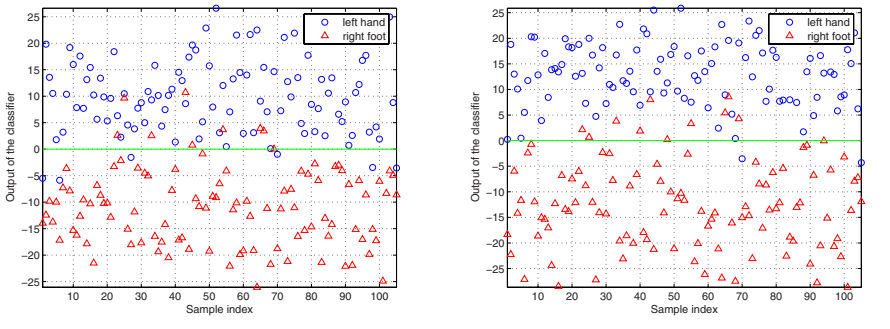


Fig. 4. Correlation matrix in pseudo color on training data with 2 classes, four diagonal squares with clearly different tone shows that there are two distinct classes. If CSP were not applied as the preprocessing step, the square would be much obscure.



(a) Initial feature scatter (b) Scatter after reinforcement training

Fig. 5. Feature scatter before and after reinforcement training, for the training dataset

3.3 Classification

For the CWT coefficients of a testing data, their correlations between training data are computed and feeded into the reinforced optimal classifier to obtain its true label.

Because that in the test data there is another "relax" class (label=0) besides "left hand" (ω_1, label = -1) and "right foot" (ω_2, label = 1) appearing in the training data, we should decide which trials belong to the third class. According to the outputs of linear classifier (denoted by x), every point in the feature space has the posterior probability, $p(\omega_1|x)$ and $p(\omega_2|x) = 1 - p(\omega_1|x)$. When $|p(\omega_1|x) - p(\omega_2|x)|$ is small, x has little confidence to belong to either ω_1 or

ω_2, so it may belong to class "relax". By setting a threshold $T_1 = 0.97$, which corresponding to $|p(\omega_1|x) - p(\omega_2|x)| = 0.97$, make a decision as follows

$$d(x) = \begin{cases} -1, & when\ \ p(\omega_1|x) - p(\omega_2|x)\ \geq T_1 \\ 1, & when\ \ p(\omega_1|x) - p(\omega_2|x)\ \leq -T_1 \\ x/T_1, & when\ |p(\omega_1|x) - p(\omega_2|x)| < T_1 \end{cases} \tag{9}$$

Finally, an MSE of 0.59 is achieved on the dataset IVc of the BCI Competition III, as applied in this paper (MSE of the contest winner is 0.3). In other datasets of the same competition, the reinforcement approach also works well and got us rather good places. (Results of the competiton can be found at http://ida.first.fraunhofer.de/projects/bci/competition_iii/results/index.html.

4 Conclusion and Discussion

Reinforcement training for classifier design is a data driven method aiming at mining the discriminative information as far as possible, and improving both the fitting and generalization ability of an existing classifier. Samples which are misclassified play an important role in this optimization procedure. Control coefficient k is introduced to ensure that the change towards better configuration of distances to decision hyperplane is smooth, and over-fitting is prevented by penalty factor β.

The proposed method for feature construction based on correlation matrix is formally similar to dissimilarity based pattern recognition, where dissimilarity matrix is constructed by the distances between every two objects, and arbitrary classifier can be performed on it [12]. Correlation is not a rigid distance but still can give an evaluation of the similar relation between two samples; furthermore, a simple initial classifier is available for reinforcement training performed in the following step.

However, reinforcement training can not always improve the performance of the classifier as expected, and effect of the misclassified samples in the procedure of reinforcement is still not very clear. These problems will be explored theoretically in our further work. Moreover, a general framework for any classifier would be a promising extension of present approach.

Acknowledgement

This work is supported by National Science Foundation (60575044), Distinguished Young Scholars Fund of China (60225015), Ministry of Education of China (TRAPOYT Project), and Specialized Research Fund for the Doctoral Program of Higher Education of China (20049998012). We would like to thank the organizers of BCI competition for providing the datasets and some other materials used in this paper, which can be found via the competition website http://ida.first.fraunhofer.de/projects/bci/competition_iii/.

References

1. Wolpaw J.R., McFarland, D.J., Pfurtscheller, G., Vaughan, T.M.: Brain-Computer Interfaces for Communication and Control. Clininal Neurophysiology. **113** (2002) 767–791
2. Wolpaw, J.R., McFarland, D.J.: Control of a Two-dimensional Movement Signal by a Noninvasive Brain-Computer Interface in Humans. PNAS. **101** (2004) 17849–17854
3. Dornhege, G., Blankertz, B., Curio, G., Muller, K.R.: Boosting Bit Rates in Noninvasive EEG Single-trial Classifications by Feature Combination and Multi-class Paradigms. IEEE Trans. Biomed. Eng. **51** (2004) 993–1002
4. Liu, Y., Zhou, Z.T., Hu, D.W., Dong, G.H.: T-weighted Approach for Neural Information Processing in P300 based Brain-Computer Interface. Proceedings of 2005 Int. Conf. on Neural Networks and Brain. (2005), to appear
5. Scherg, M., von Cramon, D.: Two Bilateral Sources of the Late AEP as Identied by a Spatio-temporal Dipole Model. Electroenceph. Clin. Neurophysiol. **62** (1985) 32–44
6. Scherg, M., von Cramon, D.: Evoked Dipole Source Potentials of the Human Auditory Cortex. Electroenceph. Clin. Neurophysiol. **65** (1986) 344–360
7. Ramoser, H., Müller-Gerking, J., Pfurtscheller, G.: Optimal Spatial Filtering of Single Trial EEG During Imagined Hand Movement. IEEE Trans. Rehab. Eng. **8** (2000) 441–446
8. Wang, Y., Berg, P., Scherg, M.: Common Spatial Subspace Decomposition Applied to Analysis of Brain Responses Under Multiple Task Conditions: A Simulation Study. Clinical Neurophysiology **110** (1999) 604–614
9. Lemm, S., Schäfer, C., Curio, G.: BCI Competition 2003–Data set III: Probabilistic Modeling of Sensorimotor μ Rhythms for Classification of Imaginary Hand Movements. IEEE Trans. Biomed. Eng. **51** (2004) 1077–1080
10. Bostanov, V.: BCI Competiton 2003 - Data Sets Ib and IIb: Feature Extraction from Event-related Brain Potentials with the Continuous Wavelet Transform and the t-value Scalogram. IEEE Trans. Biomed. Eng. **51** (2004) 1057–1060
11. Mitchell, T.M.: Machine Learning. McGraw-Hill Companies, Inc. (1997)
12. Pekalska, E., Paclik, P., Duin, R.P.W.: A Generalized Kernel Approach to Dissimilarity-based Classification. Journal of Machine Learning Research, Special Issue on Kernel Methods. **2** (2002) 175–211

Two-Class SVM Trees (2-SVMT) for Biomarker Data Analysis

Shaoning Pang, Ilkka Havukkala, and Nikola Kasabov

Knowledge Engineering & Discover Research Institute,
Auckland University of Technology,
Private Bag 92006, Auckland 1020, New Zealand
spang@aut.ac.nz

Abstract. High dimensionality two-class biomarker data (e.g. microarray and proteomics data with few samples but large numbers of variables) is often difficult to classify. Many currently used methods cannot easily deal with unbalanced datasets (when the number of samples in class 1 and class 2 are very different). This problem can be alleviated by the following new method: first, sample data space by recursive partitions, then use two-class support vector machine tree (2-SVMT) for classification. Recursive partitioning divides the feature space into more manageable portions, from which informative features are more easily found by 2-SVMT. Using two-class microarray and proteomics data for cancer diagnostics, we demonstrate that 2-SVMT results in higher classification accuracy and especially more consistent classification of various datasets than standard SVM, KNN or C4.5. The advantage of the method is its super robustness for class unbalanced datasets.

1 Introduction

In biomarker (predictive gene and protein) analysis tasks, based on microarray or proteomics data, a common approach is to find a limited set of genes or proteins from a small number of samples (patients) giving good prediction. Various feature selection procedures have been used to reduce the dimensionality of data before classification. However, feature selection is not guaranteed to give optimum gene sets, because often many equally well performing sets of genes or proteins can be found even from one dataset [2]. If features are selected from a subset of samples (the learning set), this subset can affect strongly the resulting genes or proteins selected for the classifier.

Reduction of features (genes or proteins) is normally necessary, because too many input variables cannot be handled by the algorithms efficiently, and because many redundant and correlated features confuse the classifier. Two main procedures have been used to reduce number of features before classifying: filtering and wrapping [3]. Filtering preselects a subset of genes by e.g. signal-to-noise ratio, correlation coefficient, t-test, F-test, etc. Wrapping selects genes for the classifier by directly evaluating each gene repeatedly for classifying accuracy. All these methods can introduce feature selection bias. Currently no standard

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 629–634, 2006.

widely accepted and used method for feature selection exists. This suggests it would be better to try to avoid this bias completely.

Another important source of bias is due to the unbalanced distribution of samples (patients) into the two classes, e.g. 20 tumour vs. 190 non-tumour samples. Resolving imbalance can be done by resampling the smaller class and/or subsampling the bigger sample [4]. However, reducing the number of samples in the bigger class loses information. Repeated subsampling with a desired ratio of two classes in subsamples can reuse information too much, leading to overfitting.

We avoid the above biases by not using gene selection and using overlapping recursive partitioning of search space by dividing the whole sample space into many subspaces. This is then followed by two-class support vector machine tree, 2-SVMT, a method originally developed for face recognition [6]. Partitioning leads to subspaces containing informative genes that the classifier can find more easily in a smaller subset of data. Overlapping of the partitions is especially useful for small datasets, where non-overlapping partitions might lead to too small or unbalanced numbers of samples in the subspaces.

The 2-SVMT method enables much more efficient computation, because the data is divided into smaller chunks for processing. The biggest advantage arises for classification of small and unbalanced datasets, because overlapping partitions can present more discriminatory information to the classification process from a limited number of samples, and reuse datapoints for the class with fewer samples. In this report we summarize our results on microarray and proteomics two-class data analysed by 2-SVMT and show the benefits of our approach compared to SVM, KNN, and C4.5.

2 2-Class SVM Tree

2.1 Principle

Consider a dataset G with two classes for classification, a principle of "divide and conquer" for constructing SVMT can be implemented as,

$$F(\mathbf{x}) = \begin{cases} 1 \quad if\ f_{g_i}(\mathbf{x}) = 1, \mathbf{x} \in g_i\ i = 1..L \\ -1\ if\ f_{g_i}(\mathbf{x}) = -1, \end{cases} \tag{1}$$

where the total set (G) is divided into subsets $\{g_1, g_2, \cdots, g_L\}$ by a data partitioning function, and the number of subgroups L is determined after the SVM tree model generation.

To minimize the loss function of F, each subset either contains only one class data, or is verified with an optimal classification for SVM. Hence, the whole classification system $F(\mathbf{x})$ consists of a number of local SVM classifiers, each input sample $\mathbf{x}$ is first judged to be the member of a certain subgroup, then the class of $\mathbf{x}$ is determined locally by the subset SVM or the class label of the subset. One SVM is taken as one node of the tree, and a set of SVMs encapsulated in a hierarchy into an SVM tree for 2-class data classification.

2.2 Algorithms

The structure of 2-class SVM tree is constructed by a recursive procedure of data partitioning and a 2-class SVM classification. For modelling each node of 2-class SVM tree we have two iterated steps. First, the data is split into two subsets by the selected partition method. Next, any partition that contains only one class becomes a leaf node of the tree. For any other partition containing two-class data, a 2-class SVM classifier is trained for decision making over this partition. If the partition is not linear separable by SVM, the partitioning is repeated and the procedure is iterated. Partitioning is stopped when all partitions have been resolved.

To classify a new input x, first we decide in which partition the test input data belongs to by executing the partition method $P(x)$ at the root node in the SVM tree. Depending on the decision made by the root node, we will go down to one of the children nodes. This procedure is repeated until a leaf node or a SVM node is reached. We assign the class label to the testing sample x depending on the label of the reached leaf node, or the output of the SVM node. Therefore, the final decision of classifying new data can be cooperatively made by a hierarchical set of 2-class SVMs, with a suitable kernel chosen to suit the dataset.

2.3 Partitioning Method

Due to the loss function of F in Equation (1) being proportional to size of the subset (i.e. number of samples in the subset), it is desirable to seek a bigger size subset optimal for classification in the above 2-Class SVMT algorithm, which means a scalable data partitioning function would be beneficial. For such a scalable method, we chose the Evolving Clustering Method (ECM) which a fast one-pass algorithm for dynamic clustering of an input stream of data. This algorithm is a distance-based clustering method where the cluster centers (called "prototypes") are determined online such that the maximum distance, *MaxDist*, between an input $\boldsymbol{x}_i$ and the closest prototype cannot be larger than a threshold value, *Dthr*. The details of ECM algorithm can be found in [5]

ECM partitioning helps classification of gene expression data because ECM enables splitting the data in a multi-scaled way through parameter *Dthr* adjustment. Particularly, ECM allows overlapping partitions, which overcomes the common problem of few samples (patients) for biomarker data.

3 Application to Biomarker Data

3.1 Datasets and Processing

For validation of 2-SVMT approach with recursive partitioning for biomarker data analysis, we used one proteomics and seven microarray datasets with a two-class classification problem in cancer bioinformatics. The datasets showed a range of bias ratio, from 1.0 (equal number of samples in classes 1 and 2) to 0.33. Target classes were used as reported in the original references listed in Table 1.

Table 1. Comparison of classification accuracies of KNN, C4.5, SVM and 2-SVMT for eight biomarker (gene and protein) datasets, based on 10-fold cross validation, values are means of 10 runs. Numbers in boldface indicate the best results.

Cancer Dataset/Ref	Genes	Bias ratio class 1/2 patients	KNN	C4.5	SVM	2-SVMT
Lymphoma(1)/1	7129/	19/58=0.33	75.3%	**92.2%**	84.4%	77.9%
Leukemia*/2	7219/	11/27=0.41	**82.4%**	58.8%	64.7%	78.3%
CNS Tumour/3	7129/	21/39=0.53	58.3%	63.3%	50.0%	**72.0%**
Colon Cancer/4	2000/	22/40=0.55	79.0%	75.8%	71.3%	**80.7%**
Ovarian Cancer/5	15154/	91/162=0.56	91.7%	†	**97.3%**	75.0%
Breast Cancer */6	24482/	34/44=0.77	63.2%	†	52.6%	**73.7%**
Lymphoma(2)/1	6431/	26/32=0.81	53.5%	58.6%	51.7%	**60.3%**
Lung Cancer */7	12533/	16/16=1.0	**87.9%**	†	64.4%	75.0%

* Independent validation dataset was used for the accuracy evaluation. †not determined due to memory shortage of our computing system/

Data were divided into overlapping partitions using ECM as described above. Then a hierarchical 2-SVMT (second order polynomical kernel) was built using a classifier for each partition, validated by 10-fold cross-validation with random sampling. The average classification accuracies were calculated and the 2-SVMT results compared to other standard classifiers without partitioning (standard KNN (K=1, Euclidean Distance), C4.5, SVM (the second order polynomial kernel). Algorithms were implemented in Matlab Version 6.5, run on Pentium 4 PC, 3.0GHZ 512Mb RAM.

3.2 Results and Discussions

Cross-validated classification accuracies for data without gene selection in Table 1 show that 2-SVMT method with overlapping recursive partitioning performed best in the majority of cases (numbers in bold). Calculations of the average classification accuracy are in Table 2. The 2-SVMT was clearly the best for average accuracy and notably also most consistent, having a much narrower range of accuracies (60.3% to 80.7%) than other methods (50.0% to 97.3%) over the eight datasets.

Interestingly, the classification accuracy of SVMT seemed to increase with bias in distribution of patients to the two classes. This suggests unbalanced datasets

Table 2. Classification accuracy (%) of the three methods, based on the 8 datasets in Table 1. Boldface indicates the best result.

Model	mean (%)	Max - Min (%)	Range
KNN	73.0	91.7-53.5	38.2
SVM	66.6	**97.3**-50.0	47.3
C4.5	69.7	58.6-92.2	33.6
2-SVMT	**76.0**	80.7-60.3	**20.4**

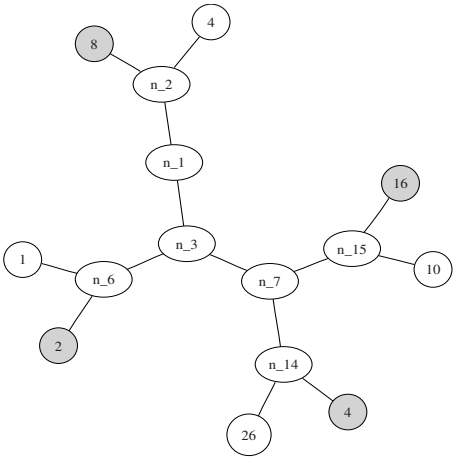


Fig. 1. An example of two-class support vector machine decision tree (2-SVMT) for CNS tumor. Ellipses: Partitioning and SVM decision nodes, numbering indicates the sequence of decisions. Open circles: samples (patients) assigned to class 1, filled circles: class 2, corresponding to Table 1. Numbers inside circles indicates the number of patients in the end of each decision path.

may be particularly suitable for analysis by 2-SVMT. This is in contrast to the KNN, C4.5, or SVM results, in which class bias did not correlate with accuracy.

An example of a decision tree for CNS tumor (Figure 1) illustrates the working of the method. Most of class 1 patients were classified into one node of 26 patients. Class 2 patients were classified into two main nodes with 16 and 8 individuals, suggesting a potential difference in these two subsets of patients. All the rest of the individuals were classified into nodes of only a few patients each, so the question arises, whether these are especially hard to classify patients, misclassifications or maybe represent some other types of cancer. This exemplifies the potentially valuable additional information that the 2-SVMT can provide, compared to other algorithms.

Other cancer datasets produced similar trees, some simpler, some more complex. It appears some cancers are inherently more difficult to classify than others, and this is reflected in the complexity of the classification trees.

4 Conclusions

The 2-SVMT method is a new useful classification tree method to analyse biomarker data. The recursive partitioning divides the problem to smaller subproblems solved in succession by a sequence of SVM nodes. Overlap of the partitions is useful in smaller and very unbalanced datasets, enabling the classifiers to utilize some of the data multiple times in more balanced subsets. Analogously, Ho [7] used random (but non-overlapping) subspace partitions to construct decision tree forests by combining multiple trees. Similarly, one can reuse information by

creating ensembles of decision trees by bagging, boosting or randomization. Our 2-SVMT results with biomarker data thus confirm further the usefulness and utility of data partitioning to improve classifier performance.

The 2-SVMT method often outperformed KNN, C4.5 and SVM, producing both more accurate and, importantly, more consistent classification, safeguarding against occasional poor performance of other algorithms. Part of the reason for success may be in allowing overlaps in subspace partitions, though this would need to be verified by using 2-SVMT with various schemes of partitioning.

The advantages of the 2-SVMT are such that it would be worth to extend the method to multiclass classification case (work in progress). This might be most beneficial for small unbalanced datasets, where overlapping partitions might help to reuse the data in a way facilitating the correct classification of easily distinguishable subsets of data. This would help in many cases of biomarker datasets, which often focus on small numbers of patients in multigenic diseases having a variety of phenotypes to be classified simultaneously.

References

1. Alon, U., Barkai, N., Notterman, D.A., Gish, K., Ybarra, S., Mack, D., Levine, A.J.: Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumour and Normal Colon Tissues Probed by Oligonucleotide Arrays. Proc Natl Acad Sci U. S. A. 96(12) (1999) 6745-6750
2. Ein-Dor, L., Kela, I., Getz, G., Givol, D., Domany, E.: Outcome Signature Genes in Breast Cancer: Is There a Unique Set. Bioinformatics 21(2) (2005) 171-178
3. Deb, K., Agrawal, S., Pratap, A., and Meyarvian, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2) (2002) 182-197
4. Chawla, N.V., Bowyer, K.W, Hall, L.O. and Kegelmeyer, W.P.: SMOTE, Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research 16 (2002) 321-357
5. Kasabov, N.K.: Evolving Connectionist Systems, Methods and Applications in Bioinformatics, Brain Study and Intelligent Machines. Springer-Verlag, (2002)
6. Pang, S., Kim D., and Bang, S.Y.: Face Membership Authentication Using SVM Classification Tree Generated by Membership-based LLE Data Partition. IEEE Trans. on Neural Networks 16(2)(2005) 436-446
7. Ho, T.K.: The Random Subspace Method for Constructing Decision Forests. IEEE Transaction on Pattern Analysis and Machine Intelligence 20(8) (1998) 832-844

Interpreting Gene Profiles from Biomedical Literature Mining with Self Organizing Maps

Shi Yu, Steven Van Vooren, Bert Coessens, and Bart De Moor

Department of Electrical Engineering,
University of Leuven, Belgium B3000

Abstract. We present an approach to interpret gene profiles derived from biomedical literature using Self Organizing Maps (SOMs). Comparison of different clustering algorithms shows that SOMs perform better in grouping high dimensional gene profiles when a lot of noise is present in the data. Qualitative analysis of the clustering results prove that SOMs allow an in-depth interpretation of gene profiles with biological relevance.

1 Introduction

In this paper, we present a self organizing maps (SOMs) based application by mining gene profiles from biomedical literature and clustering those profiles with reference to biomedical topics. SOMs has a number of features that make them particularly well in clustering and analysis of gene expression patterns. They are ideally suited to exploratory data analysis, allowing one to impose partial structure on the clusters and facilitating easy visualization and interpretation[5]. However, the profiles of genes we use in our approach are different from sequence expressions. They are extracted purely from textual resources such as the MEDLINE database. Hence the application we discuss in this paper represents a text mining approach of bioinformatics study. In this paper, we introduce this tool in clustering textual mining profiles and compare it with other clustering methods which are also widely used in pattern analysis.

2 Gene Profiling by Mining Biomedical Literature

Our approach is divided into two stages. The first stage is the extraction of gene profiles by literature mining. A corpus collection of biomedical literature titles and abstracts is investigated by the TXTGate platform [2] to create textual profiles of genes with different scopes. More details about this mining system are discussed in previous work [2]. The gene profiles we use in this paper are produced by controlled vocabulary indexing with Gene Ontology. The mining process is conducted by a list of 32,814 human genes and a pruned vocabulary of 9,175 Gene Ontology terms (GO terms). These GO terms are pruned from the standard GO terms by segmentation and indexing: The segmentation process splits the long GO terms, for example, “regulation of DNA recombination” into smaller terms in addition to the original term: “regulation”, “of”, “DNA” and

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 635–641, 2006.

"regulation DNA recombination". After that, the literature corpus is indexed with all segmented GO terms. Only those terms that occurred in the corpus at least once, are kept. Thus it leads to a 32,814 $\times 9,175$ matrix of gene-by-GO terms. By referencing Gene Ontology, each gene is associated to a certain number (usually $300 \sim 400$) of segmented GO terms. The textual score of a gene is hence represented as a vector, where each elements is represented as the inverse document frequency (IDF) scores of corresponding GO terms in the corpus. The GO term's IDF score is static in a given corpus, however, due to the variety of numbers of associated terms for genes, the resulted gene profiles are different after normalization.

Although we have used controlled vocabulary to index the literature, the profiles we get is still large in feature space and data amount. Sometimes, due to the large scale of dataset, distance measure of profiles in vector models might not be sensitive and accurate enough to unfold some important functional relationships. Topic based research, in this sense, should be a more appropriate solution to interpret the mining results. So next stage we will try to compare some clustering methods in grouping these profiles and check in which degree these groupings are consistent with our prior understandings.

3 Gene Clustering by Self Organizing Maps

3.1 Test on a Small Sample of Candidate Gene

Firstly we try a small test. We select 36 genes from 5 biomedical topics about genetic and immunological diseases. The selection of topic related genes is guided by Entrez genome database. The result dataset of different topics and their related genes is shown in Table 1. The gene profiles are trained in SOMs to generate 5 clusters. The clustering result are shown in Figure 1, but illustrated manually to have a clear view of the clusters. The qualitative analysis of the clustering result shows that the gene profiles are geometrically grouped into several areas:

Table 1. Manual clusters of 36 candidate genes from 5 biomedical topics

Breast Cancer	*Alzheimers*	*Parkinson*	*Leukaemia*	*Wolf Hirschhorn Syndrome*
BRCA1	APOE	CYP2D6	BCL11A	WHSC1
BRCA2	PSEN1	MAPT	BCL11B	WHSC2
BRCA3	PSEN2	PARK2	MLL	WHSC1L1
SNCG	APP	PARK7	TNFRSF8	MSX1
NCOA3	SNCA		BLNK	
TFF1	ACE		NR0B2	
AMPH			NOTCH1	
BCAS2			CD86	
ABCG2			P2RX7	
BRMS1			SEL1L	
SDBCAG84			CDK9	

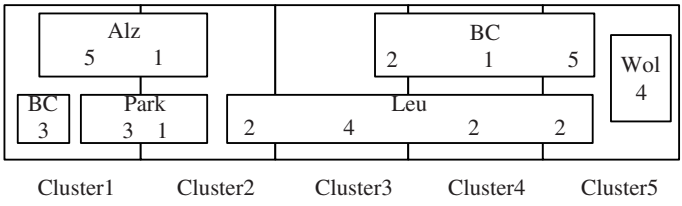


Fig. 1. The SOMs clustering result of genes in Table 1. The number marked in each cluster square represents the number of genes from the topic is grouped in this cluster.

Alz(*Alzheimers*) genes are gathered at the left side and Wol(*Wolf-Hirschhorn syndrome*) genes are gathered at right side. It seems that the clustering result by SOMs is quite informative and might be useful to group genes for topic based studies.

3.2 Clustering Result and Comparison with Other Methods

There is a problem need to be mentioned before we use SOMs to interpret large scale data. Usually we are only interested in part of the genes, for instance, 100 out of 100,000 genes, and it is reasonable to just analyze this sample part. However, in gene clustering most algorithms are sensitive to instances: the result of clustering 100 genes which we are interested might be much different from the result the total dataset clustering. In this section we use a cluster evaluation method, BCUBED[1], to check the effect of SOMs clustering in comparison with other algorithms in grouping gene topics. BCUBED was originally used for evaluating coreference resolution. It uses the complementary of distance measure to calculate the similarity between two clusters, where an "accurate" cluster is manually selected and a "evaluating" cluster is automatically generated by algorithms. The complement of the distance from the automatic cluster away from the manual one is considered as the "fitness" of clustering.

We manually select 416 candidate genes from 40 biomedical diseases with the reference of Entrez. The textual profiles of these genes are retrieved from the text mining result introduced in section 2. We compare 4 clustering methods, HCL complete linkage, HCL average linkage, K-Means and SOMs. The experiment comprises 5 repetitions. For each repetition, the grouping of 5 manually topics and their membership genes represents the "manual cluster". For example, 30 genes are selected and each topic contains 6 genes. The algorithmic clustering result of these 30 genes is regarded as "automatic cluster of pure profiles". The noisy profiles is produced by mixing the previous selected genes with the same number of other "noisy" genes, which lead to 60 in total. We use the same clustering methods to deal with 60 gene profiles as same as we deal with the pure profiles, which also produces different groupings. Then we remove those "noisy" genes from grouping results and regard them as "automatic cluster of noisy profiles". Thus we get 3 different clustering results in each repetition and 3 BCUBED scores are measured. The geometrical meaning of 3 BCUBED scores

Table 2. BCUBED evaluation and ANOVA test of clustering results. In the first sub table, the BCUBED values are expressed in the form of mean(variance).

		SOMs	*HCL Average*	*HCL Complete*	*K-Means*
M1	Precision	0.4165 (0.0044)	0.3932 (0.0093)	0.4275 (0.0091)	0.3762 (0.0040)
	Recall	0.4365 (0.0027)	0.6447 (0.0121)	0.5224 (0.0054)	0.3713 (0.0043)
M2	Precision	0.4072 (0.0069)	0.3607 (0.0081)	0.4278 (0.0064)	0.3927 (0.0063)
	Recall	0.4179 (0.0100)	0.7021 (0.0144)	0.5408 (0.0102)	0.3997 (0.0063)
M3	Precision	0.7284 (0.0059)	0.7027 (0.0181)	0.6588 (0.0058)	0.4135 (0.0084)
	Recall	0.7056 (0.0174)	0.7942 (0.0197)	0.6765 (0.0083)	0.4277 (0.0058)

		F Value(All)	*Pr > F(All)*	*F Value(KM-SOM)*	*Pr > F(KM-SOM)*
M1	Precision	0.34	0.7953	0.78	0.4036
	Recall	9.11	0.0009	2.42	0.1582
M2	Precision	0.49	0.6974	0.06	0.8082
	Recall	9.30	0.0009	0.77	0.4049
M3	Precision	8.75	0.0011	27.67	0.0008
	Recall	7.70	0.0021	13.32	0.0065

on cluster evaluation is illustrated in Figure 2, where the d_1 corresponds to the complement of similarity of pure profile clusters between manual cluster. The values of this measurement are list in first row(M1) in Table 2. The d_2 in Figure 2, corresponds to the complement of values in the second row(M2) of Table, denotes the distance of clustering results of noisy gene profiles between manual cluster. The d_3 in Figure 2 can be measured by comparing two automatic clusters, which semantically means the preservation ability of clustering methods in noisy data. It correspond to the complement of BCUBED values in the third row(M3) of Table 2. Notice that the larger values in Table 2 mean the smaller distance in Figure 2 because they are complements. So in this experiment we measure 60 BCUBED scores which are categorized in 5 repetitions, 4 clustering methods and 3 measurements of clusters.

According to our result, SOMs and KMeans show a similar ability in clustering genes under M1 and M2. The HCL methods are obviously faulty because they produce too many singleton clusters, which also explains why they have exceptional high values of recall: BCUBED score doesn't penalize enough to the occurrence of singleton clusters. However, both the mean value and the ANOVA test show that SOMs is more robust than KMeans in preserving gene clustering (F-value 27.67 of Precision and 13.32 of Recall in row M3 of Table 2). We can thereby conclude that topic groupings produced by SOMs might be insensitive to the irrelevant genes. This is a interesting feature for topic based analysis of gene profiles. Correspondingly, if we draw the positions of 3 clusters of KMeans and SOMs in Figure 2, the distances of d_1 and d_2 are similar between methods while SOMs has much smaller d_3 then KMeans.

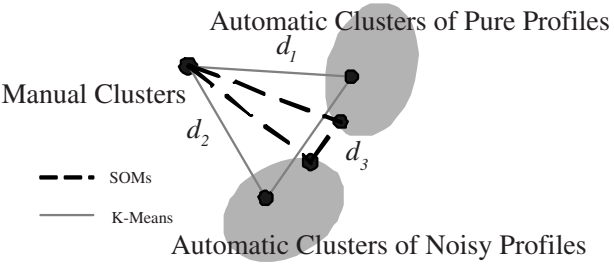


Fig. 2. The geometrical illustration of evaluation in Table 2. It represents the meaning of 3 different measures M1-M3 in Table 2. The two triangles show the difference between KMeans and SOMs clustering according to ANOVA test in Table 2.

4 Topic Map Construction of Gene Profiles

In this section we use the same dataset of 416 gene profiles in the previous section and cluster them in a big SOMs containing 50×80 neurons. The map is trained and visualized by ESOM[6]. Due to the capacity and load of training software, the 9,175 feature space of data is reduced by latent semantic methods to 300 terms. The training process iterates 5,000 epochs and costs 60 hours on 64-bit Unix Server with 8G memory. Notice here if the map is for data prediction, the dataset must be kept in full dimension and training will take much longer time. In Figure 3, it shows that the distribution of neurons and matched data in a well trained map is also well stretched. The dark area represents the activated neuron, deeper color represents the high density of data matches. Inversely, the light area represents the inactivated (gap) neurons between activated neurons and lighter colors stand for bigger gaps. Since the numbers of genes selected from different biomedical topics are varied, the map is biased towards larger topics with more genes selected such as Cro($Crohn's Disease$), Alb($Albinism$), etc.

We divide our analysis of the map into 2 levels. At the 1st level, we check the matched data at each neuron and mark its labeled biomedical topic on the map. An intuitive look of this map shows that many genes form local topic groups. Some example groups are marked with white polygons in the map. A re-labeling of these topic groups will produce the 2nd level interpreted map which allows a more intuitive interpretation of gene profiles. We also find that most topic groups are distributed in the edge of map rather than in the center. The center of map seems as a "meeting point" that all kind of genes are mixed together. This coincides to the clustering result in small test in Section 3 and Figure 1. Moreover, we construct a very simple ontology for the 40 selected topics and reference it in our map. If shows that some genes from relevant biomedical topics stay together. For example, Ost(*Osteopetrosis*), Der(*Dermatomyositis*) and Mus(*Muscular Dystrophies*) all belong to "Musculoskeletal Diseases". Correspondingly, they stay close to each other in the centered black line curve in Figure 3. Similarly, genes of "Skin and Tissue disease", for instance, Ehl(*Ehlers-Danlos Syndrome*), Sys(*Systemic Lupus Erythematosus*) and Scl(*Scleroderma*) stay close in the left

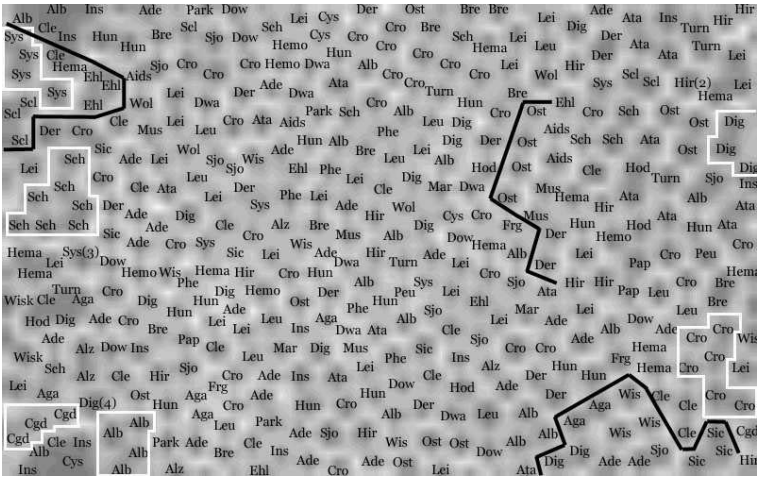


Fig. 3. The 1st level interpretation of a 50X80 neurons map trained with 416 genes selected from 40 biomedical topics

upper black line curve of the map. Some genes of "immunological disease and blood disease", for example, Aga(*Agammaglobulinemia*), Wis(*Wiskott-Aldrich Syndrome*), Dig(*DiGeorge Syndrome*), Sjo(*Sjogren's Syndrome*) and Sic(*Sickle cell anemia*)stay together in the right lower black line curve.

A possible problem here is the inaccuracy of gene selection. It is possible that some weakly related genes play much more important role in another topic but we manually classify them into an inappropriate class. Another problem is that there are quite a lot of genes strongly related to multiple topics, for example, gene

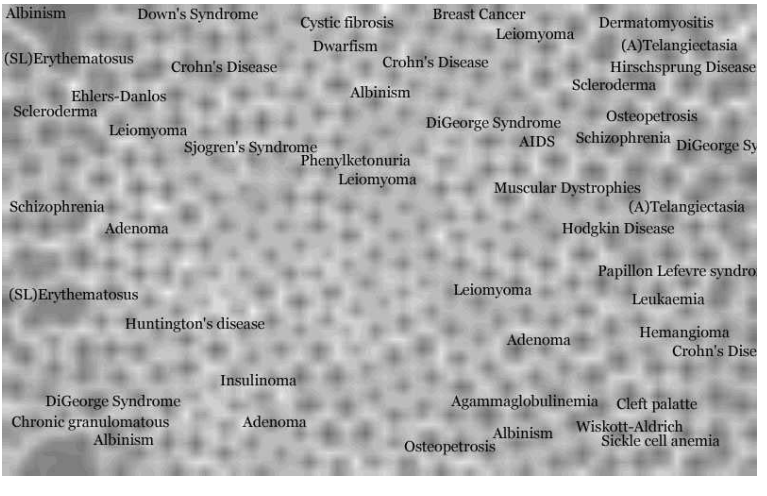


Fig. 4. The 2nd level interpretation: topics map of gene profiles

TP53 is related to Ata(*Ataxia Telangiectasia*) as well as Ade(*Adenoma*). And it is also important to care about the topic selection, for example, Ade(*Adenoma*) actually is a higher level topic than Ins(*Insulinoma*) and Lei(*Leiomyoma*), thus many genes which belong to the sub topics are also related to the higher topics. It is necessary to make sure that the selected topics are representative both in medical categorization and importance. It is also important that the number of their selected genes for SOMs interpretation is almost same to avoid bias in final map interpretation.

5 Conclusion

In this paper we used SOMs to interpret gene profiles mined from biomedical literature. It seems that SOMs performs better than other algorithms in grouping gene profiles when a lot of noise is presented in the data. This result will be meaningful to our later interpretation task of text mining because it allows us to analyze the results by small samples or incremental study without losing the true knowledge inherent in the result. The training result of using large SOMs for textual gene profile clustering also provides us in-depth knowledge of relationships between genes. Based on these interpretations, we might able to build hypothesis about the biomedical relevance of genes and find more interesting biological evidences in case studies.

References

1. Baldwin, B., Morton, T., Bagga, A., Baldridge, J., Chandraseker, R., Dimitriadis, A., Snyder, K., Wolska, M.: Description of the Upenn Cam System as Used for Coreference. In: Proceedings of the Seventh Message Understanding Conference (MUC-7). Kaufmann, San Mateo CA(1998)
2. Glenisson, P., Coessens, B., Van Vooren, S., Mathys, J., Moreau, Y., De Moor, B.: TXTGate: Profiling Gene Groups with Text-based Information. Genome Biology 5(6) (2004) 1-12
3. Van Hulle, M.: Faithful Representations and Topographic Maps: From Distortion to Information Based Self Organization. John Wiley & Sons Press (2002)
4. Kohonen, T., Kaski, S., Lagus, K., Salojärvi, J., Honkela, J., Paatero, V., Saarela, A.: Self Organization of a Massive Document Collection. IEEE Trans. Neural Networks 11(3) (2000) 574-585
5. Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E., Golub, T.: Interpreting Patterns of Gene Expression with Self-organizing Maps: Methods and Application to Hematropoietic Differentiation. Genetics 96(6) (1999) 2907-2912
6. Ultsch, A., Moerchen, F.: ESOM-Maps: Tools for Clustering, Visualization, and Classification with Emergent SOM. Technical Report No. 46. Dept. of Mathematics and Computer Science, University of Marburg, Germany(2005)

Mining Protein Interaction from Biomedical Literature with Relation Kernel Method

Jae-Hong Eom and Byoung Tak Zhang

Biointelligence Laboratory,
School of Computer Science and Engineering,
Seoul National University Seoul 151-744, South Korea
{jheom, btzhang}@bi.snu.ac.kr

Abstract. Many interaction data still exist only in the biomedical literature and they require much effort to construct well-structured data. Discovering useful knowledge from large collections of papers is becoming more important for efficient biological and biomedical researches as genomic research advances. In this paper, we present a relation kernel-based interaction extraction method to extract knowledge efficiently. We extract protein interactions of from text documents with relation kernel and *Yeast* was used as an example target organism. Kernel for relation extraction is constructed with predefined interaction corpus and set of interaction patterns. The proposed method only exploits shallow parsed documents. Experimental results show that the proposed kernel method achieves a recall rate of 79.0% and precision rate of 80.8% for protein interaction extraction from biomedical document without full parsing efforts.

1 Introduction

It is known that protein interactions are primary biochemical reactions in the organisms and they take part in important functions since they determine the biological processes [1]. Therefore, broad description and in depth analysis of these interactions would significantly help the understanding of many important biological phenomena. For *S.cerevisiae* (budding yeast), many researchers have undertaken the task of functionally analyzing the yeast genome comprising more than 6,300 proteins (YPD) [2] and abundant interaction data have been produced by many research groups after the completion of the genome sequences.

Many machine learning-based promising methods have been successfully applied to the analysis of these data. One of these methods is knowledge discovery from text data called '*text mining*.' Even though there are many databases for protein interactions, many interaction data still exist only in the biomedical literatures. They are spread in biomedical literature written in the form of natural languages and they require much effort such as data mining for constructing well-structured data forms. Thus, the studies on the methodology of how to extract protein interactions from biomedical literature have been an active research subject. Many approaches widely adopted natural language processing (NLP) techniques for this problem. Many of these methods can be regarded as the parsing based methods and both full and shallow parsing strategies have been attempted.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 642 – 647, 2006.

A general full parser approach with domain specific grammars was proposed by Yakushiji *et al*. [3]. They used general full parser with grammars for biomedical domain to extract interaction events by filling sentences into augmented structures. Park *et al*. [4] proposed bidirectional incremental full parsing with combinatory categorical grammars. Temkin *et al*. [5] also utilized a lexical analyzer and context-free grammar to extract gene, protein, and interactions of small molecules. They achieved a recall rate of 63.9% and precision rate of 70.2%. As a similar method, template generating method via preposition-based parsing proposed by Leroy *et al*. [6]. They achieved precision of 70% for biomedical literature abstract processing. For a partial parsing method, Pustejovsky *et al*. [7] used the relational parsing for the inhibition relation with recall rate of 57%. But, these methods are intrinsically complicated and the performance was not satisfactory.

In this paper, we present a relation kernel-based interaction extraction method which extract protein interactions of *S.cerevisiae* from biomedical literature. We define kernel for protein relation extraction, called '*relation kernel*', with predefined interaction corpus and set of interaction patterns. Proposed kernel method for interaction extraction only exploits shallow parsing results.

This paper is organized as follows. In Section 2, we describe the basic concept of kernel method. In Section 3, relation kernel for protein interaction extraction is explained. In Section 4, we show experimental results of interaction extraction. Finally, Section 5 we present concluding remarks and future directions.

2 Concept of Kernel Method

Kernel methods are an alternative to feature-based methods. Kernel methods preserve the original representation of objects and use the object in algorithms only via computing a kernel function between a pair of objects. Kernel function K over the object space X is binary function K: $X \times X \rightarrow [0, \infty]$ mapping a pair of objects $x, y \in X$ to their similarity score $K(x, y)$. This is embedding procedure of data items into a vector space F, called a *feature space*, and searching for linear relation in this space. This embedding is defined implicitly by specifying an inner product for the feature space via a symmetric and positive semidefinite *kernel function*: $K(x, y) = \langle \Phi(x), \Phi(y) \rangle$, where $\Phi(x)$ and $\Phi(y)$ are the embeddings of data items x and y. Additionally, any kernel function implicitly calculates the dot product of feature vectors of objects in high-dimensional feature spaces and a function defined as a dot product of the corresponding feature vectors is necessarily a kernel function [8].

In many cases, it is possible to compute the dot product of certain features without enumerating all the features. An excellent example is that of subsequence kernels [9]. In the subsequence kernels, the objects are strings of characters, and the kernel function computes the number of common subsequences of characters in two strings, where each subsequence match is additionally decreased by the factor reflecting how spread out the matched subsequence in the original sequences. Despite of exponential number of features (subsequences), it is possible to compute the subsequence kernel in polynomial time. Therefore, one can exploits long-range features in string without enumerating the features explicitly.

There are a number of learning algorithms that can operate only using the dot product of examples. The models produced by the learning algorithms are also expressed using only dot product of example. Substituting a particular kernel functions in place of dot product defines a specific instantiation of such learning algorithms

The Support Vector Machine (SVM) is known as a learning algorithm that not only allows for a dual formulation, but also provides a rigorous rationale for resisting overfitting. For a kernel-based algorithms working in extremely rich feature spaces, it is crucial to deal with the problems of overfitting. Many experimental results indicate that SVM is able to generalize the classification boundary very well and avoid overfitting in high dimensional feature spaces effectively. So, here we use SVM and relation kernel (extended version of subsequence kernel for shallow parsing) to extract protein interactions.

3 Relation Kernel for Protein Interaction Extraction

Here we use the basic notion and the idea of subsequence kernel and extend it to operate on shallow paring for protein relation extraction according to the work of Zelenko *et al*. [8] and Lodhi *et al*. [9].

For the example sentence, "In those pathways, usually the yeast protein RAP1 suppresses the up-regulation of SH3", we can generate appropriate parse tree(s) for this sentence and compare with the other parse trees generated from other sentences by conventional shallow parsing approach. The form of target relation that we want to extract in this example sentence is 'RAP – suppress – SH3' which generally has the form of 'entity 1 – action verb – entity 2.'

In this paper, we use shallow parsing only for identifying its key elements such as entities and interactions instead of providing full interpretation of the sentence because this approach may provides us a fairly robust shallow parsing and the generation ability of structured representations even for ungrammatical sentences.

Next, we convert the shallow parse tree into examples for the '*entity–interaction–entity*' relation. This relation describes protein interactions such as gene–gene, gene–protein, and protein–protein. The type of the former and the later entity includes those names of gene and protein, and the type 'relation' holds between a two 'entity' and an 'interaction.' For each example sentence, we check whether the sentence has complete relation structure which has two entities and one interaction verb structures.

Each node of example (parse tree of sentence) has its 'type' and 'role.' We define *Type* = {φ, GeNP, GeN, PrNP, PrN, VerbPos, VerbNeg, VerbUnk, PNP, NP, Prep, Adj, Det} and *Role* = {φ, Entity_1, ActionVerb, Entity_2, UNK}. The GeNP and GeN of 'type' mean a noun phrase or a noun which represents a name of gene or protein. Verbs are classified into three categories; positive (e.g., activate, up-regulate, etc.), negative (e.g., inhibit, prohibit, etc.), and other unknown action verbs (VerbUnk). Generally, 'Type' represents the category of par-of-speech (POS) tagged results. On the other hand, 'Role' represents the role of 'type' in the sentence to construct the structure of protein interaction such as '*entity–interaction–entity*.'

For two relation examples P_1, P_2, we use the similarity function $K(P_1, P_2)$ in terms of similarity function of the parent nodes. That is,

$$K(P_1, P_2) = \begin{cases} 0, \text{ if } t(P_1.p, P_2.p) = 0, \\ k(P_1.p, P_2.p) + K_c(P_1.c, P_2.c), \text{ otherwise.} \end{cases} \quad (1)$$

where, $t(\cdot,\cdot)$ is a matching function and $k(\cdot,\cdot)$ is a similarity function on nodes and it has '1' if two nodes have same type and role in the parsing tree, and '0', otherwise. The function, $k(\cdot,\cdot)$, has '1' if the two nodes have same text and '0', otherwise [9].

We also use similarity function K_c of the children node of parse tree in terms of similarities of children subsequences. That is,

$$K_c(P_1.c, P_2.c) = \sum_{\mathbf{i},\mathbf{j}, l(i)=l(j)} \lambda^{d(\mathbf{i})+d(\mathbf{j})} K(P_1[\mathbf{i}], P_2[\mathbf{j}]) \prod_{s=1,\ldots,l(\mathbf{i})} t(P_1[i_s].p, P_2[j_s].p). \quad (2)$$

where, **i** denote a sequence of indices and $d(\mathbf{i}) = l(\mathbf{i}) = i_n - i_1 + 1$ denote length of the sequence **i**. In the Equation 2, the term $P[\mathbf{i}]$ represents the sequence of children $[P[i_1],\ldots,P[i_n]]$ and $K(P_1[\mathbf{i}],P_2[\mathbf{j}])$ stands for summing up $K(P_1[i_s],P_2[j_s])$ for all $s = 1,\ldots,l(\mathbf{i})$ [8]. Equation 2 considers all subsequences of relation example children with matching parents and it accumulates the similarity for each subsequence by adding similarities of the corresponding child examples. The equation also reflects the amount of spread of the subsequences within children sequences by decreasing overall similarity with the factor λ which has the value from 0 to 1.

Generally, the index (**i** and **j**) of two example parse tree will be different. So, we can't use Equation 2 for similarity function K_c of the children node of parse tree directly. Thus, here we use the derived recurrences of Zelenko *et al.* [8] and its original construction recurrences from Lodhi *et al.* [9] to calculate K_c. That is,

$$\begin{aligned}
&K_C = \sum_{q=1,\ldots,\min(m,n)} K_{c,q}(m,n). \\
&K_{c,q}(i,j) = \lambda K_{c,q}(i,j-1) + \sum_{s=1,\ldots i} t(P_1[s].p, P_2[j].p)\lambda^2 C_{q-1}(s-1, j-1, K(P_1[s], P_2[j])). \\
&C_q(i,j,a) = aC_q(i,j) + \sum_{r=1,\ldots q} C_{q,r}(i,j). \\
&C_q(i,j) = \lambda C_q(i,j-1) + C'_q(i,j). \\
&C'_q(i,j) = t(P_1[i], P_2[j])\lambda^2 C_{q-1}(i-1, j-1) + \lambda C'_q(i,j-1). \\
&C_{q,r}(i,j) = \lambda C_{q,r}(i,j-1) + C'_{q,r}(i,j). \\
&C'_{q,r}(i,j) = \begin{cases} t(P_1[i], P_2[j])\lambda^2 C_{q-1,r}(i-1, j-1) + \lambda C'_{q,r}(i,j-1) & \text{if } q \neq r, \\ t(P_1[i], P_2[j])\lambda^2 K(P_1[i], P_2[j]) C_{q-1}(i-1, j-1) + \lambda C'_{q,r}(i,j-1) & \text{if } q = r. \end{cases}
\end{aligned} \quad (3)$$

where, a condition $m \geq n$ is assumed for the number of children of P_1 and P_2. The boundary conditions for calculating K_c are the same with the conditions of Lodhi *et al.* [9] and not presented here.

4 Experimental Results

Data

We used 504 labeled MEDLINE records containing at least one protein interaction as positive examples (50% for training, 50% for testing). About 320 negative examples

were collected from MEDLINE which contain gene or protein names but any interactions at all. The shallow parses of each record was generated by our custom shallow parsing system and the Brill tagger trained with GENIA corpus was used to guarantee more accurate POS tagging. Table 1 shows the statistics of positive examples.

Table 1. The statistics of positive examples for the experiment

Categories	# of Positive Interaction	# of Negative Interaction	Total
# of Interaction	584	302	886

Algorithm Settings

We set parameter λ of kernel computation to 0.5 from experimental search to reflect the amount of spread of interaction structure (gene names, interaction verbs, and protein names). We evaluated kernels based relation extraction result in comparison with the general feature based baseline classifier, the Naïve Bayes. Full parsing results were used as a feature set for the Naïve Bayes. In this paper, this baseline model represents the method of full parsing based relation extraction approach.

Results

The extraction performance was measured by *precision*, *recall*, and *F-measure*. Precision is the ratio of the number of correctly predicted positive examples to the number of predicted positive examples. Recall is the ratio of the number of correctly predicted positive examples to the number of true positive examples. F-measure combines precision and recall as follows:

$$F = \frac{2 * recall * precision}{(recall + precision)}. \tag{4}$$

Table 2 shows the performance results of the protein relation extraction experiments.

Table 2. The extraction performance of relation kernel method on interaction extraction task for extracting interactions represented by major five interaction verbs.

Interaction Verb	Recall (%)		Precision (%)		Total *F* (%)	
	Kernel	*baseline*	*Kernel*	*baseline*	*Kernel*	*baseline*
Interact	81.11	79.16	80.27	79.36	80.69	79.25
Bind	80.83	78.55	82.12	78.54	81.46	78.54
Up-regulate	76.24	72.12	80.14	74.22	78.14	73.15
Suppress	79.92	76.96	81.91	81.84	80.90	79.32
Inhibit	76.91	76.54	79.90	78.79	78.37	77.64
Average	**79.00**	76.66	**80.8**	78.5	**79.91**	77.58

From Table 2, kernel based relation extraction outperform full parsing feature based baseline method (Naïve Bayes) overall. The number of interaction verb which generally describes protein interactions in the biomedical literature and recognized by this experiment was about 150 including ‘activate’, ‘bind’, ‘control’, ‘enhance’, ‘inhibit’, ‘interact’, ‘prevent’, etc. Some verbs occurred frequently but other occurred rarely.

5 Conclusions

In this paper, we presented kernel based protein relation extraction method and the result shows that kernel based method slightly outperform conventional feature based (full parsing information based) approach. The presented relation kernel only requires domain specific knowledge when we define matching function and similarity kernel function. We used the dictionary of protein names and its aliases as a domain specific knowledge. So it would be not difficult to expand this method to other domain by changing this domain knowledge simply.

But, this kernel method on NLP problem has one big drawback for practical use because it requires relatively high computational costs. Thus, the more efficient kernel computation method should be devised.

Acknowledgement

This research was supported in part by the National Research Laboratory Program of the Korea Ministry of Science and Technology.

References

1. Deng, M., et al.: Inferring Domain–Domain Interactions from Protein–Protein Interactions. Genome Res. **12** (2002) 1540–1548
2. Goffeau, A., et al.: Life with 6000 Genes. Science **274** (1996) 546–567
3. Yakushiji, A., et al.: Event Extraction from Biomedical Parsers Using a Full Parser. In: Proc. of the 6th Pacific Symposium on Biocomputing (2001) 408–419
4. Park, J.C., et al.: Bidirectional Incremental Parsing for Automatic Pathway Identification with Combinatory Categorical Grammar. In Proc. of the 6th Pacific Symposium on Biocomputing (2001) 396–407
5. Temkin, J.M., et al.: Extraction of Protein Interaction Information from Unstructured Text Using a Content–free Grammar. Bioinformatics **19(16)** (2003) 2046–2053
6. Leroy, G., et al.: Filling Preposition–based Templates to Capture Information from Medical Abstracts. In: Proc. of the 7th Pacific Symposium on Biocomputing (2002) 350–361
7. Pustejovsky, J., et al.: Robust Relational Parsing Over Biomedical Literature: Extracting Inhibit Relations. In Proc. of the 7th Pacific Symposium on Biocomputing (2002) 362–373
8. Zelenko, D., et al.: Kernel Methods for Relation Extraction. J. Machine Learning Res. **3** (2003) 1083–1106.
9. Lodi, H., et al.: Text Classification Using String Kernels. J. Machine Learning Res. **2** (2002) 419–444.

A Study of Particle Swarm Optimization in Gene Regulatory Networks Inference

Rui Xu[1], Ganesh Venayagamoorthy[2], and Donald C. Wunsch II[1]

[1] Applied Computational Intelligence Laboratory,
[2] Real-Time Power and Intelligent Systems Laboratory,
Department of Electrical and Computer Engineering,
University of Missouri – Rolla, MO 65409, USA
{rxu, ganeshv, dwunsch}@umr.edu

Abstract. Gene regulatory inference from time series gene expression data, generated from DNA microarray, has become increasingly important in investigating genes functions and unveiling fundamental cellular processes. Computational methods in machine learning and neural networks play an active role in analyzing the obtained data. Here, we investigate the performance of particle swarm optimization (PSO) on the reconstruction of gene networks, which is modeled with recurrent neural networks (RNN). The experimental results on a synthetic data set are presented to show the parameter effects of PSO on RNN training and the effectiveness of the proposed method in revealing the gene relations.

1 Introduction

As one of the major advancement in genetic experimental technologies, DNA microarray provides a systematic method to characterize genes functions, discriminate disease types, and test drug effects [1-2]. Particularly, to infer genetic regulatory networks from time series gene expression data has attracted more and more attention, due to its importance in revealing fundamental cellular processes and investigating complicated genes relations [3]. Several computational models have been proposed for regulatory networks inference, such as Boolean networks [4], Bayesian networks [5], dynamic Bayesian networks [6-7], and linear additive regulation models [8]. Here, recurrent neural network (RNN) [9] models, trained with particle swarm optimization (PSO) [10], are used to infer gene regulatory networks from time series gene expression data. In using RNNs for gene network inference, we are mainly concerned with the ability of RNNs to interpret sophisticated temporal behavior and capture nonlinear system dynamics. PSO is an evolutionary computational algorithm for global optimization, which has been proved to be a powerful tool to explore complicated problem space [10-11]. We investigate the effect of PSO parameters on the RNN training and show the effectiveness of the method in revealing the potential gene relations.

The paper is organized as follows. Section 2 describes how PSO is used to train the RNN, which is applied for gene networks modeling. Results on a synthetic data set are summarized in Section 4. Finally, Section 4 concludes the paper.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 648 – 653, 2006.

2 Methods and Systems

A discrete form of the genetic regulatory network, modeled with a recurrent neural network, can be represented as [8, 12-13],

$$e_i(t+\Delta t)=\frac{\Delta t}{\tau_i} f\Big(\sum_{j=1}^{N} w_{ij} e_j(t)+\sum_{k=1}^{K} v_{ik} u_k(t)+\beta_i\Big)+\left(1-\frac{\lambda_i \Delta t}{\tau_i}\right) e_i(t), \tag{1}$$

where e_i is the gene expression level for the i^{th} gene ($1 \le i \le N$, N is the number of genes in the model), $f(\bullet)$ is a nonlinear function (sigmoid function is usually used $f(z)=1/(1+e^{-z})$), w_{ij} represents the effect of the j^{th} gene on the i^{th} gene, u_k is the k^{th} ($1 \le k \le K$, K is the number of external variables) external variable, v_{ik} represents the effect of the k^{th} external variable on the i^{th} gene, τ is the time constant, β is the bias term, and λ is the decay rate parameter. A negative value of w_{ij} represents the inhibition of the j^{th} gene on the i^{th} gene, while a positive value indicates the activation controls. When w_{ij} is zero, the j^{th} gene has no influence on the i^{th} gene.

A simplified form by ignoring the external variables and assuming λ as 1 is represented as

$$e_i(t+\Delta t)=\frac{\Delta t}{\tau_i} \times f\Big(\sum_{j=1}^{N} w_{ij} e_j(t)+\beta_i\Big)+\left(1-\frac{\Delta t}{\tau_i}\right) e_i(t) \ . \tag{2}$$

RNN is well-known as its difficulty in training [14]. Currently, backpropagation through time (BPTT) [15] and evolutionary algorithms [16] are the major methods used for RNN training. Here, we use a novel evolutionary technique, particle swarm optimization (PSO) [10], for RNN parameters learning. Compared with other evolutionary algorithms, PSO has many desirable characteristics, such as the flexibility in balancing global and local search, computational efficiency on both time and memory, and the ease to implement. Particularly, PSO is equipped with the memory mechanism for tracking the previous best solutions and avoiding the possible information loss. It has been shown that PSO require less computational cost and can achieve faster convergence than conventional backpropagation in feedforward neural networks training for approximating nonlinear functions [11].

PSO consists of a swarm of particles, each of which represents a candidate solution in the multidimensional problem space. Each particle i with a position represented as $\mathbf{x}_i$ moves in the space with a corresponding velocity $\mathbf{v}_i$. The basic idea of PSO is to accelerate each particle towards its two best locations at each time step. One is the particle's own previous best position, recorded as a vector $\mathbf{p}_i$, the other is the best position in the whole swarm, represented as $\mathbf{p}_g$.

Given a set of particles $\mathbf{X}=(\mathbf{x}_1,\mathbf{x}_2,...,\mathbf{x}_M)$, where M is the number of particles in the swarm, the i^{th} particle can be represented as a D-dimensional vector $\mathbf{x}_i=(w_{i,11},...,w_{i,N1},w_{i,12},...,w_{i,1N},...,w_{i,NN},\beta_{i,1},...,\beta_{i,N},\tau_{i,1},...,\tau_{i,N})$, where $D=N(N+2)$. The velocity is described as $\mathbf{v}_i=(v_{i,1},v_{i,2},...,v_{i,D})$. A fitness function, used to measure

the deviation of network output $\mathbf{e}(t)$ from the real measurement (target) $\mathbf{d}(t)$, is defined as

$$Fitness(x_i) = \frac{1}{TN}\sum_{t=0}^{T}\sum_{i=1}^{N}(e_i(t) - d_i(t))^2 \quad . \tag{3}$$

The basic procedure of implementing PSO for RNN training can be summarized as follows.

i). Initialize a population of particles with random positions and velocities of D dimensions.

ii). Calculate the estimated gene expression time series based on the RNN model and evaluate the optimization fitness function for each particle.

iii). Compare the fitness value of each particle $Fit(\mathbf{x}_i)$ with $Fit(\mathbf{p}_i)$. If current value is better, reset both $Fit(\mathbf{p}_i)$ and $\mathbf{p}_i$ to the current value and location.

iv). Compare the fitness value of each particle $Fit(\mathbf{x}_i)$ with $Fit(\mathbf{p}_g)$. If current value is better, reset both $Fit(\mathbf{p}_g)$ and $\mathbf{p}_g$ to the current value and location.

v). Update the velocity and position of the particles with the following equations

$$\mathbf{v}_i(t+1) = W_I \times \mathbf{v}_i(t) + c_1 \times \varphi_1 \times (\mathbf{p}_i - \mathbf{x}_i(t)) + c_2 \times \varphi_2 \times (\mathbf{p}_g - \mathbf{x}_i(t)) \, . \tag{4}$$

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(\mathrm{t}) \qquad (1 \le i \le M) \, . \tag{5}$$

where W_I is the inertia weight, c_1 and c_2 are the acceleration constants, and φ_1 and φ_2 are uniform random functions.

vi). Return to step ii) until a stop criterion is satisfied. Usually the learning stops when a maximum number of iterations or high-quality solutions are met.

3 Experimental Results

We applied the proposed method to a synthetic data set consisting of 8 genes. The objective here is to study the effects of the PSO parameters on the RNN training and the possibility in unveiling the connections between genes. Based on the biological assumption that the gene networks are usually sparsely connected, we limit the number of connections for each gene to no more than 4, which leads to 21 non-zero weights. We generated three time series with 30 time points, since multiple series are more effective as demonstrated in [13].

The performance of PSO is dependent on the parameter selection. The inertia weight W_I is designed as a tradeoff between the global and local search. Larger values of W_I facilitate global exploration while lower values encourage local search. W_I can be fixed to some certain value or vary with a random component, such as $W_I = W_{\max} - rand/2$. c_1 and c_2 are known as the cognitive and social components and are used to control the effects of a particle itself and its surrounding environment, which is achieved through adjusting the velocity towards $\mathbf{p}_i$ and $\mathbf{p}_g$. We tested the PSO performance with the commonly recommended values of c_1 and c_2, together with both fixed and varied W_I. We compared the performance in terms of number of iterations required to reach the specified error. Also, we set the maximum number of iterations

as 500. The results, over 20 runs, are summarized in Table 1, which consists of the number of times that the performance exceeds the maximum iterations and the average numbers if it converges. As indicated in Table 1, the most effective performance is achieved when c_1 and c_2 are set as 2.5 and 1.5. For both cases, PSO can achieve the expected accuracy within 40 iterations, except for one run when W_I fixed at 0.7. For other common parameters, the convergence depends more on the initialization and is not consistently stable.

Table 1. Parameter effects on the PSO performance

c_1	c_2	W_I	Performance	
			>500 iterations	Average number of iterations
2	2	Fixed at 0.7	2	39.9
		0.8-*rand*/2	2	43.6
0.5	2	Fixed at 0.7	10	14.2
		0.8-*rand*/2	17	16
2	0.5	Fixed at 0.7	1	55.3
		0.8-*rand*/2	8	181.6
0.5	0.5	Fixed at 0.7	17	15.3
		0.8-*rand*/2	20	-
1.5	2.5	Fixed at 0.7	3	69.5
		0.8-*rand*/2	2	26.2
2.5	1.5	Fixed at 0.7	1	20.4
		0.8-*rand*/2	0	35.8

Although the curse of dimensionality exists in current analysis, as a result of small time points versus thousands of gene measurements, it is still reasonable to identify the weights whose values are non-zeroes from expression data, since a gene is assumed to be regulated only by a limited number of genes. Based on the $\mathbf{p}_g$ solutions of the 1000 runs with random initial values, we discretize the weights into three classes according to the criterion similar to those used in [7]:

- class + representing activation: $\mu_{ij}>\mu+\sigma$ and $|\mu_{ij}|/\sigma_{ij}>\eta$;
- class – representing inhibition: $\mu_{ij}<\mu-\sigma$ and $|\mu_{ij}|/\sigma_{ij}>\eta$;
- class 0 representing absence of regulation: otherwise.

where μ_{ij} and σ^2_{ij} are the mean and variance for the element w_{ij} in the weight connection matrix, μ and σ^2 are the mean and variance of the means of all the 64 elements, and η is a allowed ratio between the mean and standard. The parameters for PSO are set as follows: W_I=0.7, c_1=2.5, c_2=1.5, and η=0.5. The network weights were evolved for 1,000 generations. We summarize the original and the identified weight connection matrix in Table 2. Compared with the original weight matrix, the model can identify 14 out of 21 correct connections in the network. One inhibition connection between node 5 and 6 is indicated as activation relation in the inferred matrix. The results include only 6 false positives, which are wrongly identified as activation or

Table 2. The generated connection matrix (upper panel) and the learned connection matrix (lower panel). Each element in the matrix represents the activation (+), inhibition (-), and absence of regulation (0), between a pair of genes.

w_{ij}							
+	0	0	0	0	0	0	0
+	+	0	0	0	0	0	0
0	-	+	0	0	0	0	0
0	-	-	+	0	0	0	0
0	+	+	-	+	-	0	0
0	0	0	-	+	+	+	-
0	0	0	0	0	0	+	-
0	0	0	0	0	0	0	-
0	0	0	-	0	0	0	0
0	+	+	0	0	0	0	0
0	0	+	0	0	-	0	0
0	-	-	0	0	0	0	0
0	+	+	-	+	+	0	0
0	+	0	-	+	+	0	0
+	0	0	0	+	0	+	-
0	0	0	0	0	0	0	-

inhibition between nodes. Further improvement is achieved by using PSO to evolve both network architecture and weights. The results will be reported elsewhere**.**

4 Conclusions

To understand the genetic regulatory mechanisms is one of the central tasks in molecular genetics. Inference of genetic regulatory networks based on the time series gene expression data has become an important and effective way to achieve this goal. Here, we use particle swarm optimization to train a recurrent neural network, in order to model regulatory systems and unveil potential gene relations. Experimental results show the effects of PSO parameters on the RNN training and suggest effective parameter combination. The potential of the model in gene regulatory network inference is also demonstrated through the analysis. Further work includes the combination of other data sources, particularly at proteins level, and prior information such as promoter elements. The improvement of PSO search, such as the hybrid method with other evolutionary algorithms and the effect of different topologies, is also an interesting direction for future research.

Acknowledgment

Partial support for this research from the National Science Foundation, and from the M.K. Finley Missouri endowment, is gratefully acknowledged by authors.

References

1. McLachlan, G., Do, K., Ambroise, C.: Analyzing Microarray Gene Expression Data. John Wiley & Sons, Inc., Hoboken, New Jersey (2004)
2. Xu, R., Wunsch, D.: Survey of Clustering Algorithms. IEEE Transactions on Neural Networks, 16(3) (2005) 645-678
3. De Jong, H.: Modeling and Simulation of Genetic Regulatory Systems: A Literature Review. Journal of Computational Biology, 9(1) (2002) 67-103
4. Shmulevich, I., Dougherty, E., Zhang, W.: From Boolean to Probabilistic Boolean Networks as Models of Genetic Regulatory Networks. Proceedings of IEEE, 90(11) (2002) 1778-1792
5. Friedman, N., Linial, M., Nachman, I., Pe'er, D.: Using Bayesian Networks to Analyze Expression data. Journal of Computational Biology, 7(3-4) (2000) 601-620
6. Murphy, K., Mian, S.: Modeling Gene Expression Data Using Dynamic Bayesian Networks. Technical Report, Computer Science Division, University of California – Berkeley (1999)
7. Perrin, B., Ralaivola, L., Mazurie, A., Battani, S., Mallet, J., d'Alchė-Buc, F.: Gene Networks Inference Using Dynamic Bayesian Networks. Bioinformatics, 19(2) (2003) 138- 148
8. D'haeseleer, P.: Reconstructing Gene Network from Large Scale Gene Expression Data. Dissertation, University of New Mexico (2000)
9. Kolen, J., Kremer, S.: A Field Guide to Dynamical Recurrent Networks. IEEE Press, Piscataway, New Jersey (2001)
10. Kennedy, J., Eberhart, R., Shi, Y.: Swarm Intelligence. Morgan Kaufmann Publishers, San Francisco, California (2001)
11. Gudise, V., Venayagamoorthy, G.: Comparison of Particle Swarm Optimization and Backpropagation as Training Algorithms for Neural Networks. In Proceedings of the 2003 IEEE Swarm Intelligence Symposium (2003) 110-117
12. Xu, R., Hu, X., Wunsch, D.: Inference of Genetic Regulatory Networks with Recurrent Neural Network Models. In Proceedings of the 26th Annual International Conference of IEEE Engineering in Medicine and Biology Society, Vol. 2, (2004) 2905-2908
13. Wahde, M., Hertz, J.: Coarse–grained Reverse Engineering of Genetic Regulatory Networks. Biosystems, 55(1) (2000) 129–136
14. Jaeger, H.: A Tutorial on Training Recurrent Neural Networks, Covering BPTT, RTRL, EKF and the Echo State Network Approach. GMD Report 159, German National Research Center for Information Technology (2002)
15. Werbos, P.: Backpropagation Through Time: What It Does and How to Do It. Proceedings of IEEE, 78(10) (1990) 1550-1560
16. Yao, X.: Evolving Artificial Neural Networks. Proceedings of the IEEE, 87(9) (1999) 1423-1447

Support Vector Machine Approach for Retained Introns Prediction Using Sequence Features

Huiyu Xia, Jianning Bi, and Yanda Li

MOE Key Laboratory of Bioinformatics / Department of Automation,
Tsinghua University, Beijing, 100084, China
xiahuiyu00@mails.tsinghua.edu.cn

Abstract. It is estimated that 40-60% of human genes undergo alternative splicing. Currently, expressed sequence tags (ESTs) alignment and microarray analysis are the most efficient methods for large-scale detection of alternative splice events. Because of the inherent limitation of these methods, it is hard to detect retained introns using them. Thus, it is highly desirable to predict retained introns using only their own sequence information. In this paper, support vector machine is introduced to predict retained introns merely based on their own sequences. It can achieve a total accuracy of 98.54%. No other data, such as ESTs, are required for the prediction. The results indicate that support vector machine can achieve a reasonable acceptant prediction performance for retained introns with effective rejection of constitutive introns.

1 Introduction

Alternative splicing is an important mechanism expanding proteomic diversity by generating multiple transcripts from a single gene [1]. Recent studies have indicated that 40-60% of human genes undergo alternative splicing [2]. The four major types of alternative splicing are exon skipping, alternative donor splice site (5'ss), alternative acceptor splice site (3'ss), and intron retention. Intron retention is defined as an intronic region that is not spliced out and form part of an exon. Variants containing retained introns are usually believed to be derived from un-spliced or partially spliced pre-mRNAs, thus intron retention is the least studied among the four major types of alternative splicing [3]. However, there are some reports of intron retention events which have biological significance. Michael et al. [4] reported that intron retention is a common event within the human kallikrein (KLK) gene family and the KLK15 splice variant appears to be a candidate biomarker for prostate cancer.

Most large-scale studies used expressed sequence tags (ESTs) or cDNAs for detection of alternative splicing [2]. These studies can help detect the former three types of alternative splicing but not intron retention. It is insufficient to identify a biologically functional intron inclusion by observing a given splice in one EST but not in another, because the latter could be an un-spliced EST [2]. Microarray analysis is also insufficient for the detection of retained intron because probes are usually designed as spanning specific exon-exon junctions [5]. As the limitations of these methods, Hiller et al. presented a method which can predict intron retention events using the annotation of Pfam domains [6]. This method has reliance on Pfam

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 654 – 659, 2006.

domains annotation, which limits its application. Here we describe the application of a machine learning method, support vector machine (SVM), for predicting retained introns. Our results show that SVM can successfully distinguish retained introns from constitutive ones using only basic information of their sequences.

2 Materials and Methods

2.1 Dataset

We extracted an intron set which contains 3,084 simple retained introns and 107,725 constitutive introns (dataset_1) that start with GT and end with AG from AltSplice database (Human release 2) [7]. The simple intron retention event is that the exons which flank the retained intron do not undergo modifications.

It has been suggested that alternative splicing events can be classified as ancient ones that have undergone evolutionary selection and newly emerged ones [8]. To characterize sequence features that are not randomly appeared, we used BLAST to select introns which contain conserved segments in their orthologous genes in mouse from dataset_1 using 10^{-5} expectation value cutoff (dataset_2). Dataset_2 contains 76 retained introns and 34,940 constitutive ones. There are 39 genes, which contains both these two types of introns, including 53 retained introns and 116 constitutive ones. We took these 169 introns as training set since introns in the same gene may exist in the same environment so that we can characterize their sequence features without regard to other splicing factors. Then we took other introns in dataset_2 as testing set, including 23 retained introns and 34,824 constitutive ones.

2.2 Feature Selection

It is reported that retained introns which interrupt coding regions might be under selection for coding potential [3]. We observed this trend in our data set by analyzing (1) intron length, (2) G+C content, (3) stop codon usage, (4) frame-preservation (an exact multiple of 3 in length) and (5) pyrimidine (C or T) intensity of polypyrimidine tract (PPT) related region (see Table 1). It is known that pyrimidine composition of PPT may influence the branch-site (BS) selection [9], which is very important to splicing. An analysis of 19 experimentally proven BSs showed that the average distance between BS and 3'ss is 33-34 nt and the minimal length of PPT is 14 nt [10]. Thus we calculated the number of pyrimidines in the last 35 nt upstream the 3'ss for each intron via a sliding window of 14 nt, taking the maximum pyrimidine content as the pyrimidine intensity of PPT related region for that intron. Thus we got a 5-D (dimension) vector $\vec{x}_B$ of the five basic features mentioned above, given by: $\vec{x}_B = [v_1, v_2, \ldots, v_5]^T$, where v_i (i=1,2,…,5) is the value of the ith feature. Furthermore, nucleotide and dinucleotide composition contain information about intron sequences. These features can be represented by a 20-D vector $\vec{x}_N$, written as: $\vec{x}_N = [p_1, p_2, p_3, p_4, q_1, q_2, \ldots, q_{16}]^T$, where p_i (i=1,2,3,4) and q_i (i=1,2,…,16) are the occurrence frequency of nucleotide (A, G, C or T) and dinucleotide, respectively.

We added $\vec{x}_N$ on $\vec{x}_B$ to form a combined feature vector $\vec{x}$, defined by:

$$\vec{x} = \begin{bmatrix} \vec{x}_B \\ \vec{x}_N \end{bmatrix} . \tag{1}$$

In this paper, feature vector $\vec{x}_B$ and $\vec{x}$ are accepted to represent an intron sequence.

Table 1. Features differentiating between retained and constitutive introns

	Retained intron	Constitutive intron
Average length (nt)	324	3695
Average G+C content	53.64%	45.46%
Present stop codon in all three phases	47.17% (25/53)	97.41% (113/116)
Frame-preserving	49.06% (26/53)	41.38% (48/116)
Average pyrimidine intensity of PPT related region	83.96%	85.41%

2.3 Support Vector Machine

SVM is a popular machine learning algorithm, which was initially proposed by Vapnik based on statistical learning theory [11]. It has been successfully applied to some bioinformatics investigations [12]. Intuitively, the basic idea of SVM is mapping data into a high-dimensional space, and then constructs a hyperplane as the decision surface between positive and negative data. The actual mapping is achieved through some kernel functions, such as:

linear kernel: $K(x_i, x_j) = x_i^T x_j$, (2)

polynomial kernel: $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d$, $\gamma > 0$, (3)

radial basis function (RBF) kernel: $K(x_i, x_j) = exp\left(-\gamma \left\| x_i - x_j \right\|^2\right)$, $\gamma > 0$, (4)

sigmoid kernel: $K(x_i, x_j) = tanh(\gamma x_i^T x_j + r)$. (5)

We used software SVM^{light} [13] to implement SVM. This software can be downloaded from http://svmlight.joachims.org/.

2.4 Performance Assessment

Sensitivity (SN), specificity (SP) and total accuracy (TA) are used to evaluate the performance of our algorithm. We used TP (True Positive) and FN (False Negative) to denote the number of retained introns that are predicted as retained and constitutive, respectively. Similarly, TN (True Negative) and FP (False Positive) are used to denote the number of constitutive introns that are predicted as constitutive and retained, respectively. Then SN and SP are defined as:

$$SN = TP / (TP + FN) , \quad (6)$$

$$SP = TN / (TN + FP) . \quad (7)$$

That is, SN and SP are the proportion of retained and constitutive introns which are correctly predicted, respectively. And the TA is defined as:

$$TA = (TP + TN) / (TP + FN + TN + FP) . \quad (8)$$

In addition, since we try to discriminate retained introns from a large amount of constitutive ones, high discrimination power at low false positive rate (FPR) is needed. To this end, we further used precision to measure the prediction performance. Precision (PS) is the proportion of retained introns to the whole predicted retained introns, defined as:

$$PS = TP / (TP + FP) . \quad (9)$$

In our study, a leave-one-out test is performed on the training set to identify the prediction performance of each SVM kernel function with various parameter values.

3 Results and Discussion

3.1 SVM Kernel and Parameters Selection

We used four popular kernel functions mentioned above. For polynomial and sigmoid kernel function, we chose $r = 0$ and $\gamma = 1/k$ where k is the number of features. A grid search was performed over the values of penalty parameter C for each kernel, the parameter d for polynomial kernel, and for RBF kernel, also the γ parameter. Fig. 1 gives an example for the influence of penalty parameter C upon the prediction performance on training set. Furthermore, various values of cost-factor j, by which training errors on positive examples outweigh errors on negative ones, has been tested. The value of j was selected under the rule that as good as prediction performance can be achieved at a sensitivity no less than 26/53. The best leave-one-out test performance based on $\vec{x}_B$ was achieved by the polynomial kernel of degree 3, with penalty parameter $C = 8$ and cost-factor $j = 0.2$; while the best result based on $\vec{x}$ was obtained by the RBF kernel, with $\gamma = 0.04$, $C = 2$, and $j = 0.2$.

3.2 Prediction Results of SVM

SVM prediction results on testing set are shown in Table 2. Results show that no matter which feature vector is used the rate of overall correct prediction on testing set is higher than 98% at a low FPR around 1.5%. Furthermore, though the SN of SVM using $\vec{x}$ is the same as that using $\vec{x}_B$, the SP of the former is higher than the latter. Thus, SVM using $\vec{x}$ produces higher TA and PS. This suggests that the five basic features in $\vec{x}_B$ are much more informative for predicting retained introns than the others; while the nucleotide and dinucleotide composition features which contain some sequence order information can improve the prediction performance slightly.

We also used our SVM classifier to predict retained introns which don't have conserved segments, that is, data in dataset_1 but not in dataset_2. We get a TA of 87.03% and the SN, SP and PS are 37.53%, 89.07% and 12.43%, respectively. These results show that though our SVM classifier is based on the introns which have conserved segments, the features we used are not merely conservation-based, but related with some common sequence characters of each type of introns. Our classifier can be effectively used in other introns to some extent.

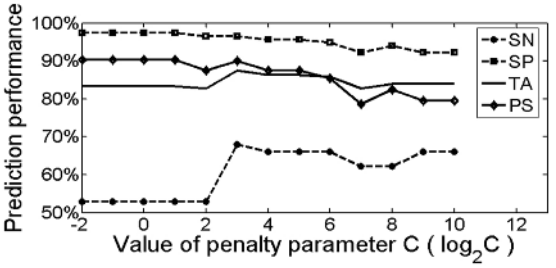


Fig. 1. Prediction performance with various values of penalty parameter C on training set (polynomial kernel function, using $\vec{x}_B$, with $d = 3$, $r = 0$, $\gamma = 0.2$, and $j = 1$)

Table 2. The prediction performance comparison between SVM and neural network (NN)

	Feature Vector	SN (%)	SP (%)	TA (%)	PS (%)
SVM	$\vec{x}_B$	52.17	98.45	98.42	2.17
	$\vec{x}$	52.17	98.57	98.54	2.36
NN	$\vec{x}_B$	56.52	91.84	91.82	0.46
	$\vec{x}$	60.87	93.63	93.61	0.63

3.3 Comparison with Other Method

In order to examine the prediction performance of our method, we compared our SVM algorithm with a three-layer feed-forward backpropagation neural network (NN) classifier. The comparison results are also shown in Table 2. For all features vector $\vec{x}$, the SN of NN is 8.7% higher than SVM while the SP of NN is nearly 5% lower than SVM. Since we are trying to identify retained introns within a population in which the great majority of introns are constitutive, high SP is much more important than high SN when assess the prediction performance. We can also see that the TA of NN is about 5% lower than SVM, and the PS of NN is about a quarter of that of SVM. The comparison results using basic feature vector are similar. The comparison indicates that our SVM classifier outperformed the NN method.

4 Conclusions

In this paper, we analyzed some features of intron sequences and introduced SVM for distinguishing retained introns from constitutive ones using these features. The rate

of overall correct identification of our SVM classifier is 98.54% at a low false positive rate, which is superior to neural network. This result implies that we can predict retained introns to a considerably extent using machine learning method solely based on the features of their sequences, and help discover their biological significance, even when there are no ESTs that indicate their retention.

Acknowledgement

This work is supported by National Natural Science Foundation of China (60234020).

References

1. Graveley, B.R.: Alternative Splicing: Increasing Diversity in the Proteomic World. Trends Genet., **17**(2) (2001) 100-107
2. Modrek, B., Lee, C.: A Genomic View of Alternative Splicing. Nat. Genet., **30**(1) (2002) 13-19
3. Galante, P.A., Sakabe, N.J., Kirschbaum-Slager, N., de Souza, S.J.: Detection and Evaluation of Intron Retention Events in the Human Transcriptome. RNA, **10**(5) (2004) 757-765
4. Michael, I.P., Kurlender, L., Memari, N., et al.: Intron Retention: a Common Splicing Event within the Human Kallikrein Gene Family. Clin. Chem., **51**(3) (2005) 506-515
5. Johnson, J.M., Castle, J., Garrett-Engele, P., et al.: Genome-Wide Survey of Human Alternative Pre-mRNA Splicing with Exon Junction Microarrays. Science, **302**(5653) (2003) 2141-2144
6. Hiller, M., Huse, K., Platzer, M., Backofen, R.: Non-EST Based Prediction of Exon Skipping and Intron Retention Events Using Pfam Information. Nucleic Acids Res., **33**(17) (2005) 5611-5621
7. Thanaraj, T.A., Stamm, S., Clark, F., Riethoven, J.J., Le Texier, V., Muilu, J.: ASD: the Alternative Splicing Database. Nucleic Acids Res., **32**(Database Issue) (2004) D64-D69
8. Wen, F., Li, F., Xia, H.Y., Lu, X. Zhang, X.G., Li, Y.D.: The Impact of Very Short Alternative Splicing on Protein Structures and Functions in the Human Genome. Trends Genet., **20**(5) (2004) 232-236
9. Norton, P.A.: Polypyrimidine Tract Sequences Direct Selection of Alternative Branch Sites and Influence Protein Binding. Nucleic Acids Res., **22**(19) (1994) 3854-3860
10. Kol, G., Lev-Maor, G., Ast, G.: Human-Mouse Comparative Analysis Reveals that Branch-Site Plasticity Contributes to Splicing Regulation. Hum. Mol. Genet., **14**(11) (2005) 1559-1568
11. Vapnik, V.N.: Statistical Learning Theory. Wiley, New York (1998)
12. Dror, G., Sorek, R., Shamir, R.: Accurate Identification of Alternatively Spliced Exons Using Support Vector Machine. Bioinformatics, **21**(7) (2005) 897-901
13. Joachims, T.: Making Large-Scale SVM Learning Practical. In: Schölkopf, B., Burges, C., Smola, A. (eds.): Advances in Kernel Methods - Support Vector Learning, Chapter 11, MIT Press, Cambridge, MA (1999) 169-184

Prediction of Protein-Protein Interface Residues Using Sequence Neighborhood and Surface Properties

Yasir Arafat, Joarder Kamruzzaman, and Gour Karmakar

Faculty of Information Technology, Monash University, Australia 3842
{Yasir.Arafat, Joarder.Kamruzzaman,
Gour.Karmakar}@infotech.monash.edu.au

Abstract. An increasing number of protein structures with unknown functions have been solved in the recent years. But understanding the mechanism of protein-protein association still remains one of the biggest problems in structural bioinformatics. Significant research efforts have been dedicated to the identification of protein binding sites and detecting specific amino acid residues which have important connotations ranging from rational drug design to analysis of metabolic and signal transduction networks. In this paper, we present a support vector machine (SVM) based model to predict interface residues from amino acid sequences using sequence neighborhood and surface properties. Experiments with a number of surface properties reveal that the prediction accuracy enhances when residue interface propensity and coil interface propensity of amino acid residues are incorporated in the prediction model which is an improvement over the previously proposed model based on sequence neighborhood only. We also analyzed the effectiveness of a recently proposed coding scheme [1] of secondary structures on the proposed model.

1 Introduction

Proteins rarely act in isolation. It performs biological functions with other proteins, DNA, RNA and compounds. The complexity at different levels of biological systems arises not only from the number of proteins in the organism but also from the combinatorial interactions among them [2]. Identifying the characteristics of interfacial sites is a requirement for understanding the molecular recognition process. In addition, it provides important clues to the function of a protein and can reduce the search space required by docking algorithm. However, protein structures determined using X-ray or NMR methods is extremely small as compared to those of sequences. Hence computational methods are of great interest in predicting interface residues.

Different physical and chemical aspects of proteins have been investigated in a number of studies for several years. Based on these characteristics of known protein binding sites, several methods have been developed for predicting these sites which include methods based on the analyzing the hydrophobicity, solvation, protrusion, accessibility, charge distribution [2, 3] and results show that no single parameter differentiates interface from other surface [3]. Sequence and structural conservatism, sequence and spatial neighborhood have also been investigated [4, 5].

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 660 – 666, 2006.

None of the previous proposed sequence based methods attempted to incorporate other possible physio-chemcial properties with sequence neighborhood information and investigate their combined influence on prediction performance. In this study, we trained support vector machine (SVM) based model to distinguish between interface and noninterface residues using physiochemical properties in conjunction with sequence neighborhood. Result shows that these properties could be used along with sequence neighborhood successfully to improve the prediction accuracy.

2 Interface Residue Propensities

Studies have demonstrated a number of physical and chemical properties, e.g., secondary structures, hydrophobicity, surface accessibility, residue interface propensity, etc. to have presence and influence at the binding sites of protein-protein interaction. In our study, we investigated secondary structures, hydrophobicity and residue interface propensity (RIP). A number of studies show that both α-helix and β-sheet has lower propensity to be on binding site as compared to coil [3, 6]. Hydrophobicity is also an important property of protein surface and hydrophobic residues tend to form clusters in binding sites. Hydrophobicity of the target residue was also investigated. RIP was chosen as it has been reported to show most impact among all properties representing protein binding sites [7]. A dataset of 61 proteins (see experimental setup) was used to derive the RIP for each of the 20 amino acids using the following relationship:

$$\text{Interface Propensity of amino acid, } r = (A/B)/(C/D) \quad (1)$$

where A = number of interface residues associated with r, B = total number of interface residues, C = number of surface residues associated with r and D = total number of surface residues. Coil interface propensity was calculated by taking the ratio of secondary structure distribution (coil) [6] and coil conformation propensity [1].

3 Proposed Prediction Model

3.1 Input Vector Representation

The input to SVM is a vector of the identities of 9 contiguous residues of the target residue. Each of the 9 residues in the window is represented by a 20 bit vector with 1 bit for each letter of the 20 letter amino acid alphabet. Thus input to SVM classifier was consisting of 180 bit vector. The Boolean output produced by the classifier is either 1 (denoting interface residue) or 0 (denoting non-interface residue). The window size of nine (four at each side) is chosen due to the fact supported in other studies [6]. The size of the input vector to SVM is increased by one for an extra physio-chemical property of the target residue added each time. Figure 1 shows the proposed model with two properties of target residue making input vector size of 182 (=180+2). Once the model was built, unknown sequence was fed as input and the model then predicts whether a target residue belongs to interface or non-interface.

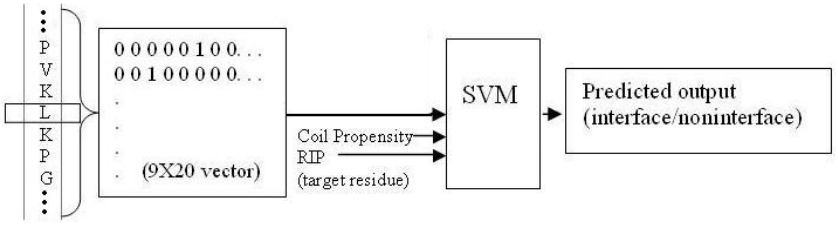


Fig. 1. The proposed model for interface residue prediction

3.2 SVM

SVM attempts to locate a separating hyperplane by converting input vector space $\Re^n$ to higher directional space [8]. The input data Z is transformed to higher dimensional space. The separating hyperplane in higher dimension space satisfy:

$$W.Z_i + b = 0 \tag{2}$$

To maximize the margin between two classes (3) and (4) are used.

$$Max \quad 1 \Big/ \|W\|^2 \tag{3}$$

Subject to the condition

$$y_i(W.Z_i + b) \geq 1 \tag{4}$$

This is essentially a quadratic programming problem. Using the Kuhn-Tucker condition [21] and LaGrange Multiplier methods it is equivalent to solving the dual problem in (5).

$$Max\left[\sum_{i=1}^{l} a_i - \frac{1}{2}\sum_{i=1}^{l} a_i a_j a_i y_j K(x_i x_j)\right] \tag{5}$$

where $0 \leq a_i \leq C$, l = number of inputs, i=1…l and $\sum_{i=1}^{l} a_i y_i = 0$. In (5) $K(x_i x_j)$ is called kernel function. In our experiments, RBF kernel function was used. We optimised regularization parameter C and γ in RBF kernel using technique proposed by Lee and Lin [9].

4 Experimental Setup

4.1.1 Datasets

In this study, we used 61 nonredundant proteins. This dataset of proteins was selected from the 70 protein complexes of eight different categories according to the scheme of Chakrabarti and Janin [10] after removal of proteins with fewer than 10 residues

and keeping proteins with sequence identity < 30%. These eight different categories of heterocomplexes are: antibody-antigen, protease-inhibitor, enzyme complexes, large protein complexes, G-proteins, cell cycle, signal transduction and miscellaneous. The proteins used in this study are listed in the Appendix.

4.1.2 Surface and Interface Residues

Surface residues were obtained from the corresponding PDB files of proteins using the server InterProSurf[1] and interface residues of the corresponding PDB files were obtained from the protein-protein interaction server of UCL[2]. From our dataset, we obtained 1789 positive examples corresponding to interface residues and 4261 negative examples corresponding to non-interface residues.

4.2 Performance Measures

To evaluate the effectiveness of this approach, we performed leave-one-out cross validation. In each experiment, an SVM classifier was trained using a training set of 1650 randomly chosen interface residues and equal number of non-interface residues from 60 proteins. The trained classifier was then used to classify the surface residues of the remaining protein into interface or non-interface residues. The following measures were used to evaluate the model.

$$\text{Sensitivity+} = \frac{TP}{TP+FN}, \text{Specificity+} = \frac{TP}{TP+FP}, \text{Accuracy} = \frac{TP+TN}{N} \text{ and}$$

$$\text{Correlation coefficient} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FN)(TP+FP)(TN+FP)(TN+FN)}}$$

where, TP = number of true positives, FP = number of false positives, TN = number of true negatives, FN = the number of false negatives. Sensitivity+ measures the fraction of interface residues that are identified as such and Specificity+ measures the fraction of the predicted interface residues that are actually interface residues. Accuracy computes the estimated probability of correct predictions. Correlation coefficient (CC), which ranges from -1 to +1, is a measure of how well the predicted class labels correlate with the actual class labels. Performance was also measured in terms of receiver operating characteristics (ROC) analysis.

5 Experiments and Results

An investigation on sequence neighbourhood of protein-protein interface residues in a set of 33 proteins reported that 98% of protein-protein interface residues have at least one additional residue within 4 positions of N- or C- terminal and 74% have at least 4 [11]. In this study, window size 9 (4 on each side of the target residue) was chosen as suggested in the above study.

We used secondary structures interface propensities along with sequence neighbourhood. Table 1 compares results obtained using sequence neighbourhood

[1] http://curie.utmb.edu/prosurf.html
[2] http://www.biochem.ucl.ac.uk/bsm/PP/server

only (shown under the SVM column heading) and incorporating other properties. Results indicate no significant preference for β-sheet to be on the binding site while preference for α-helix is a bit stronger, though in some previous experiments contradictory results were reported in favour of both α-helix and β-sheets [6, 12]. However, such properties were not yet investigated along with sequence neighbourhood. But both the previous investigations [6, 12] found strong preference for coil to be on the binding site. Therefore, coil interface propensity was chosen as additional inputs to SVM.

Table 1. Performance of the prediction model incorporating different physio-chemical properties. SVM stands for model with sequence neighbourhood only.

	SVM	SVM				
		α helix	β sheet	coil	RIP	RIP + coil
Sensitivity+ (%)	61.2	61	61	62	63	63.1
Specificity+ (%)	40.4	41	40	41	42	42
Accuracy (%)	61.8	62	61	62	62	63.3
Correlation Coefficient	0.216	0.22	0.20	0.22	0.24	0.242

The residue interface propensities (RIP) revealed that hydrophobic residues favour interface locations, in addition the charged residue arginine and polar aromatic residues such as tryptophan, tyrosine and histidine show preferences for the interface. This property was chosen because it incorporates effect of both hydrophobic and hydrophilic effects. When RIP was used along with sequence neighbourhood the sensitivity+ and specificity+ increased by 1.8% and 1.6% respectively with an increase of CC by 2.4%. When both RIP and coil interface propensity were incorporated in the model, results show improvement in sensitivity+, specificity+, accuracy and CC by 1.9%, 1.6%, 1.5 and 2.6%, respectively.

Gallet *et al.* [13] proposed a method to identify interface residues using sequence hydrophobicity analysis. On a dataset of non-redundant proteins they reported 51% accuracy with correlation coefficient -0.02. In our study, the accuracy of prediction using SVM is 61.2% with correlation coefficient of 0.216. When residue interface and coil interface propensity were added accuracy improved to 63.1% with correlation coefficient 0.242. This indicates that the proposed SVM model could lead to better prediction performance.

A control experiment was also carried out to compare its performance with random predictions in which case class labels were randomly shuffled to destroy the attributes-class relationship in the original dataset. The correlation coefficient obtained on the class label shuffled dataset is -0.003 (as compared to 0.242 in the original database). This means that our model performs significantly better than random prediction. A protein can interact with multiple proteins, and residues identified here as false positives could be in fact interface residues of other partners not known yet other than their known partners in databases.

In addition to the above measures, model performance was also evaluated in terms of receiver operating characteristics (ROC). The area under the ROC curve (AUC) of

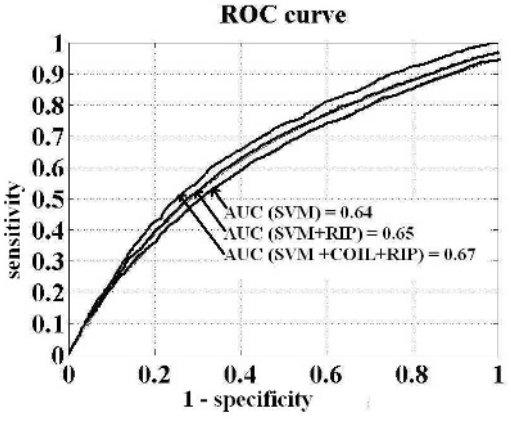


Fig. 2. ROC plot. SVM, SVM with RIP and SVM with RIP and coil interface propensity.

the model with and without different properties is shows in Figure 2 which also supports improved performance when properties are added.

We also investigated the effectiveness of a recently proposed coding scheme [1]. This coding encodes each amino acid into a four-element vector. Results show degrades in performances (Table 2). This may be because the relative weight of residues in a sequence is not represented well by this scheme. To investigate the effect of hydrophobicity, we used it along with sequence neighbourhood but no significant improvements found (results not shown). This again establishes the fact that hydrophobicity distribution of individual amino acid residues may not have any significant impact on proteins binding sites.

Table 2. Comparison of SVM with residue coded by binary numbers and with residue coded with properties [1]

	SVM (20-bit vector)	SVM (4-element vector)
Sensitivity+ (%)	61.2	54
Specificity+ (%)	40.4	35
Accuracy (%)	61.8	57
Correlation Coefficient	0.216	0.11

6 Conclusion

This method falls in the broad category of sequence based methods. Predicting protein interface residues from protein sequence and its unbound structures are of great interests because large numbers of protein complexes are still not solved experimentally.

One of the aims of this paper was to investigate whether properties of protein surface residues could be used along with sequence neighborhood to improve prediction of binding sites. Experimental results demonstrate that the use of coil

interface propensity and RIP in conjunction with sequence neighborhood as inputs to SVM predictor leads to better performance measured in terms of widely used performance metrics.

There are other residue properties that also need to be investigated. Neuvirth *et al*. [6] considered 13 different properties of residues and atoms. Future studies will include incorporation of other properties with sequence neighborhood as inputs to the classifier and investigating their impact in predicting interface residues.

References

1. Wang, L.H., Liu, J., Li, Y.F., Zhou, H.B.: Predicting Protein Secondary Structure by a Support Vector Machine Based on a New Coding Scheme. Genome Informatics 15 (2004) 181-190
2. Valencia, A., Pazos, F.: Computational Methods for the Prediction of Protein Interactions. Current Opinion in Structural Biology 12 (2002) 368-373
3. Jones, S., Thornton, J.M.: Prediction of Protein–Protein Interaction Sites Using Patch Analysis. J. Mol. Biol. 272 (1997) 133–143
4. Koike, A., Takagi, T.: Prediction of Protein-Protein Interaction Sites using Support Vector Machine. Protein Engineering, Design and Selection 17 (2004) 165-173
5. Ma, B., ELkayam, T., Wolfson, H., Nussinov, R.: Protein-Protein Interactions: Structurally Conserved Residues Distinguish between Binding Sites and Exposed Protein Surfaces. PNAS 100 (2003) 5772-5777
6. Neuvirth, H., Ran, R., Neuvirth, G.: Promate: A Structure Based Prediction Program to Identify the Location of Protein-Protein Binding Sites. J. Mol. Biol. 338 (2004) 181-199
7. Bradford, J.R., Westhead, D.R.: Improved Prediction of Protein-Protein Binding Sites using a Support Vector Machines Approach. Bioinformatics 21 (2005) 1487–1494
8. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, New York (1995)
9. Hsu, C.W., Lin, C.J.: A Comparison of Methods for Multi-Class Support Vector Machines, IEEE Transactions on Neural Networks 13 (2002) 415–425
10. Chakrabarti, P., Janin, J.: Dissecting Protein–Protein Recognition Sites. J. Mol. Biol. 272 (2002) 132–143
11. Ofran, Y., Rost, B.: Predicted Protein–Protein Interaction Sites from Local Sequence Information. FEBS Lett. 544 (2003b) 236–239
12. Jones, S., Thornton, J.M.: Protein-Protein Interactions: A Review of Protein Dimer Structures. Prog Biophys Mol Biol 63 (1995) 31-65
13. Gallet, X., Charloteaux, B., Thomas, A., Brasseur, R.: A Fast Method to Predict Protein Interaction Sites from Sequences. J. Mol. Biol. 302 (2000) 917–926

Appendix

List of 61 proteins used in this study: 1a0oB, 1a2kA, 1a2kC, 1agrE, 1aipA, 1aipC, 1ak4A, 1ak4C, 1ao7A, 1brsA, 1brsD, 1bthP, 1choI, 1danT, 1dfjE, 1dfjE, 1dfjI, 1dhkA, 1dhkB, 1dkgD, 1efnA, 1efnB, 1fc2C, 1fc2D, 1finA, 1finB, 1fleI, 1fssA, 1fssB, 1glaF, 1glaG, 1glaG, 1gotB, 1guaA, 1hiaA, 1hiaB, 1hiaI, 1hwgA, 1hwgB, 1kb5A, 1melA, 1nsnS, 1qfuA, 1qfuB, 1danU, 1sebA, 1sebD, 1stfE, 1stfI, 1tbqR, 1tgsI, 1tocR, 1tx4A, 1tx4B, 1udiE, 1udiI, 1ydrE, 2jelH, 2jelP, 2sicI, 4htcI

Prediction of Protein Subcellular Multi-locations with a Min-Max Modular Support Vector Machine

Yang Yang and Bao-Liang Lu*

Department of Computer Science and Engineering,
Shanghai Jiao Tong University,
800 Dong Chuan Rd., Shanghai 200240, China
{alayman, bllu}@sjtu.edu.cn

Abstract. How to predict subcellular multi-locations of proteins with machine learning techniques is a challenging problem in computational biology community. Regarding the protein multi-location problem as a multi-label pattern classification problem, we propose a new predicting method for dealing with the protein subcellular localization problem in this paper. Two key points of the proposed method are to divide a seriously unbalanced multi-location problem into a number of more balanced two-class subproblems by using the part-versus-part task decomposition approach, and learn all of the subproblems by using the min-max modular support vector machine (M^3-SVM). To evaluate the effectiveness of the proposed method, we perform experiments on yeast protein data set by using two kinds of task decomposition strategies and three kinds of feature extraction methods. The experimental results demonstrate that our method achieves the highest prediction accuracy, which is much better than that obtained by the existing approach based on the traditional support vector machine.

1 Introduction

The localization of a protein in a cell is very important for understanding its function. Due to the difficulties of conducting biological experiments to determine the subcellular locations, a lot of efforts have been made to develop automatic tools for localization. As the numbers of new genome and protein sequences in the public databases have increased dramatically in recent years, methods based on analyzing protein sequences have been largely developed. In 1994, Nakashima and Nishikawa discriminated intracellular and extracellular proteins successfully by amino acid composition and residue-pair frequencies [1]. Till now, many locations have been successfully discriminated and various pattern classification and machine learning methods have been used, such as Mahalanobis distance

* To whome correspondence should be addressed. This work was supported by the National Natural Science Foundation of China under the grants NSFC 60375022 and NSFC 60473040.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 667–673, 2006.

[2], neural network [3], hidden Markov model (HMM) [4] and support vector machine [5].

Most of these researches focus on mono-locational proteins, i.e., proteins existing in only one location. However, a lot of proteins bear multi-locational characteristics. According to our statistics of Swiss-Prot database [8], there are more than five thousands proteins locating in more than one location. Recently, Cai and Chou first tackled the classification of multi-locational proteins in yeast [10]. They used GO-FunD-PseAA method, which hybridizes gene ontology, functional domain composition and pseudo-amino acid composition approach. Although this method improves the prediction accuracy a lot, it fails to give a general classification method for this multi-location problem. In addition, there are a large portion of proteins lack the information like GO and FunD.

In this paper, we apply M^3-SVM to solve the problem. Several feature extraction methods are also discussed, including amino acid composition, amino acid pair composition and segmentation method. A series of standard measures are used to evaluate the classification performance. The experimental results show that using M^3-SVM and the part-versus-part strategy can get a much higher prediction accuracy than traditional SVM and other classification methods.

2 Our Method

2.1 Min-Max Modular Support Vector Machine

The min-max modular network has been shown to be an efficient classifier, especially in solving large-scale and complex multi-class pattern classification problems [6]. It divides a complex classification problem into many small independent two-class classification problems, which can be learned parallelly without communication with each other. And then it integrates these modules to get a final solution to the original problem according to two module combination rules, namely minimization and maximization principles. The min-max modular support vector machine [7], which use SVM as base classifier and M^3 network structure, has been successfully used in many pattern classification problems, such as text categorization, human face recognition and industrial fault image detection.

2.2 Part-Versus-Part Strategy

As for multi-class problems, one-versus-rest decomposition is usually used [9]. Given a K-class multi-label problem, its training set is as follows:

$$\mathcal{X} = \{(x_m, t_m)\}_{m=1}^{l}, t_m = \{t_m^k\}, k = 1, ..., \tau_m \tag{1}$$

where $x_m \in \mathrm{R}^n$ is the mth sample in the data set, t_m is the label set of x_m, t_m^k is the kth label of x_m, and τ_m denotes the number of labels of x_m.

Decompose the K-class multi-label problem T to K two-class problems $T_i, i = 1, ..., K$. The training set of T_i is defined as

$$\mathcal{X}_i = \{(x_m^{i+}, +1)\}_{m=1}^{l_i^+} \cup \{(x_m^{i-}, -1)\}_{m=1}^{l_i^-} \tag{2}$$

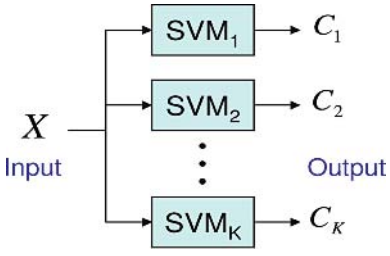


Fig. 1. A multi-label problem divided into several two-class subproblems with the one-versus-rest strategy

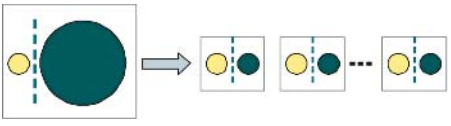


Fig. 2. Decomposition of a seriously unbalanced multi-lable problem into a number of balanced two-class subproblems with the part-versus-part strategy for M^3-SVM

where l_i^+ is the number of positive samples of the two-class problem T_i, and l_i^- is the number of the negative samples. For T_i, positive samples are those whose label sets contain the label C_i and negative samples are the remaining ones. Figure 1 depicts a multi-label problem divided into several two-class subproblems with the one-versus-rest strategy and SVM as the two-class classifier.

Considering that many biological problems have unbalanced data distribution for the classes, such as proteins occurring in Cytoplasmic, Nuclear and Plasma membrane being much more than those in other locations, we adopt part-versus-part strategy here [7]. An important advantage of the part-versus-part method over existing popular pairwise-classification approach is that a large-scale two-class subproblem can be further divided into a number of relatively smaller and balanced two-class subproblems, and fast training of SVMs on massive multi-class classification problems can be easily implemented in a massively parallel way.

The part-versus-part decomposition is straightforward which further decomposes the two-class problems to smaller ones as shown in Figure 2. For a two-class problem T_i, its positive and negative training set $\mathcal{X}_i^+$ and $\mathcal{X}_i^-$ can be further decomposed into N_i^+ and N_i^- subsets, where $1 \le N_i^+ \le l_i^+$, $1 \le N_i^- \le l_i^-$.

$$\mathcal{X}_{ij}^+ = \{(x_m^+, +1)\}_{m=1}^{l_{ij}^+}, j = 1, ..., N_i^+ \tag{3}$$

$$\mathcal{X}_{ij}^- = \{(x_m^-, -1)\}_{m=1}^{l_{ij}^-}, j = 1, ..., N_i^- \tag{4}$$

The l_{ij}^+ and l_{ij}^- are numbers of samples in $\mathcal{X}_{ij}^+$ and $\mathcal{X}_{ij}^-$, respectively.

After the original problem is divided into related balanced subproblems, each of which can be handled by a SVM. And We use min-max modular network to organize all the subproblem together.

2.3 Task Decomposition

Task decomposition is a key problem for M^3-SVM. A good decomposition method can maintain or even improve the generalization performance. In this paper, we use two kinds of methods, namely random decomposition and hyperplane decomposition [13]. The former is simple and straightforward. Given a

specific module size, it chooses samples randomly from the training set to form a new smaller training set. This method can not obtain a stable performance and may hurt the generalization ability sometimes.

As for the hyperplane decomposition method, a series of specific hyperplanes are introduced and the training data are sorted according to their distances to the hyperplanes. Then the ordered sequence of training data will be divided into relatively balanced subsets.

3 Results and Discussion

3.1 Data Set

We conducted experiments on a data set collected from Swiss-Prot according to the list of codes of the 4,709 budding yeast proteins given in [10]. None of the proteins included here has 40% sequence identity with any other. Because some sequences are absent in the database, the data set we used is 19 ones less than theirs. But it would not has much impact on the overall accuracy. The distribution of the data set is listed in Tables 1 and 2. We adopted 10-fold cross-validation test. All experiments were performed on a 3GHz Pentium 4 PC with 2GB RAM.

Table 1. Numbers of proteins for every class

Location	Sequence No.	Location	Sequence No.
Actin	29	Lipid particle	19
Bud	23	Microtubule	20
Bud neck	59	Mitochondrion	491
Cell periphery	104	Nuclear periphery	59
Cytoplasm	1565	Nucleolus	156
Early Golgi	51	Nucleus	1323
Endosome	43	Peroxisome	20
ER	271	Punctuate composite	123
ER to Golgi	6	Spindle pole	58
Golgi	40	Vacuolar membrane	54
Late Golgi	36	Vacuole	129
Summation of all classes		4679	
Number of different proteins		3536	

Table 2. Distribution of multi-locational proteins

Number of Locations	1	2	3	4	5
Number of Sequences	2465	1007	57	6	1

3.2 Experimental Results

A proper representation for protein sequences is very important to the classification of proteins. Researchers have developed a lot of features extraction methods

for protein sequences. Here we experimented three approaches: amino acid composition (AAC), amino acid pair composition (AAP) and segmentation method (SEG)[12]. Each protein in the data set of l proteins will be characterized by a vector $v_i (i = 1, \cdots, l)$, which represents sequence features.

AAC is a conventional method which converts a protein sequence $\mathcal{S}$ to a vector $v = \{a_1, a_2, \ldots, a_{20}\}$, where $a_i (1 \leq i \leq 20)$ reflects the occurrence frequency of one of the 20 amino acids ({A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}) in a protein. AAP contains 400 components, each representing an amino acid pair composition by counting two adjacent amino acids overlappingly. The SEG method regards protein sequences as text and segment them nonoverlappingly by match words in an established dictionary. The length of words used for segmentation is not limited to two but determined according to need. Moreover, it does not use all the k-mers but select informative ones by some criteria. Here we establish a dictionary of 30 words, including 20 amino acid, 5 most frequent amino acid pair and 5 3-kmers. The SEG method performs the best with traditional SVM. All of the three methods can obtain better prediction accuracy using M^3-SVM.

To evaluate the effectiveness of the multi-label classification comprehensively, we use recall, precision and F_1 measure for each class. We trained the classifier with a RBF kernel and set the module size of M^3-SVM to 100. Since the task

Table 3. Results by using SVM and M^3-SVM

Location	M^3-SVM(R)			M^3-SVM(H)			SVM		
	R	P	F_1	R	P	F_1	R	P	F_1
Actin	13.8	4.1	6.3	17.2	13.9	**15.4**	0.0	0.0	0.0
Bud	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Bud neck	76.3	3.5	6.7	54.2	5.3	**9.7**	1.7	33.3	3.2
Cell periphery	15.4	6.6	9.2	26.9	7.5	**11.7**	1.0	9.1	1.7
Cytoplasm	96.3	46.4	62.4	87.5	46.8	61.0	80.3	56.2	**66.3**
Early Golgi	33.3	7.8	12.6	35.3	7.7	**12.6**	0.0	0.0	0.0
Endosome	9.3	2.5	3.9	20.9	2.8	**5.0**	0.0	0.0	0.0
Endoplasmic reticulum	52.8	23.4	32.4	64.9	20.8	31.5	33.9	45.1	**38.7**
ER to Golgi	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Golgi	10.0	2.9	4.5	5.0	3.9	4.4	2.5	50.0	**4.8**
Late Golgi	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Lipid particle	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Microtubule	5.0	2.0	**2.8**	0.0	0.0	0.0	0.0	0.0	0.0
Mitochondrion	75.4	23.9	36.3	69.5	31.8	43.7	40.5	55.3	**46.8**
Nuclear periphery	6.8	2.7	3.8	27.1	5.3	8.8	8.5	38.5	**13.9**
Nucleolus	61.5	8.7	15.2	58.3	9.8	**16.8**	1.3	28.6	2.5
Nucleus	46.3	44.5	45.4	86.3	40.9	**55.5**	41.2	57.1	47.8
Peroxisome	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Punctate composite	12.2	3.0	4.8	54.5	4.7	**8.7**	0.0	0.0	0.0
Spindle pole	31.0	10.3	15.5	51.7	10.0	**16.7**	1.7	100.0	3.4
Vacuolar membrane	11.1	4.0	5.9	14.8	5.7	**8.2**	0.0	0.0	0.0
Vacuole	41.1	11.9	**18.4**	63.6	10.7	18.3	3.9	17.9	6.4

decomposition is one of the two key problems of M^3-SVM, two kinds of task decomposition methods were experimented. One is the random task decomposition strategy, the other is hyperplane task decomposition strategy. The detailed values of recall, precision, F_1 of 22 classes are given in Tables 3. Here amino acid composition method is adopted. Let M^3-SVM(R) stand for M^3-SVM with the random strategy, and M^3-SVM(H) the hyperplane strategy.

From the experimental results, we can observe that M^3-SVM(H) performs the best among the three methods. And many small classes were successfully discriminated by using M^3-SVM with part-versus-part decomposition, while SVM classified all the proteins to several big classes.

3.3 Comparison with Other Methods

Chou and Cai has reported that the likelihood of hitting the localization of a protein in budding yeast could be as high as 90% [14] using GO-FunD-PseAA method. In their method, gene ontology and functional domain knowledge are used for prediction. Since we aim to propose a general classification method, we make comparisons with other methods based on the same feature vectors, i.e., the amino acid composition. The Least Euclidean Distance algorithm, Least Hamming Distance algorithm and ProtLoc predictor obtained success rates of 13.89, 14.03 and 13.95%, respectively [10]. According to our experimental results, traditional SVM obtained overall success rate of 46%. The M^3-SVM(H) and M^3-SVM(R) obtained accuracies of 73% and 64%, respectively, which are much higher than other classification methods.

4 Conclusions and Future Work

This study focuses on seeking efficient classification method to predict subcellular locations for proteins existing in one or more locations. We apply M^3-SVM and part-versus-part strategy to solve this multi-label problem. And several feature extraction methods for protein sequences are compared. The experiments were conducted on a data set of yeast proteins. Results show that the classification method we proposed is superior to other methods on a series of performance measures and improves the accuracy significantly.

As a future work, we will consider referring other available field knowledge to get more precise prediction results. And now we are constructing large-scale data sets covering various species from Swiss-Prot. We believe that our methods will be competent in solving new complex classification tasks.

Acknowledgements

The authors thank Mr. Ken Chen for his helpful work on data preparation.

References

1. Nakashima, H., Nishikawa, K.: Discrimination of Intracellular and Extracellular Proteins Using Amino Acid Composition and Residue-pair Frequencies. J. Mol. Biol. 238 (1994) 54-61
2. Cedano, J., Aloy, P., Perez-Pons, J.A., Querol, E.: Relation Between Amino Acid Composition and Cellular Location of Proteins. J. Mol. Biol. 266 (1997) 594-600
3. Reinhardt, A., Hubbard, T.: Using Neural Networks for Prediction of the Subcellular Location of Proteins. Nucleic Acids Research 26 (1998) 2230-2236
4. Fujiwara, Y., Asogawa, M., Nakai, K.: Prediction of Mitochondrial Targeting Signals Using Hidden Markov Models. Genome Informatics (1997) 53-60
5. Hua, S., Sun, Z.: Support Vector Machine Approach for Protein Subcellular Localization Prediction. Bioinformatics 17 (2001) 721-728
6. Lu, B.L., Ito, M.: Task Decomposition and Module Combination Based on Class Relations: a Modular Neural Network for Pattern Classification. IEEE Transactions on Neural Networks 10 (1999) 1244-1256
7. Lu, B.L., Wang, K.A., Utiyama, M., Isahara, H.: A Part-Versus-Part Method for Massively Parallel Training of Support Vector Machines. Proceedings of International Joint Conference on Neural Networks (2005) 735-740
8. Chen, K., Liang, W.M., Lu, B.L.: Data Analysis of Swiss-Prot Database. BCMI Technical Report BCMI-TR-0501, Shanghai Jiao Tong University (2005)
9. Joachims, T.: Learning to Classify Text Using Support Vector Machine: Method, Theory, and Algorithms. Kluwer Academic Publishers (2002)
10. Chou, K.C., Cai, Y.D.: Prediction of Protein Subcellular Locations by GO-FunD-PseAA Predictor. Biochemical and Biophysical Research Communications 320 (2004) 1236-1239
11. Apweiler, R.: The InterPro Database, an Integrated Documentation Resource for Protein Families, Domains and Functional Sites. Nucleic Acids Research 29 (2001) 37-40
12. Yang, Y., Lu, B.L.: Extracting Features from Protein Sequences Using Chinese Segmentation Techniques for Subcellular Localization. Proceedings of 2005 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (2005) 288-295
13. Liu, F.Y., Wu, K., Zhao, H., Lu, B.L.: Fast Text Categorization with Min-Max Modular Support Vector Machines. Proceedings of International Joint Conference on Neural Networks (2005) 570-575
14. Chou, K.C., Cai, Y.D.: Predicting Protein Localization in Budding Yeast. Bioinformatics 21(7) (2005) 944-950

Prediction of Protein Domains from Sequence Information Using Support Vector Machines

Shuxue Zou, Yanxin Huang, Yan Wang, and Chunguang Zhou

College of Computer Science and Technology,
Jilin University, 130012, China
sandror@163.com, cgzhou@jlu.edu.cn

Abstract. Guessing the boundaries of structural domains has been an important and challenging problem in experimental and computational structural biology. Predictions were based on intuition, biochemical properties, statistics, sequence homology and other aspects of predicted protein structure. In this paper a promising method for detecting the domain structure of a protein from sequence information alone was presented. The method is based on analyzing multiple sequence alignments that are derived from a database search. Multiple measures are defined to quantify the domain information content of each position along the sequence and are combined into a single predictor using support vector machines. The overall accuracy of the method for a single protein chains dataset, is about 85%. The result demonstrates that the utility of the method can help not only in predicting the complete 3D structure of a protein but also in the study of proteins' building blocks and for functional analysis.

1 Introduction

The prediction of protein structure is that the three dimensional structure of protein is predicted from the sequences of amino acid. Domain structure is one of the structure levels of protein, which is considered as the fundamental unit of the protein structure, folding, function, evolution and design [1]. A protein may be comprised of a single domain or several different domains, or several copies of the same domain. It is the domain structure of a protein that determines its function, the biological pathways in which it is involved and the molecules it interacts with.

Detecting the domain structure of a protein is a challenging problem. Previous methods for domain detection could be roughly classified into five categories: (i) Methods based on the use of similarity searches and multiple alignments to delineate domain boundaries. Methods like Domainer [2], DOMO [3] and Domination [4] fall in this category. (ii) Methods that rely on expert experiences on known protein structures to identify the domains. SCOP [5], CATH [6] fall in this category. (iii) Methods that try to infer domain boundaries by using the dimensional structure of proteins. DALI [7], PDP [8] and DomainParser [9] are examples of this approach. (iv)Methods that utilize the representational special structure to construct models like HMMs to identify other members of the family such as Pfam [10] and SMART [11]. (v) Other methods, that do

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 674 – 681, 2006.

not fall into any of the previous categories (domain guess by size [12] and based on the conformational entropy [13]).

Generally structural information can help in detecting the domain structure of a protein. Domain delineation based on structure is currently best done manually by experts and the SCOP domain classification, which is based on extensive expert knowledge, is an excellent example. However, structural information is available for only a small portion of the protein space. And with the current rapid growth in the number of sequences with unknown structures, it is very important not only to accurately define protein structural domains, but to predict domain boundaries on the basis of amino-acid sequence alone.

Given a query sequence, our algorithm starts by searching the protein sequence database and generating a multiple alignment of all significant hits. The columns of the multiple alignment are analyzed using a variety of sources to define scores that reflect the domain-information-content of alignment columns. Information theory based principles are employed to maximize the information content. These scores are then combined using a support vector machine to label single columns as core-domain or boundary positions The overall accuracy of the method for a single protein chains dataset, is about 85 %. The result demonstrates that the utility of the method can help not only in predicting the complete 3D structure of a protein but also in the study of proteins' building blocks and for functional analysis.

2 Datasets

The SCOP database, the up to date version 1.65, which include 20,619 proteins and 54,745 chains. For consideration of the personal computer we focus on the proteins which have only one chain. The datasets are selected according to the statistical results respectively on the single domain and more than two domains. Then among the 3960 protein chains that have single domain, the length between 40 and 300 and representative fold 142 chains are, whose homologous proteins are 55.13% of all of the proteins with single chain and single domain. And there are 66 chains from 596 protein chains with the length between 200 and 400 and representative fold, whose homologous proteins are 49.79% of all of the proteins with more chains and more domains.

3 Feature Extraction

3.1 Domain Definitions

For each protein chain we defined the domain positions to be the positions that are at least x residues apart from a domain boundary. Domain boundaries are obtained from SCOP definitions where for a SCOP definition of the form $(start_1; end_1)::(start_n; end_n)$ the domain boundaries are set to $(end_i+start_i+1)/2$. All positions that are within x residues from domain boundaries are considered boundary positions. This process allows us to classify all the positions in the proteins being considered as domain or boundary positions.

3.2 The Domain-Information-Content of an Alignment Column

Sequence alignments provide a powerful way to compare novel sequences. BLAST 2.0, (Basic Local Alignment Search Tool), provides a method for rapid searching of nucleotide and protein databases. PSI-BLAST is an important tool for predicting both biochemical activities and function from sequence relationships.

To quantify the likelihood that a sequence position is part of a domain, or at the boundary of a domain we defined several measures based on the multiple alignments that we believe reflect structural properties of proteins and would therefore be informative of the domain structure of the seed protein. While some of these measures are more directly related to structural properties than others, none of these measures actually rely on structural information, as our goal was to devise a novel technique that can suggest domain delineation from sequence information alone.

3.2.1 Conservation Measures

Multiple alignments of protein families can expose the core positions along the backbone that are crucial to stabilize the protein structure, or play an important functional role. These positions tend to be more conserved than others and strongly favor amino acids with similar and very specific physio-chemical properties.

Amino acid entropy: One possible measure of the conservation of an alignment column is given by the entropy of the corresponding distribution. For a given probability distribution P over the set A of the 20 amino acids P = (p1, p2, …, p20)t, the entropy is defined as

$$E_a(P) = -\sum_{i=1}^{20} p_i \log_2 p_i \tag{1}$$

This is a measure of the disorder or uncertainty we have about the type of amino acid in each position. In information theory terms, the entropy is the average number of bits needed to encode an arbitrary member of A .

Class entropy: Quite frequently one may observe positions in protein families that have a preference for a class of amino acids, all of which have similar physio-chemical properties. The amino acid entropy measure is not effective in such cases since it ignores amino acid similarities. We tried a new criterion and method for amino acid classification [14]. The 20 amino acid residues are divided into subsets according to the criterion which is based on the description of protein evolution by a Markov process and the corresponding matrix of instantaneous replacement rates. In order for groupings to be interpretable in terms of the processes of evolutionary replacement of amino acids, the standard amino acid replacement matrices PAM and WAG are used, which lead to different two groupings. The grouping from the ordered PAM matrix is {A,R,N,D,Q,E,G,H,I,K,M,P,S,T,V}, {C}, {L,F,Y}, {W}. And the grouping from the ordered WAG matrix is {A,R,N,D,Q,E,G,H,K,P,S,T}, {C,I,V}, {M,L,F,Y}, {W}.Given the set C of amino acid classes and the empirical probabilities P the class entropy is defined in a similar way to the amino acid entropy.

$$E_c(P) = -\sum_{i \in C} p_i \log_2 p_i \tag{2}$$

3.2.2 Consistency and Correlation Measures

Since protein domains are believed to be stable building blocks of protein folds, it is reasonable to assume that all appearances of a domain in database sequences will maintain the domain's integrity. However, domains may be coupled with other domains and therefore a sequence alignment will not be informative. Integrating the information from multiple sequences can generate a strong signal, indicative of domain boundaries by detecting changes in sequence participation and evolutionary divergence. We quote several different measures from [15]. These measures quantify the consistency and correlation of neighboring columns in an alignment.

Consistency: This simple coarse-grained measure is based on sequence counts. The measure is defined as the difference in sequence counts of a column and the average of the surrounding columns in a window of size w. If c_k is the sequence count in position k then

$$Consistenc \quad y(k) = \left| c_i - \frac{1}{2\omega} \sum_{i \neq k, |i-k| \leq \omega} c_i \right| \tag{3}$$

Asymmetric correlation: This is a more refined measure that considers the consistency of individual sequences and sums their contributions. To measure the correlation of two columns we first transform each alignment column into a binary vector of dimension n (the number of sequences in the alignment) with 1's signifying aligned residues and 0's for gaps. Given two binary vectors $\vec{u}$ and $\vec{v}$ their asymmetric correlation (bitwise AND) is defined as

$$Corr_a(\vec{u}, \vec{v}) = \langle \vec{u}, \vec{v} \rangle = \sum_{i=1}^{n} u_i \cdot v_i \tag{4}$$

Symmetric correlation: the asymmetric correlation measure does not reward for sequences that are missing from both positions. However, these may reinforce a weak signal based only on participating sequences. The symmetric correlation measure corrects this by using bitwise XNOR when comparing two alignment columns, i.e.

$$Corr_s(\vec{u}, \vec{v}) = \sum_{i=1}^{n} \delta(u_i, v_i) \tag{5}$$

where δ is the delta function $\delta(x, y) = 1 \Leftrightarrow x = y$.

To enhance the signal and smooth random fluctuations the contributions of all positions in a local neighborhood around a sequence position are added, and all correlation measures for an alignment column are calculated as the average correlation over a window of size w centered at the column (the parameter w is optimized, as described in section Appendix).

3.2.3 Measures of Structural Flexibility

Regions of substantial structural flexibility in a protein often correspond to domain boundaries where the structure is usually exposed and less constrained. *Indel entropy:* In a multiple alignment of related sequences, positions with indels with respect to the seed sequence indicate regions where there is a certain level of structural flexibility. The larger the number of insertions and the more prominent the variability in the indel

length at a position the more flexible we would expect the structure to be in that region. The indel entropy based on the distribution of indel lengths is defined as

$$E_g(P) = -\sum_i p_i \log_2 p_i \tag{6}$$

where the p_i are the various indel lengths seen at a position.

3.3 The Conformational Entropy of Seed Sequence

In this work, we refer to a new method [13] to predict domain boundary from protein sequence alone using a simple physical approach based on the fact that the protein unique three dimensional structure is a result of the balance between the gain of attractive native interactions and the loss of conformational entropy, that is, that the topology of the chain determines how much the chain entropy is lost as native interactions are formed. Considering here the conformational entropy as the number of degrees of freedom on the angles ϕ,ψ and χ for each amino acid along the chain, the method for domain boundary prediction relies on finding the extreme values in a latent entropy profile. And the number of degrees of freedom for the angles ϕ,ψ and χ is determined for each residue in Table 1.

Table 1. Number of degrees of freedom for the angles ϕ,ψ andχ for each amino acid

aa	A	E	Q	D	N	L	G	K	S	V	R	T	P	I	M	F	Y	C	W	H
n	2	5	5	4	4	4	3	6	4	3	6	4	1	4	5	4	5	4	4	4

(aa) The name of the residue shown in one letter code, (n) the number of degrees of freedom.

4 The SVMs Learning System

Support Vector machines (SVMs) are a new statistical learning technique that can be seen as a new method for training classifiers based on polynomial functions, radial basis functions, neural networks, splines or other functions. At present Sequential Minimal Optimization (SMO) algorithm is a quite efficient method for training large-scale SVMs . The SVMs trained by improved SMO algorithm [15] are used to identify the boundary positions of protein domains in this paper. Without loss of generality we choose the SVMs coupled with the RBF kernel widely used in pattern recognition. Then two SVMs parameters have to be chosen prior to training: σ^2, the RBF kernel smoothing parameter, and C the (soft margin) SVMs training error to generalization performance trade-off parameter.

$$K(\vec{x}, \vec{x}_i) = \exp\left(\frac{-\left\|\vec{x} - \vec{x}_i\right\|^2}{2\sigma^2}\right) \tag{7}$$

To find the best pair of C and σ^2 .over some ranges, a grid search using cross-validation is employed. Since there are only limited numbers of instances obtained by the time-consuming alignment running on PC, we adopt a k-fold (k = 5 in this study) cross-validation procedure for an unbiased evaluation, which is a common

technique used in the machine learning community for model selection. all the examples in the dataset are eventually used for both training and testing. In the k-fold cross-validation, we do not divide the amino acids but the sequence into k subsets of approximately equal size for the integrality of the protein. The performance for each classifier is evaluated in terms of accuracy, which is defined as follows:

$$arrucacy \quad = (TP + TN)/N \tag{8}$$

where *TP, TN* and *N* denote true positive, true negative and the total of the samples, respectively.

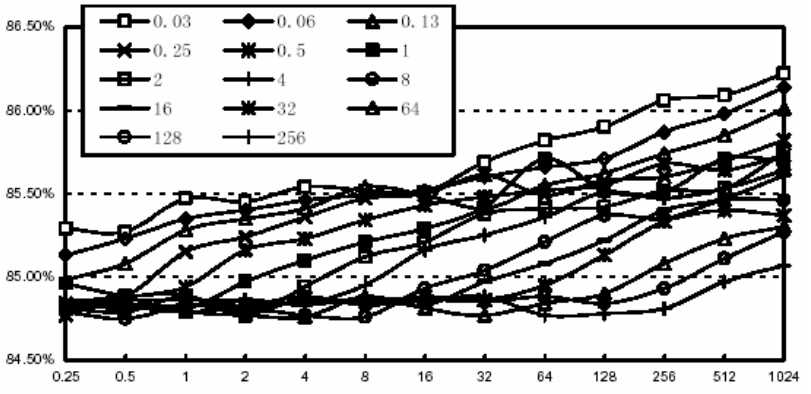


Fig. 1. Curves of the accuracy vary with the combinations (C , σ^2)

5 Analysis of Results

The SVMs parameters grid search was iterated over the following values: $C = \{2^{-2}, 2^{-1}, \cdots, 2^{9}, 2^{10}\}$ and $\sigma^2 = \{2^{-5}, 2^{-4}, \cdots, 2^{7}, 2^{8}\}$. The smaller values for C and the larger values for σ^2 make the accuracies increase as well as result in the more simple decision surfaces , and vice-versa. The accuracy gets the maximal value of 86.22% at $C = 1024,\ \ \sigma^2 = 0.03$ after running 104 minutes, while it gets the minimal value of 84.75% at $C = 0.5,\ \ \sigma^2 = 128$ after only running 2 minutes. As the result of the minimal value only 1.47 percent less than the maximal one, the learning system is not highly sensitive to the parameters of C and σ^2. After the further observation to the Figure 1 the accuracy gets the local maximum at $C = 64,\ \ \sigma^2 = 1$ as well as the consume time does not exceed 3 minutes. The combination is regarded as the parameters of the SVMs for the trade-off between the accuracy and the time.

6 Summary and Outlook

In this paper we presented a promising method for detecting the domain structure of a protein from sequence information alone. Firstly each protein in the selected dataset has

to been aligned in sequence databases. To quantify the likelihood that a sequence position is part of a domain, or at the boundary of a domain we defined several measures based on the multiple alignments that reflect structural properties of proteins. And the scores are calculated not only from the single column of alignments but also among several columns. Besides the conformational entropy of the seed sequence is considered. Further more the information theory principles are used to optimize the scores. Finally support vector machines with RBF kernel are trained to learn a non-linear mapping from the original scores to a single output. And a grid search using cross-validation is employed in order to identify a good C and σ^2 .pair that are the two SVMs parameters. At last the prediction of protein domain from sequence gets the accuracy about 85% by the method, which hopefully show a significant improvement for the biological macromolecular constructions in bioinformatics. For further validate our method the datasets have to been enlarged in the near future. The majority of the samples in SVMs called strong class, and the minority called weak class. The boundary of protein domain is the weak class. Another possible improvement is the SVMs specifically for our learning system.

Acknowledgement

This paper is supported by the national nature science foundation of China (Grant No. 60433020), “symbol computation and knowledge engineering” Lab of Chinese ministry of education, and the project “985” of Jilin University.

References

1. Rose, G. D.: Hierarchic Organization of Domains in Globular Proteins. J. Mol. Biol. 134 (1979) 447-470
2. Sonnhammer, E. L., Kahn, D.: Modular Arrangement of Proteins as Inferred From Analysis of Homology. Protein Sci.3 (1994) 482-492
3. Gracy, J., Argos, P.: Automated Protein Sequence Database Classification. I. Integration of Copositional Similarity Search, Local Similarity Search and Multiple Sequence Alignment. II. Delineation of domain boundries from sequence similarity. Bioinformatics, 14 (1998) 164-187
4. George, R. A. and Heringa, J.: Protein Domain Identification and Improved Sequence Similarity Searching Ssing PSI-BLAST. Proteins 48(2002) 672-681
5. Murzin, G., Brenner, S. E., Hubbard, T. and Chothia, C.: SCOP: a Structural Classification of Proteins Database for the Investigation of Sequences and Structures. J. Mol. Biol. 247(1995) 536–540
6. Orengo, A., Michie, A. D., Jones, S., Jones, D. T.: Swindells M. B. and Thornton J. M., CATH-a Hierarchic Classification of Protein Domain Structures. Structure 5 (1997) 1093-1108
7. Holm, L. and Sander, C.: Mapping the Protein Universe. Science 273 (1996) 595–602
8. Alexandrov N. and Shindyalov I. PDP:protein domain parser. Bioinf. 19 (2003) 429-430
9. Xu, Y. and Xu, D.: Protein Domain Decomposition Using a Graph-Theoretic Approach. Bioinformatics 16 (2000) 1091–1104

10. Bateman, A., Birney, E., Durbin, R., Eddy, S. R., Finn, R. D. and Sonnhammer, E. L.: Pfam 3.1: 1313 Multiple Alignments and Profile HMMs Match the Majority of Proteins. Nucl. Acids Res. 27 (1999) 260-262
11. Ponting, P., Schultz, J., Milpetz, F. and Bork, P.: SMART: Identification and Annotation of domains from Signaling and Extracellular Protein Sequences. Nucl. Acids Res. (1999) 27 229-232
12. Wheelan, S. J., Marchler-Bauer, A. and Bryant, S. H.: Domain Size Distributions Can Predict Domain Boundaries. Bioinformatics 16 (2000) 613-618
13. Galzitskaya, O.V., Melnik, B.S.: Prediction of Protein Domain Boundaries from Sequence alone. Protein Science 12 (2003) 696–701
14. Kosiol, C., Goldman, N. and Buttimore, N. H.: A New Criterion and Method for Amino Acid Classification. Journal of Theoretical Biology 228 (2004) 97-106
15. Nagaragan, N. and Yona, G.: Automatic Prediction of Protein Domains from Sequence Information Using a Hybrid Learn System. Bioinformatics 1 (2004) 1–27

Appendix

The parameters mentioned in section 3.3 should be optimized for the learning system. One is CLASS , the two different values in calculating class entropy. The other three are smoothing factors the windows of size w mentioned in eq. (3),(4),(5), respectively. The running time and accuracy with the combinations of the parameters are listed in Table 2.

Table 2. Results with several parameters

CLASS	CSW	CAW	CW	TIME	ACCURACY	CLASS	CSW	CAW	CW	TIME	ACCURACY
1	2	2	2	193	85.08%	2	2	2	2	195	85.05%
1	2	3	2	194	85.06%	2	2	3	2	196	85.07%
1	3	3	2	199	85.08%	2	3	3	2	195	85.08%
1	3	3	3	208	85.19%	2	3	3	3	199	85.19%
1	4	3	3	208	85.16%	2	4	3	3	201	85.17%
1	4	4	3	206	85.19%	2	4	4	3	202	85.17%
1	4	4	4	205	85.24%	2	4	4	4	200	85.25%
1	5	5	4	207	85.21%	2	5	5	4	200	85.23%
1	5	5	5	198	85.39%	2	5	5	5	197	85.37%
1	6	6	5	200	85.37%	2	6	6	5	198	85.39%
1	6	6	6	197	85.48%	2	6	6	6	193	85.47%
1	7	7	7	192	85.50%	2	7	7	7	190	85.50%
1	8	8	8	190	85.54%	***2***	***8***	***8***	***8***	***184***	***85.55%***
1	9	9	9	188	85.55%	2	9	9	9	185	85.52%

Using a Neural Networking Method to Predict the Protein Phosphorylation Sites with Specific Kinase

Kunpeng Zhang[1,2], Yun Xu[1,2], Yifei Shen[1,2], and Guoliang Chen[1,2]

[1] Department of Computer Science,
University of Science and Technology of China, 230026 Hefei, China
[2] Laboratory of Computer Science
Institute of Software Chinese Academy of Sciences,
100080 Beiing, China
kpzhang@mail.ustc.edu.cn

Abstract. Protein phosphorylation at Serine(S), Threonine(T) or Tyrosine(Y) residues is an important reversible post-translational modification, and it is an important mechanism for modulating(regulating) many cellular processes such as proliferation, differentiation and apoptosis. Experimental identification of phosphorylation site is labor-intensive and often limited by the availability and optimization of enzymatic reaction. In silico prediction methods may facilitate the identification of potential phosphorylation sites with ease. Methods based on primary protein sequences is much desirable and popular for its convenience and fast speed. It is obvious that structural-based methods show excellent performance, however, the 3-D structure information of protein is very limited compared to the huge number of protein in the public databases. Here we present a novel and accurate computational method named NNPhosPhoK: sequence and structural-based neural network method of protein phosphorylation sites prediction with considering specific kinase. The data in this paper is from Phospho.ELM[1].We test NNPhosPhoK with both simulational and real data, whatever in speed or in accuracy, we can realize that NNPhosPhoK shows greater computational ability with superior performance compared to two existing phosphorylation sites prediction system: ScanSite 2.0[2] and PredPhospho[3].

Keywords: Neural Network; phosphorylation site predicting; protein kinase; protein structure; PKA; IKK.

1 Introduction

Reversible protein phosphorylation is considered as being a key event in many signal transduction pathways of biological systems.Biochemically, this includes a transfer of a phosphate moiety from a high-energy phosphate donor like ATP, GTP, etc. to an acceptor amino acid like tyrosine,serine or threonine in a protein chain. Eukaryotic protein kinases(short for PKs in this paper) form one of the largest multigene families, and the full sequencing of the human genome has

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 682–689, 2006.

allowed the identification of almost all human protein kinases, representing about $1.7 - 2\%$ of all human genes[4]. Protein kinases catalyze the phosphorylation events that are essential for the regulation of cellular processes like metabolism, proliferation, differentiation and apoptosis[5-9]. It is estimated that one-third of all the proteins could be phosphorylated, and about half of kinome were disease- or cancer-related by chromosomal mapping[4].

It is in urgent need to identify the substrates accompanied with their phosphorylation sites in large-scale Phosphoproteome, which would help drug design greatly in order to improve our health. PKs' substrates and their phosphorylation sites can be acquired in vivo or in vitro, although they are time-consuming, labor-intensive and expensive. Recently, there are many interest in developing novel technologies to acquire phosphorylation sites in large scale, such as mass spectrometry (MS)[10], peptide microarray[11], and phosphospecific proteolysis [12]. On the other hand, in silico prediction of phosphorylation sites based on known primary protein sequences which have some the experimentally verified phosphorylation sites catalyzed by specific PKs is desirable and popular for its convenience and fast speed. So far, there are many tools implemented by computational biologist who almost applied sequences-based methods, such as ScanSite[2], PredPhospho[3], GPS[4], etc.

In this paper, we present a novel method combinating protein structure information and sequence information, and apply neural network technology. Obviously it is superior to methods without considering structure information. This can greatly help biologists working efficiently. Crystallization studies indicate that a region, between seven and twelve residues in size, surrounding the acceptor residue contacts the kinase active site[13].

2 Materials and Methods

2.1 Data Collection

We get the data set of phosphorylation sites from Phospho.ELM[1]. It contains 1552 substrates entries from different species covering 840 tyrosine, 2769 serine and 648 threonine instances. After clustering some homologous Pks into a unique group which contains at least 10 phosphorylated sites, we got 42 PK families/PK groups including ABL, AMPK, ATM, BTK, CaM-II, CDK, CDK1, Cdk2, CHK2, CK-1, CK-2, DNA-PK, EGFR, FAK, Fyn, GRK, GRK-2, GRK-3, GRK-5, GSK-3beta, IKK, IR, Jak2, JNK, Lck, Lyn, MAP2K1, MAPK, MAPK14, PDK-1, PGK/cGK, PK, PKA, PKB, PKC, Rho-kinase, S6K, SKF, Src, Syk, TRKB, ZAP70, etc.

2.2 Analysis of Protein Primary Structure Information

Scoring Strategy. The biochemical characteristic of a phosphorylation site mainly depend on the neighboring amino acids around the acceptor residue. Phosphorylation site peptide $PSP(m, n)$ means m upstream and n downstream. In this article, we only consider the nine-peptide sequence $PSP(4, 4)$ according

the Crystallization studies. The amino acid substitute matrix BLOSUM62[14] is selected to evaluate the similarity between two nine-peptide sequences. For two amino acids a and b, the similarity between two peptides A and B with length 9 amino acids is defined as:

$$S = \frac{S_{real} - S_{rand}}{S_{iden} - S_{rand}}$$
$$S_{real} = \sum_{i=1}^{9} Score(A[i], B[i]), \text{ for the two real sequences } A \text{ and } B$$
$$S_{iden} = \frac{\sum_{i=1}^{9} Score(A[i], A[i]) + Score(B[i], B[i])}{2}$$
$$S_{rand} \text{ is the score of shuffled sequence } A \text{ and } B$$

$Score(a, b)$ is defined as the substitute score between the two sequences in the BLOSUM62[14]. The distance between them is defined as: $D(A, B) = -logS$. As described in the above, the two peptides have similar biochemical characteristic if their score is high enough.

Clustering Strategy. Taking all the $PSP(4, 4)$ of a given kinase K as nodes,we connect them with edge whose weight is the distance between the pair of nodes. We adopt the Markov Cluster Algorithm(MCL for short)[15] to partition the acquired graph into several clusters. After this operation, we get a set of clusters of phosphorylation site peptides for each kinase, respectively. For each nine-peptide sequence, we can calculate the score between it and each subset in the cluster which produced by applying MCL to given kinase nine-peptide. And we chose the maximum as the sequence information of this nine-peptide be phosphorylated, of course, this maximum is also the first input item in our neural network system which will formulate in the following.

2.3 Protein Secondary Structure Information

In [16], we can analyze statistically that most of the phosphorylation sites were located in coil structures, while the ordered helical and extended structures were much less presented. It summarized in Table 1. Nevertheless, one cannot conclude from this study that all the phosphorylation sites must be in coiled conformation since $20 - 30\%$ of the sequence were predicted to be in helical or extended structure. The related analysis in [16] also revealed that the phosphorylated Serine residues seem to more frequently predicted in coiled conformation than Tyreonine residues. On the other hand, both Serine and Threonine are located in more coiled sequence than Tyrosine. This may be related to different bulkiness or hydrophobicity of these amino acid side chains, probably influencing the spatial protein structure. Since the spatial structure of many phosphorylated proteins are not available, some secondary structure prediction algorithms (PHDSec[17], nnpredict[18], and predator[19]) were used in the analysis. In this article, we choose nnpredict[18] as the prediction tool. It takes the primary sequence as an input and predict for each residue either a helical (H), an extended (E) or an unordered (coiled, C) conformation. This secondary structure information can be added into our neural network system as the second input item to improve the prediction accuracy.

Table 1. Result *of* secondary structure prediction of phosphorylation sequences

Residue	Method	Coil(C)	Helical(H)	Extended(E)
Serine	nnpredict	81%	12%	7%
	Predator	68%	24%	8%
	PHDSec	76%	17%	6%
Threonine	nnpredict	78%	13%	9%
	Predator	61%	27%	12
	PHDSec	70%	18%	12%
Tyrosine	nnpredict	66%	21%	13%
	Predator	58%	24%	18%
	PHDSec	75%	14%	11%

2.4 Protein Tertiary Structure Information

In this article, we use the surface solvent accessibility to describe the location of those phosphorylated residues in protein tertiary structure. This approach classifies residues as either buried in protein core or exposed on the surface. In[16], surface solvent exposure of the phosphorylation sites was estimated by the PHDAcc solvent accessibility prediction algorithm[20]. The PHDAcc predicts each residue in the sequence to be in a buried (B) or exposed (E) state in a water solution. This article also summarized it in Table 2. From the table 2, we can see that about $65 - 70\%$ of all phosphorylation residues were predicted to be on the surface of the protein. It supports that the phosphorylation sites should be probably located on the surface of the substrate proteins to accessible for protein kinase. Obviously, our neural network system also consider this tertiary structure information, for more details in the following.

Table 2. Results *of* solvent accessibility predictions of phosphorylation sequences

Residue	Method	Exposed(E)	Buried(B)	Not predicted(N)
Serine	PHDAcc	70%	25%	5%
Threonine	PHDAcc	68%	25%	6%
Tyrosine	PHDAcc	66%	29%	6%

2.5 Neural Network System

Protein Test Set and Parameters. Experimentally verified phosphorylation sites were extracted mainly from Phospho.ELM[1], no sites were identical with a nine-peptide sequence. Negative examples of phosphorylation sites were assigned by the following approach: For each of the three acceptor types, a subset of the protein entries were categorized as being well characterized. All acceptor residues in the selected subsets, not reported as being phosphorylated, were assigned as negative sites. Five parameters are derived for each nine-peptide sequence (phosphorylated or non-phosphorylated) in all test protein dataset:

- Sequence-based Conservation. For each nine-peptide sequence in test dataset, we get the score embracing the conservation degree of phosphorylated by specific kinase, by using the MCL algorithm described in the above.
- Secondary Structure. nnpredict[18] was used to extract the secondary structure for each phosphorylated site in the nine-peptide sequence. The nnpredict classification was simplified to three categories: helix (H), sheet (E) or coil /other (C).
- Tertiary Structure. PHDAcc[20] was used to extract the tertiary structure information for each phosphorylated site in the nine-peptide sequence. The PHDAcc solvent accessibility prediction algorithm predicts each residue in the nine-peptide sequence to be in a buried (B) or exposed (E) state in a water solution.
- Residue Type. Phosphorylated site is simplified to three categories: S (Serine) Thnoeine (T), Tyrosine (Y).
- Protein Kinase Type. In this article, We only consider 10 protein kinases in which each contains greater than 100 phosphorylated sites.

Encoding. Sequence-based conservation score is encoded as a suitably scaled factor between 0 and 1 before presentation to the network, Other parameters: secondary structure, surface exposure information and residue type are encoded using 1-of-C encoding. Secondary structure is encoded by three input parameters: helix (H), sheet (E) and coil (C)/other. Tertiary structure is divided into two categories: buried (B) and Exposed (E). Residue type is encoded by three input parameters: S, T and Y. Protein Kinase type is encoded as an array of 10 inputs where one input is set to 1 and the rest to 0.

An example encoding is shown in Figure 1 for a nine-peptide sequence (LSGLSFKRN) with sequence-based conservation score 7, acceptor residue secondary structure (C), and exposed in protein surface (E), residue type (S). protein kinase type (IKK).

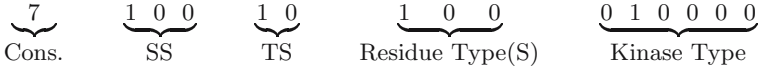


Fig. 1. Example of the neural network input encoding

Training the Neural Networking. The neural network used is FFNN[21], a feed forward neural networking trained using a scaled conjugate gradients algorithm[see Figure 2]. One Hidden-layer architecture is used in our case. In order to accurately measure the performance of the network it is trained using a ten fold cross validation experiment. The dataset is divided into ten equal subgroups, and the in each training run nine of the groups are used for training, while the network is tested on the single remaining group. The network is run ten times using a different subgroup as the test group each time. Training is for 100 epochs, in every case the network converged to a stable error-level before training is terminated.

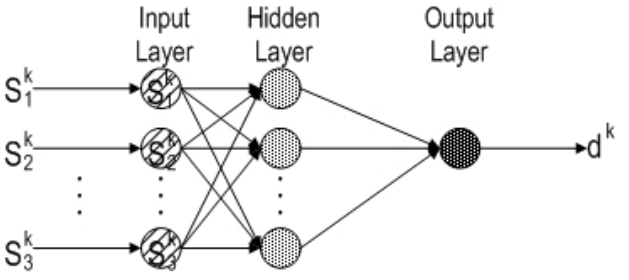


Fig. 2. FFNN:feed forward neural networking $F(x_1, x_2, ..., x_n) = f(\sum_{i=1}^{n} \omega_i(t)s_i^k - \theta)$ for each input pattern vector: $S^k = (s_1^k, s_2^k, ..., s_n^k)^T$,corresponding expected output: d^k, weight value: $\omega = (\omega_1, \omega_2, ..., \omega_n)^T$

3 Experimental Results

Performance on kinase PKA

In order to judge the neural networking learning process, a suitable measure of performance is required. In the following, we will mainly evaluate the performance of the three systems on kinase PKA.

True Positives TP = Number of correctly classified phosphorylation sites

True Negative TN = Number of correctly classified non-phosphorylation sites

False Positive FP = Number of non-phosphorylation sites incorrectly predicted to be phosphorylated (over-predictions)

False Negative FN = Number of phosphorylation sites incorrectly predicted to be non-phosphorylated (under-predictions)

Total Sites TS = Number of total sites $(TP + TN + FP + FN)$.

A measure of performance that takes these factors into account is the MCC. The formula for calculating MCC is shown in the following equation. The closer the value is to +1, the better the prediction system.

$$MCC = \frac{TP*TN - FP*FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

Obviously, you can see that our NNPhosPhoK method is better on performance than ScanSite2.0[2] and PhrePhospho[3] from Table 3 in the following:

Table 3. Comparision *of* MCC of phosphorylation site prediction for PKA among NNPhosPhoK, ScanSite 2.0, and PredPhospho

PKA	ScanSite2.0(stringency)			PredPhospho	NNPhosPhoK
	High	Medium	Low		
MCC	0.35	0.57	0.69	0.79	0.87

4 Discussion and Conclusion

Before our work, there are many methods focusing on in silico prediction of protein phosphorylation site, while most of them did not consider structure information. The NNPhosPhoK method acquire better performance than previous methods by considering structure information and using neural network technology. In our method, however, various prediction tools are not very accurate to predict the structure. On the other hand, the number of experimentally verified phosphorylated sites is also important for predicting the unknown protein. So far, the number of phosphorylation sites in Phospho.ELM[1] is very limited, and in our data, there are many phosphorylated sites which are not included in literatures. As increasing number of PDB with structure and experimentally verified phosphorylated sites, the NNPhosPhoK will show better ability of predicting.

Acknowledgement

This work is supported by The Key Project of The National Nature Science Foundation of China under the grant No. 60533020. We would like to thank all members in our bioinformatic group for giving many good advices.

References

1. Diella, etal.: Phospho.ELM: A Database of Experimentally Verified Phosphorylation Sites in Eukaryotic Proteins. BMC Bioinformatics 5 (2004) 79
2. Obenauer, J.C., Cantley, L.C., Yaffe, M.B.: Scansite 2.0: Proteomewide Prediction of Cell Signaling Interactions Using Short Sequence Motifs. Nucl. Acids. Res. (31) (2003) 3635-3641
3. Kim, J.H., Lee, J., Oh, B., Kimm, K., Koh, I.: Prediction of Phosphorylation Sites Using SVMs. Bioinformatics (2004)
4. Manning, G., Whyte, D.B., Martinez, R., Hunter, T., Sudarsanam, S.: The protein kinase complement of the human genome. Science 298(5600) (2002) 1912-1934
5. Kolibaba, K., Druker, B.: Protein Tyrosine Kinases And Cancer. Biochim. Biophys. Acta. 1333 (1997) 217-248
6. Hunter, T.: The Phosphorylation of Proteins on Tyrosine: Its Role in Cell Growth And Disease. Philos. Trans. R. Soc. Lond. B. 353 (1998) 583-605
7. Johnson, L., Noble, M., Owen, D.: Active And Inactive Protein Kinases: Structural Basis for Regulation. Cell 85 (1996) 149-158
8. Johnson, L., Lowe, E., Noble, M., Owen, D.: The Structural Basis for Substrate Recognition and Control by Protein Kinases. FEBS Lett. 430 (1998) 1-11
9. Pinna,L. A., Ruzzene, M.: How Do Protein Kinases Recognize Their Substrates?. Biochim. Biophys. Acta. 1314 (1996) 191-225
10. Kraft, C., Herzog, F., Gieffers, C., Mechtler, K., Hagting, A., Pines, J., Peters, J.M.: Mititic Regulation of The Human Anaphase-Promoting Complex by Phosphorylation. EMBO J. 22 (2003) 6598-6609
11. Rychlewski, L., Kschischo, M., Dong, L., Schutkowski, M., Reimerm, U.: Targer Specificity Analusis of The Abl Kinase Using Peptide Microarray Data. J. Mol. Biol. 336 (2004) 307-311

12. Knight, Z.A., Schilling, B., Row, R.H., Kenski, D.M., Gibson, B.W., Shokat, K.M.: Phosphospecific Proteolysis for Mapping Sites of Protein Phosphorylation. Nat. Biotechnol. 21 (2003) 1047-1054
13. Songyang, Z., Blechner, S., Hoagland, N., Hoekstra, M. F., et al.: Use of An Oriented Peptide Library to Determine The Optimal Substrates of Protein Kinases. Curr. Biol. 4 (1994) 973-982
14. Henikoff, S., Henikoff, J.G.: Amino Acid Substitution Matrices from Protein Blocks. Proc. Natl. Acad. Sci. USA. 89 (1992) 10915-10919
15. Enright, A.J., Van, Dongen, S., Ouzounis, C.A.: An Efficient Algorithm for Large-Scale Detection of Protein Families. Nucleic Acids Research 30(7) (2002) 1575-1584
16. Andres Kregipuu, Nikolaj Blom, Soren Brunak, Jaak Jarv: Statistical Analysis of Protein Kinase Specificity Determinants. FEBS Letters. 430 (1998) 45-50
17. Rost, B., Sander, C.: Combining Evolutionary Information and Neural Networks to Predict Protein Secondary Structure. Proteins 19 (1994) 55-72
18. Kneller, D.G., Cohen, F.E., Langridge, R.: Improvemnets in Protein Secondary Structure Prediction by and Enhanced Neural Network. J. Mol. Biol. 214 (1990) 171-182
19. Frishman, D., Argos, P.: Incorporation of Non-local Interactions in Protein Secondary Structure Prediction from The Amino Acid Sequence. Proteins Eng. 9 (1996) 133-142
20. Rost, B., Sander, C.: Conservation and Prediction of Solvent Accessibility in Protein Families. Proteins 20 (1994) 216-226
21. Gustafsson, C., Reid, R., Greene, P.J., Santi, D.V.: Identification of New Modifying Enzymes by Iterative Genome Search Using Known Modifying Enzymes as Probes. Nucl. Acids. Res. 24 (1996) 3756-3762

Neural Feature Association Rule Mining for Protein Interaction Prediction

Jae-Hong Eom

Biointelligence Laboratory, School of Computer Science and Engineering,
Seoul National University, Seoul 151-744, South Korea
jheom@bi.snu.ac.kr

Abstract. The prediction of protein interactions is an important problem in post–genomic biology. In this paper, we present an association rule mining method for protein interaction prediction. A neural network is used to cluster protein interaction data and a feature selection is used to reduce the dimension of protein features. For model training, the preliminary network model was constructed with existing protein interaction data in terms of their functional categories and interactions. A set of association rules for protein interaction prediction are derived by decoding a set of learned weights of trained neural network after this model training. The protein interaction data of *Yeast* from public databases are used. The prediction performance was compared with simple association rule-based approach. According to the experimental results, proposed method achieved about 95.5% accuracy.

1 Introduction

Many challenges have tried to predict protein interactions and functions with different data such as protein–protein interaction (PPI) data, microarray gene expression data, etc. For example, clustering-based gene expression data analysis also adopted for un-annotated protein function prediction based on the idea that genes with similar functions are about to be co-expressed [1]. A protein interaction analysis approach between protein domains in terms of the interactions involving structural families of evolutionarily related domains was proposed by Park *et al*. [2]. Iossifov *et al*. [3] and Ng *et al*. [4] inferred new interaction with previously discovered interaction data.

Here, we propose an adaptive neural network (ANN) based feature association mining method for PPI prediction. We used additional association rules for PPI prediction. These are generated by decoding a set of learned weights of adaptive neural network. We assumed that these association rules decoded from neural network (NN) would make the whole prediction procedure more robust to unexpected error factors by accounting relatively robust characteristic of NNs.

Basically, we use ART-1 version of adaptive resonance theory [5] as an ANN clustering model to construct prediction model. The ART-1 [6] is a modified version of ART [7] for clustering binary vectors. Here, we assume again PPI of yeast as '*feature–to–feature*' associations of each interacting proteins. Basically, we used the same

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 690 – 695, 2006.

approaches of Rangarajan *et al*. [8] for designing clustering model. Also, the same feature selection filter of Yu *et al*. [9] was used to reduce computational complexity by feature dimension reduction.

This paper is organized as follows. In Section 2, we describe feature selection filter and overall architecture of ART-1 based protein interaction clustering model. Next, we present detailed NN training method with PPI data and the decoding method of association rule in Section 3. In Section 4, we present the representation scheme of PPI for the NN input, association mining method, and experimental results. Finally, concluding remarks and future works are drawn in Section 5.

2 Feature Dimension Reduction and Protein Cluster Learning

Feature Dimension Reduction by Feature Selection

In this paper, each PPI was considered as associations among set of features. We constructed massive feature sets for each protein and interacting protein pairs from public protein databases in the previous work of Eom *et al*. [10]. But, there are also many features which have no information of its association with other proteins. Therefore, feature selection may be needed in advance of clustering PPIs. Especially, this feature selection is necessary when dealing with such high dimensional data. So, to filter out these features we used entropy and information gain based measure, *symmetrical uncertainty*, as a measure of feature correlation and which is defined in the work of Press *et al*. [11]. The overall filtering procedures are described in the paper of Eom *et al*. [10]

Clustering of Protein Interactions with NN

We use ART-1 NN to group the class of PPIs by their 13 functional classes and the class of interacting counter parts. In our ART-1 based clustering, each PPI is represented by a prototype vector that is generalized representation of the set of features of each interacting proteins. The degree of similarity between the members of each cluster can be controlled by changing the vigilance parameter ρ of Eom *et al*. [12]. The more detailed overall procedures for clustering PPIs with the ART-1 based clustering model is described in our previous work by Eom *et al*. [12]. The set of weights of trained NN were decoded as the form of association rule with the weight-to-rule decoding procedures described in Eom *et al*. [13] to enrich the protein features.

3 Rule Extraction from Trained Neural Network

Learning Feature Association with Neural Network

A supervised ANN uses a set of training examples or records include N attributes. Each attribute, A_n (n = 1, 2, … , N), can be encoded into a fixed length binary substring $\{x_1 \ldots x_i \ldots x_{m(n)}\}$, where $m(n)$ is the number of possible values for an attribute A_n. The element x_i = 1 if its corresponding attribute value exists and 0 otherwise. So, the proposed number of input nodes, $\boldsymbol{I}$, in the input layer of ANN can be given by

$$I = \sum_{n=1}^{N} m(n) \tag{1}$$

The input attributes vectors, $\boldsymbol{X}_m$, to the input layer can be rewritten as $\boldsymbol{X}_m = \{x_1 \ldots x_i \ldots x_I\}_m$, $m = (1,2,\ldots, M)$. The M is the total number of input training patterns. The output class vector, $\boldsymbol{C}_k (k = 1, 2, \ldots, K)$, can be encoded as a bit vector of a fixed length K as $C_k\{\psi_1 \ldots \psi_k \ldots \psi_K\}$. Here, K is the number of different possible classes. If the output vector belongs to class_k then the element ψ_k is equal to 1 and 0 otherwise. Therefore, the proposed number of output nodes in the output layer of ANN is K. Then, the input and the output nodes of the ANN are determined and the structure of the ANN. The ANN is trained on the encoded vectors of the input attributes and the corresponding vectors of the output classes. The training of ANN is processed until the convergence rate between the actual and the desired output will be achieved.

After training the ANN, two groups of weights can be obtained. The first group, $(\text{WG1})_{i,j}$, includes the weights between the input node i and the hidden node j. The second group, $(\text{WG2})_{j,k}$, includes the weights between the hidden node j and the output node k. A sigmoid is used for the activation function of the hidden and output nodes of the ANN. The total input to the j–th hidden node, IHN_j and the output of the j–th hidden node, OHN_j, and the total input to the k–th output node, ION_k, are

$$IHN_j = \sum_{i=1}^{I} x_i (WG1)_{i,j}, \; OHN_j = \frac{1}{1+e^{-\left[\sum_{i=1}^{I} x_i (WG1)_{i,j}\right]}}, \; ION_k = \sum_{j=1}^{J} (WG2)_{j,k} \frac{1}{1+e^{-\left[\sum_{i=1}^{I} x_i (WG1)_{i,j}\right]}} \tag{2}$$

So, the final value of the k–th output node, ψ_k, is given by

$$\psi_k = \left\{ \frac{1}{1+e^{-\left[\sum_{j=1}^{J} WG2_{j,k} \left(\frac{1}{1+e^{-\left[\sum_{i=1}^{I} x_i (WG1)_{i,j}\right]}} \right)\right]}} \right\} \tag{3}$$

The function, $\psi_k = f(x_i, (\text{WG1})_{i,j}, (\text{WG2})_{j,k})$ is an exponential function in x_i since $(\text{WG1})_{i,j}$, $(\text{WG2})_{j,k}$ are constants. Its maximum output value is equal to one.

Association Rule Construction from Trained Neural Network with GA

To extract relations (rules) among the input attributes, $\boldsymbol{X_m}$ relating to a specific $\boldsymbol{class_k}$ one must find the input vector, which maximizes $\boldsymbol{\psi_k}$. This is an optimization problem and can be stated as $\psi_k(x_i)$ by considering binary data feature vector $\boldsymbol{x}$. In $\psi_k(x_i)$, x_i are binary values (0 or 1).

Since the objective function $\psi_k(x_i)$ is nonlinear and the constraints are binary so, it is a nonlinear integer optimization problem. Thus the genetic algorithm (GA) can be used to solve this optimization problem by maximizing the objective function $\psi_k(x_i)$. In this paper, we used conventional generational-GA procedures with this objective function $\psi_k(x_i)$ to find the best chromosome which provided as an input of NN and produce best network output (highest interaction prediction accuracy).

After we obtain the best chromosomes which produces best network output, we decode these chromosome into the form of association rule (we call this association rule as '*neural feature association rule*', since it's extracted from trained NN). To extract a rule for $\boldsymbol{class_k}$ from the best chromosome selected by GA procedures, we decoded it with several procedures presented in our technical report [13]. The basic approach of above procedures and notations are borrowed from the work of Elalfi *et al.* [14].

4 Experimental Results

Protein Interaction as Binary Feature Vector
An interaction is represented as a pair of two proteins that directly binds to each other. This protein interaction is represented by binary feature vector of interacting proteins and their associations. These interaction representation processes and the processing steps are described in the work of Eom *et al*. [10].

Data Sets
Each *Yeast* proteins have various functions or characteristics which are called 'feature.' Here, set of features of each protein are collected from public genome databases as the same manner of Eom *et al*. [10]. Table 1 shows the statistics of each interaction data source and the number of features before and after the feature filtering.

Table 1. The statistics for the dataset

Data Source	# of interactions	# of initial features	# of filtered features
MIPS	10,623	6,014 (total)	1,108 (total)
YPD	2,940		
SGD	1,462		
Y2H (Ito *et al*.)	955		
Y2H (Uetz *et al*.)	5,072		

Table 2. Accuracy of the proposed methods. The effect of the FDRF-based feature selection and NN-based are shown in terms of prediction accuracy.

Prediction method	Number of interactions			Accuracy (|P|/|T|)
	Training set Size	Test set (T)	Predicted correctly (P)	
Asc. (△)	4,230	420	382	90.9 %
FDRF + Asc. (▽)	4,230	420	392	93.3 %
Asc. + N-Asc. (◇)	4,230	420	389	92.6 %
FDRF + Asc. + N-Asc. (☆)	4,230	420	401	95.5 %

Experiment Procedures
First, we predicted the classes of new PPIs with NN for their 13 functional categories obtained from MIPS [11]. The accuracy of class prediction is measured whether the predicted class of interaction is correctly corresponds to the class of MIPS. After this, we constructed feature association rule from this trained NN.

Next, we trained another NN with PPI data represented as binary feature vector according to the method of Eom *et al*. [10]. After the model training, we extracted again feature association rule from the model with the procedure of Eom *et al*. [13]. Then we predicted test PPI set with these two set of association rules and measured the prediction accuracy of each approaches. Results are measured with 10-fold cross-validation.

Results

Table 2 show the interaction prediction performance of various combination of associantion mining, feature filtering, and exploitation of rules derived from NN.

In Table 2, Simple association mining approach (△) achieved the lowest performance. The number of total feature used in this approach was 6,014. This is quite high feature dimension. We can guess that it may includes lots of non-informative and redundant features and these features may affect the prediction accuracy in negative way by interfering correct rule mining. This assumption confirmed by investigating the result of second approach, FDRF + Asc. (▽), association mining with non-informative and redundant feature filtering. This feature filtering approach improved overall prediction performance about 2.4% than the first approach. But the third approach, Asc. + N-Asc. (◇), prediction with the rules from association rule mining and the rule derived from trained NN only improved overall prediction performance about 1.7% than the first approach. This result can be explained again with the feature dimension problem. In this third approach, there also exist redundant and non-informative garbage features which decrease the prediction performance. But in this approach, eventhough there still lots of garbage features, the over all performance improved about 1.7%. This is the effect of the rule exploitation derived from trained NN. This inference can be confirmed again by investigating the result of fourth approach, FDRF + Asc. + N-Asc (☆), prediction with the rule from association mining and the rule derived from trained NN along with feature filtering. Non-informative and redundant features are filtered out in this approach. Consequently, his approach improved over all prediction accuracy up to 4.6%.

Thus, we can say that both the information theory based feature filtering and the exploitation of the rule derived from trained NN and conventional association rule mining methods are helpful for improving overall performance of feature-to-feature association-based PPI prediction. By considering these experimental results, the proposed approaches in this paper will be useful as a data preprocessing and prediction methods especially when we handle the data which have many features.

5 Conclusions

In this paper, we presented NN based protein interaction learning and association rule mining method from feature set and trained NN model for PPI prediction task. The proposed method (combination of all methods) achieved the improvement of accuracy about 4.6%. The experimental results of various approaches suggest that the NN based feature association learning model could be used for more detailed investigation of the PPIs when the proposed model can learn effectively the hidden patterns of the data which have many features and implicit association of these features. From the result of Section 4, we can conclude that the proposed method is suitable for efficient analysis of PPIs through its hidden feature association learning.

However, current public interaction data have many false positives and some interactions of these false positives are corrected as true positives by recent researches through reinvestigation with new experimental approaches. Thus, the study on the new method for adapting these changes in data set which is related to false positive

screening remains as future works. Also, consideration of more biological features such as pseudo amino acid composition or protein localization facts will be helpful for improving overall performance.

References

1. Eisen, M.B., et al.: Cluster Analysis and Display of Genome–Wide Expression Patterns. Proc. Natl. Acad. Sci. USA **95** (1998) 14863–68
2. Park, J., et al.: Mapping Protein Family Interactions: Intramolecular and Intermolecular Protein Family Interaction Repertoires in the PDB and Yeast. J. Mol. Biol. **307** (2001) 929–39
3. Iossifov, I., et al. Probabilistic Inference of Molecular Networks from Noisy Data Sources. Bioinformatics **20**(8) (2004) 1205–13
4. Ng, S.K., et al. Integrative Approach for Computationally Inferring Protein Domain Interactions. Bioinformatics **19**(8) (2003) 923–29
5. Carpenter, G.A. and Grossberg, S.: A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine. Computer Vision, Graphics and Image Processing **37** (1987) 54–115
6. Barbara, M.: ART1 and pattern clustering. In: Proceedings of the Summer School of Connectionist Models. Morgan Kaufmann (1988) 174–185
7. Heins, L.G. and Tauritz, D.R.: Adaptive Resonance Theory (ART): An Introduction. Internal Report 95-35, Dept. of Computer Science, Leiden University, Netherlands, (1995) 174–85
8. Rangarajan, et al.: Adaptive Neural Network Clustering of Web Users. IEEE Computer **37**(4) (2004) 34–40
9. Yu, L. and Liu, H.: Feature Selection for High Dimensional Data: A Fast Correlation-based Filter Solution. In: Proceeding of ICML-03 (2003) 856–863
10. Eom, J.-H., et al.: Prediction of Implicit Protein–Protein Interaction by Optimal Associative Feature Mining. Lecture Notes in Computer Science (IDEAL'04) **3177**, (2004) 85–91
11. Press, W.H., et al.: Numerical Recipes in C. Cambridge University Press. (1988)
12. Eom, J.-H., et al.: Adaptive Neural Network Based Clustering of Yeast Protein-Protein Interactions. Lecture Notes in Computer Science (CIT'04) **3356**, (2004) 49–57
13. Eom, J.-H.: Prediction of Yeast Protein–Protein Interactions by Neural Association Rule. Internal Report 04-03, Dept. of Computer Sci.&Eng., Seoul National University, Republic of Korea, (2004) 1–12
14. Elalfi, A.E., et al.: Extracting Rules from Trained Neural Network Using GA for Managing E-business. Applied Soft Computing **4**, (2004) 65–77

Prediction of Contact Maps Using Modified Transiently Chaotic Neural Network

Guixia Liu, Yuanxian Zhu, Wengang Zhou,
Chunguang Zhou, and Rongxing Wang

College of Computer Science and Technology,
Jilin University,
Changchun 130012, P.R. China
lgx1034@163.com, cgzhou@jlu.edu.cn

Abstract. Contact maps are considered one of the most useful strategic steps in protein folding recognition. And there are a variety of measures of residues contact used in the literature. In this paper, we address our question on using a transiently chaotic neural network to predict the contact maps and whether it is reasonable. Our results show that it is more successful that we predict proteins contact maps based on modified transiently chaotic neural network.

1 Introduction

Prediction of the three dimensional structure of a protein from its amino acids sequence is one of the most challenging problems in computational molecular biology. It is at present intractable to find the structures only use molecular modeling, and the accuracy of the methods which predict the three-dimensional structures directly from the amino acids sequences is not high enough, so intermediate steps, such as residue contacts prediction , and residue spatial distance prediction, were put forward and have been developed rapidly recently. Contacts between protein residues constrain protein folding and characterize different protein structures. Therefore its solution may be very useful in protein folding recognition and de novo design. It is much easier to get the major features of the three-dimensional (3D) structure of a protein if the residue contacts are known for the protein sequence, and methods that reconstruct the protein structure from its contact map have been developed [2] (see Fig. 1). A similarity based on contact map overlaps is the only approach for structural comparison that does not require a pre-calculated set of residues equivalences as one of the goals of the method.

There are a variety of measures of residues contact used in the literature. Some use the distance between the C_α-C_α atoms [10], while others prefer to use the distance between the C_β-C_β [2].Contact maps are two dimensional, binary representations of protein structures. For a protein with N residues, the contact map for each pair of amino acids k and l ($1 \le k, l \le N$), will have a value C(k,l)=1, if a suitably defined distance $d(k,l) < d_{thr}$ where d_{thr} is a user-defined threshold distance between the amino acids, and C(k,l)=0 otherwise. We consider two residues to be in contact if the

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 696 – 701, 2006.

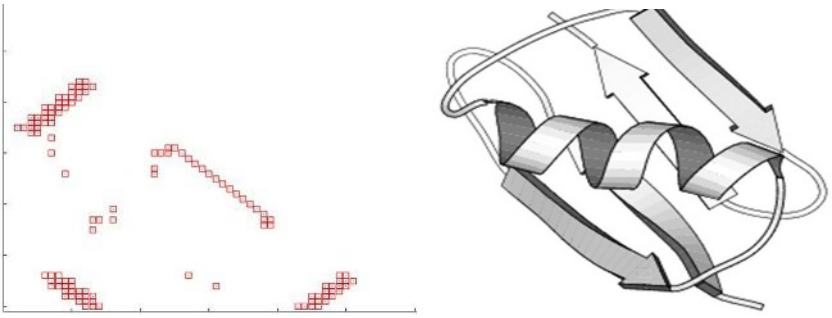


Fig. 1. Contact map and 3D structure of PDB protein 2igd. The figure on right shows the ribbon representation of secondary structure in the protein 2igd.On the left is the contact map for this protein.

distance between their C_α atoms is less than 8.5Å while their sequence separation is not smaller than 4.

Previous contact map prediction methods have used neural networks [1-2], correlated mutations [2], association rules [7] etc. Our earlier work led us to believe that neural networks are efficient tools in solving the contact map prediction. Recurrent neural network [3] with bias units has been used in our pioneer work and has gotten a satisfactory result. In this work, we address the question how to apply chaotic neural network and whether a chaotic neural network system is capable of learning the correlation between the residue covalent structure of a protein and its contact map as it can be computed from its known 3D structure.

2 Background Material

2.1 Database

In this work, we use a large set of non-homologous proteins of solved 3D structure. The set which consists of proteins with an identity value <25%is extracted from the Protein Data Bank (PDB) using PDB_select list [6] of October 2004, it is the most recent 25% threshold list and containing 2485 proteins. This set is first reduced by excluding those proteins which has non-standard amino acid residues, and then the set is further reduced by removing those proteins whose backbone is interrupted. We use the DSSP program [7] on all the PDB files to extract 3D coordinates and to assign secondary structures, then we remove also sequences on which DSSP crashes. Finally the set contains 2095 proteins. Furthermore, 125 proteins randomly selected are given in Table 1. The sub-dataset is divided into 5 classes: $L<100$,$100\leq L\leq 199$,$200\leq L\leq 299$, $300\leq L\leq 399$, $L\geq 400$, according to their residue length. The contacts between residues which are less than five residues separation are not included while we train or test the networks.

Table 1. The database of proteins used to train and test the modified transiently chaotic NN

L<100	100≤-L≤-199	200≤L≤299	300≤ L≤399	L>400
1CEUA	1BEA_	1JXIA	1QMLA	1PMOC
1YUJA	1XD7A	1G8AA	1IXH_	1QKIB
1BXYA	1MWWB	1V8AA	1MUWA	10T5A
1H4JB	1MAI_	1QHHB	1W3BA	1LKXD
1EKTA	1PSRA	1O3SA	1ST4A	1KP0A
1MHMB	1L1PA	1EL6A	1T5JA	1KS8A
1KIKA	2PVBA	1HAVA	1FZEF	1PMOC
1URQA	1I4VA	1FUJA	1UF4A	1FSU_
11OAIA	1LUQB	1BU2A	1B3OA	1H6VC
1JNIA	1N32I	1SE1A	1SIG_	1GLLO
1PCFA	1UUZB	1XO1A	1BOB_	1N4WA
1IGL_	1H31B	1P1XA	1EFPA	1QATA
1OS6A	1HDKA	1OL1B	1D3VA	1M1NB
1PL5A	1K3KA	1JKUA	1T2DA	1AYL_
1C01A	1VKBA	1OJRA	1JIKA	1UWKA
1E44A	1LM8V	1R52B	1I9YA	4AAHA
1WKT_	1V2BB	1HM7A	1R5YA	1M6BB
1CY5A	1BYSA	1I78B	1NOYA	1EQRA
1FBR_	1N1FA	1NH1A	3SIL_	1WD9A
1TTG_	1I12A	1O58A	1MUWA	1SLY_
1IM3D	1SK3A	1A3GA	1T9GC	1SU8A
1PA4A	1QOLA	1SM4A	1QOPB	1BJT_
1HS7A	1KD6A	1GC1G	1I24A	1H2WA
1LRIA	1PVMB	1U4GA	1QHDA	1JB0B
2EZL_	1M4UA	1GLV_	1JFBA	1HE8A

Protein length (L) is the residue number of the covalent structure.

2.2 Features

In this work, we capture two features of the amino acids: predicted secondary structure and hydrophobicity. The predicted secondary structures for each protein are obtained by using PSIPRED, we use 6 neurons to denote the 6 possible secondary structure pairs since a amino acid residue has three possible secondary structures: α-helix, β-sheet and coil. Hydrophobicity is a measure of nonpolarity of the side chains. As the nonpolarity (hydrophobicity) of the side chain increases, it avoids being in contact with water and buried within the protein nonpolar core. This is seen as the essential driving force in protein folding. This quantity is used to encode residue specific information to the network. Since the hydrophobicity of a residue affects the non-covalent bonding between its surroundings, it can be a contributing factor to contact decision of that residue with others. Here we use ROSEF hydrophobicity scale since it is one of frequently used scale.

3 Modified Transiently Chaotic NN and Input Encoding

3.1 Network

In this work, we use a modified transiently chaotic network to deal with prediction of contact maps in proteins. The architecture of the network is depicted in Figure 2. This topology consists of three layers of neurons: one output neuron representing the contact propensity, one hidden layer containing ten neurons and one input layer with different number of neurons depending on the amount of information encoded, we use 1050 neurons for 5 residue pairwise, 10 neurons for residue classification according to hydrophobicity, polar, acidic and basic, 6 neurons for secondary structure information.

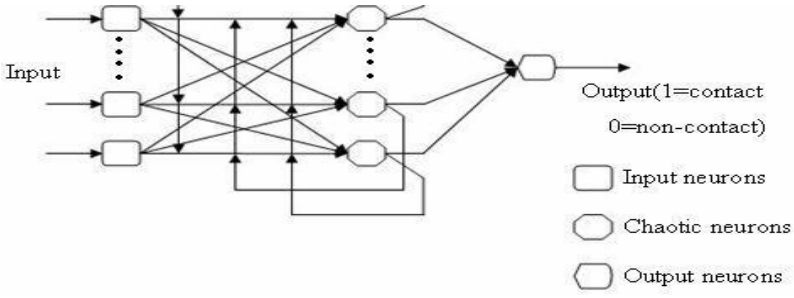


Fig. 2. Neural network structure implemented in this paper

Transiently chaotic neural network (TCNN) model is defined as follows:

$$x_i(t) = f(y_i(t)) = \frac{1}{1 + e^{-y_i(t)/\varepsilon}} \tag{1}$$

$$y_i(t+1) = ky_i(t) + \alpha(\sum_{j=1, j\neq i}^{n} w_{ij}x_j + I_i) - z_i(t)(x_i(t) - I_0) \tag{2}$$

$$z_i(t+1) = (1-\beta)z_i(t) \tag{3}$$

where

xi = output of neuron i.
yi = internal state of neuron i.
wij = connection weight from neuron j to neuron i.
Ii = input bias of neuron i.
α = positive scaling parameter for inputs.
k = damping factor of nerve membrane ($0 \leq k \leq 1$).
zi(t) = self-feedback connection weight or refractory strength .
ε = steepness parameter of the output function (ε>0)
β =damping factor of the time-dependent zi(t).($0 \leq \beta \leq 1$)
I_0 =positive parameter.

The variable zi(t) corresponds to the temperature in the usual stochastic annealing process in this network architecture. So Eq.3 is an exponential cooling schedule for the annealing. The network will change into hopfield network when the value of zi(t) becomes small enough. In this work, we update three time-independent variables(α(t), β(t), ε(t)) into TCNN which are all constants(α, β, ε) in the original model.

$$\alpha(t+1) = (1+\lambda)\alpha(t), \quad \text{if } \alpha(t)<0.1 \quad \text{else } \alpha(t)=0.08 \tag{4}$$

$$\beta(t+1) = (1+\varphi)\beta(t), \quad \text{if } \beta(t)<0.2 \quad \text{else } \beta(t)=0.1 \tag{5}$$

$$\varepsilon(t+1) = (1-\eta)\varepsilon(t), \quad \text{if } \varepsilon(t)>0.001 \quad \text{else } \varepsilon(t)=0.00 \tag{6}$$

Where λ, φ and η are small positive constants.

At the beginning, α(0) and β(0) are set to small values. When α(t) is small, the influence on the energy function is not strong enough to generate transient chaos. When β(t) is a small value, it can make z(t) decrease slowly at the beginning so that the network has enough time to keep the self-feedback large enough to generate chaos. When β(t) becomes large, z(t) decreases quickly. At the same time the influence of the energy function becomes strong when α(t) becomes large gradually. This means the self-feedback signal becomes weak in the motion equation. At the beginning, ε(t) is set to a large value. This means that the steepness of the output function is small. This can help the neural network generate chaos easily. However once ε(t) becomes small, the steepness of the output function becomes large .With this method, the modified transiently chaotic neural network has relatively rich dynamics at the beginning, and converges to stable states faster after bifurcation for all neurons.

4 Results and Discussions

We are actually interested in the network capability of predicting residue contacts. Therefore accuracy *(*A) of the network is defined as the ratio of the correctly predicted contacts by the network to the actual number of contacts in a protein and calculated according to:

$$A=N^{*}_{C}/N_{C.} \tag{7}$$

Where N_c^* is the number of correctly predicted contacting residues by the network; Nc is the actual number of contact within the protein.

A random predictor makes Nc number of guess in order to predict the contacting pairs, assuming that there are Np number of residue pairs in which Nc of them are contacting. Therefore, its performance *(*Ar) is calculated by the following formula:

$$A_r=N_c/N_{\mathrm{p}} \tag{8}$$

Where N_P=(L-4)(L-3)/2,L is the protein length.

In order to calculate the improvement over a random predictor, accuracy A of the network is divided to performance of the random predictor A. Improvement over a random predictor is denoted by R and calculated according to the formula:

$$R=A/Ar \tag{9}$$

The performance of the network is tested on five data sets. All performance results are shown in table 2.

Table 2. Accuracy of predicted contact maps

RESIDUE LENGTH	A	R
L<100	0.125	3.54
100≤L≤199	0.091	3.71
200≤L≤299	0.078	4.78
300≤L≤399	0.070	5.23
L>400	0.067	7.95
ALL PROTEINS	0.082	5.83

5 Conclusion

In this paper, we present a Modified transiently chaotic neural network for predicting contact maps within 125 proteins, and it is clearly demonstrated that our network performs with a better efficiency than our previous studies. Chaotic neural networks have shown to be a powerful tool for this problem and have a nice prospect in the application of protein folding recognition. In the future, we will predict contact maps in non-homologous proteins with novel methods and make more research on protein folding recognition and structure comparison.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant No. 60433020, the Key Laboratory of Symbol Computation and Knowledge Engineering of the Ministry of Education, and "985" project of Jilin University.

References

1. Fariselli, P., Casadio, R.: Neural Network Based Predictor of Residue Contacts in Proteins. Protein Engineering 12 (1999) 15–21
2. Fariselli, P., Olmea, O., Valencia, A., Cassadio R.: Prediction of Contact Maps with Neural Networks and Correlated Mutations. Protein Eng. 14 (2001) 835-843
3. Crippen, G.M., Havel, T.F.: Distance Geometry and Molecular Conformation. Chemometrics Series, 15. New York: John Wiley & Sons (1988)
4. Aszodi, A., Munro, R.E., Taylor, W.R.: Distance Geometry Based Comparative Modelling. Fold Des 2 (1997) S3–S6
5. Yin, F., Wang, J. and Guo, C. (Eds.): ISNN 2004, LNCS Vol. 3173.Springer-Verlag Berlin Heidelberg New York (2004) 401–405
6. Hobohm, U., Scharf, M., Schneider, R., Sander, C.: Selection of Representative Data Sets. Prot. Sci. 1 (1992) 409-417
7. Kabsch, W., Sander, C.: Dictionary of Protein Secondary Structure: Pattern Recognition of Hydrogen-Bonded and Geometrical Features. Biopolymers 22 (1983) 2577-2637
8. Rose, G.D., Geselowitz, A.R., Lesser, G.J., Lee, R.H., Zehfus, M.H.: Hydrophobicity of Amino Acid Residue in Globular protein. Science 229 (1985) 834-838
9. Vendruscolo, M., Kussell, E., Domany, E.: Recover of Protein Structure from Contact Maps. Structure Fold. Des. 2 (1997) 295-306
10. Mirny, L., Domany, E.: Protein Fold Recognition and Dynamics in the Space of Contact Maps. Proteins 26 (1996) 319-410

Identification of Cell-Cycle Phases Using Neural Network and Steerable Filter Features

Xiaodong Yang[1], Houqiang Li[1], Xiaobo Zhou[2], and Stephen T.C. Wong[2]

[1] Department of Electronic Engineering and Information Science, University of Science and Technology of China
xdyang@mail.ustc.edu.cn, lihq@ustc.edu.cn
[2] Center for Bioinformatics, HCNR, Harvard Medical School, Boston, MA 02115, USA
zhou@crystal.harvard.edu, stephen_wong@hms.harvard.edu

Abstract. In this paper, we aim to address the cell phase identification problem, and two important aspects, the feature extraction methods and the classifier design, are discussed. In our study, we first propose extracting high frequency information of different orientations using Steerable filters. Next, we employ a multi-layer neural network using the back-propagation algorithm to replace K-Nearest Neighbor (KNN) classifier which has been implemented in the Cellular Image Quantitator (CELLIQ) system [3]. Experimental results provide a comparison between the proposed steerable filter features and existing regular features which have been used in published papers [3, 5]. From the comparison, it can be concluded that Steerable filter features can effectively represent the cells in different phases and improve the classification accuracy. Neural network also has a better performance than KNN currently deployed in CELLIQ system [3].

1 Introduction

Understanding cancer cells' cycle progression is important in understanding drug effects on cancer cells. Automatic techniques to analyze cell cycle progression are of considerable interest since recent developments of time-lapse microscopy make large volumes of image data available. To measure the cell progression as a function of time, it is necessary to segment and track the cells in successive frames. After a successful track of the detected cells, classifier is applied to identify the cell phases. Combining the segmentation, tracking, and classification results, it is possible to obtain the average duration in each phase, which is useful for doctors to judge the drug effects on cancer cells. Segmentation and tracking of cells in microscopy have been finished in our previous works [1, 2, and 3]. We focus on the phase identify-cation problem in this paper.

Cell classification, also known as cell cycle phase identification [3], is to classify each cell into one of the four phases: inter phase, prophase, metaphase, and anaphase. Among many existing classifiers [4], we choose the simple and flexible K-Nearest

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 702 – 709, 2006.

Neighbor and the neural network. As for feature extraction [3, 5, and 6], we propose a new set of steerable filter features based on steering theories [7]. Unlike Gabor filters [8], this method does not need many filters to get arbitrary orientation information, and more importantly, the proposed steerable filter features are rotation-invariant. The effectiveness of the new steerable filter features is proved by experiments presented in Section 5. Adoption of the new steerable filter features can significantly improve the classification accuracy within the framework of KNN and neural network.

It should be pointed out that the presented algorithm is an extension of the algorithm in [3]. In the previous work, Chen et al. employed KNN and some regular features in the CELLIQ system [3]. In order to improve the classification accuracy, we replace KNN with neural network and introduce a new set of steerable filter features.

The rest of this paper is organized as follows. In section 2, we briefly review segmentation and tracking of migration cells, and our methods to solve this problem in previous work. In section 3, regular features and the new steerable filter features are introduced. Section 4 explains phase identification problem using the proposed steerable filter features and two popular classifiers. Section 5 gives the experimental results. Section 6 concludes this paper.

2 Cell Segmentation and Tracking

The cell image analysis system consists of three parts: segmentation, tracking and classification. Segmentation step extracts every cell in each frame. Tracking step finds the cell trajectories and cell associations in consecutive frames and every cell is tracked from the first frame to the last frame. Phase identification is then applied to the detected cells. Thus, drug effects on cancer cells can be estimated based on the classification and tracking results.

In [2], we use the classical threshold method to extract cells in the video sequences. To separate the touching cells, a new method based on marker-controlled watershed is presented. We do not employ UEPs as cell seeds, which often cause much over-segmentation. Instead, to extract cell markers, we proposed a group of cell-like masks to perform erosion, which has been proven to be both robust and effective. In order to improve segmentation accuracy, we use context information of video sequence. By extracting markers from the former frame or the next frame, we can reduce some over-segmentation and under-segmentation.

Our tracking algorithm [2] is based on mean shift and Kalman filter. Kalman filter is used to predict the position in the next frame, which can make the tracking algorithm more robust. To make better use of shape and intensity information of cell nuclei, we modify classical Gaussian kernel mean shift. Two new kernels with scale, shape, and direction adaptation have been proposed for this aim due to the fact that, cell nuclei have ellipse shape in different directions and have nearly uniform intensity within the cell region. Mean shift algorithm can shift the position predicted from the former frame using Kalman filters to the target position which has largest kernel density in the current frame. Because mean shift tracking is the so-called one-by-one tracking, in order to detect cell division, we track the cells from the last frame to the first frame.

3 Feature Extraction

Many regular features have been used or proposed in previous classifications [3, 5, and 6]. The first set of features is extracted directly from the image intensity. Threshold technique is first used to detect the object region, and then many effective features based on pixel intensity are extracted from the detected object, such as average intensity, standard deviation, object area, elongation and compactness [3].The second set of features is moment features, including invariant moments [6] and Zernike moments [5]. The third set of features is Fourier descriptors [6]. Daubechies-4 Wavelet transform and Gabor wavelet have been used in [5] for subcellular structure identification. Besides these regular features, we introduce a new set of steerable filter features in this paper.

In many image processing and computer vision applications, we need to filter the same image with different orientations. It is neither necessary nor realistic to design filters with arbitrary directions. Steerable filters [7] are a group of filters with finite different directions. Filter with arbitrary orientation can be obtained by linear combination of these basis filters. Because convolution is a linear operator, we can first convolve the image with the basis filters. After the convolved images with basis filters have been generated, filtered image with arbitrary orientation can be obtained through linear interpolation of convolved images with basis filters, which makes large number of computations avoidable.

Let M basis filters be $f^{\theta_j}(x, y)$, $j = 1, 2, \cdots\cdots, M$. The steering constraint is

$$f^{\theta}(x, y) = \sum_{j=1}^{M} k_j(\theta) f^{\theta_j}(x, y) \tag{1}$$

Freeman et al. [7] gave three theories to explain what functions can satisfy the constraint, how many terms M are required and what the interpolation functions are. In this paper, we use two derivative Gaussian functions and their Hilbert Transform which can be approximated by a third-order, odd parity polynomial. They are referred to as G_2 *and* H_2 [7], which have 3 basis functions and 4 basis functions respectively. We use the following equation to calculate global orientation energies in different directions.

$$E(\theta) = (G_2^{\theta})^2 + (H_2^{\theta})^2 \tag{2}$$

Energies of arbitrary orientations can be calculated from equation (2). Three features are selected from the energy histogram: maximum energy, minimum energy, and the ratio of maximum energy to minimum energy. Similar to Gabor wavelet features, another set of steerable features are obtained by interpolating the angle in an interval of half circle. But unlike the Gabor features interpolating from zero angle, the proposed steerable features interpolate from direction corresponding to the maximum energy, which makes the features rotation invariant.

According to characteristics of cell image, we use 5 morphological features, 7 invariant moments, 9 Zernike moments, 10 wavelet moments, 5 Fourier descriptors, and 8 steerable filter features. A total number of 44 features are used.

4 Phase Identification

After each cell has been segmented and tracked successfully, the following step is to determine their phases. There are four phases in cell cycle progression: inter phase, prophase, metaphase, and anaphase. Phase identification aims to classify each cell into one of the four phases using classifiers and extracted features. Fig.1 gives the original cell image containing inter phase cells, prophase cells, metaphase cells, and anaphase cells. Fig. 2 shows four examples of cells in different phases.

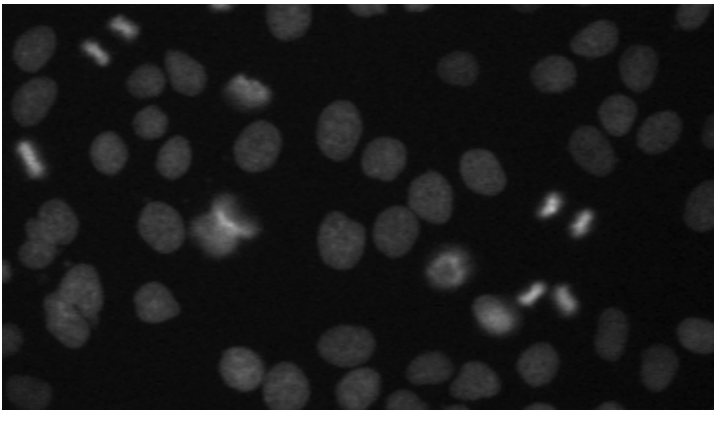


Fig. 1. Original cell image containing four types of cells

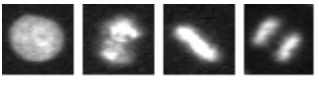


Fig. 2. Four example cells. From left to right: Inter phase, Prophase, Metaphase, and Anaphase.

Before classifying, the features of all cells in the training set and test set are calculated. Since the feature values have different ranges, each components of the feature vector should be normalized to zero mean and unit variance using z-transformation [3].

The KNN and neural network are selected as classifiers for their simplicity and flexibility. Before training, we need to set the network size and the number of hidden units. It is not the fact that big network can obtain better classification performance. Besides the associated computational complexity problems, another major reason why the size of the network should be kept as small as possible is that when the number of free parameters is large, the network tends to adapt to the particular details of the specific training data set, which is known as overfitting. The network should have the smallest possible size to adjust its weights to the largest regularities in the data. It is out of the question to determine the best number of hidden units without training several networks. Too few hidden units will cause high training error and high

generalization error due to underfitting and high statistical bias. Too many hidden units may cause low training error but still have high generalization error due to overfitting and high variance. A convenient rule of thumb is to choose the number of hidden units such that the total number of weights in the net is roughly $n/10$, where n is the number of training points [4].

Therefore, one layer network with 8 hidden units is chosen as classifier (the training point is 72). In the experiment, the inputs are all components of feature vectors and outputs are four Units representing four classes of cells. In the training procedure, the output corresponding to the phase of input cell type is set to 1, otherwise it is set to 0. The training process will halt when the training error converges to the predefined threshold. In the testing step, the phase corresponding to the biggest output is chosen as the phase of the input cell.

5 Experiments

5.1 Analysis of Steerable Filter Features

A feature is effective only when the feature distributions of different phases are in different ranges, thus we will test the steerable filter features from this aspect. Fig. 3 shows four classes of cell images and their energy maps in polar coordinates. From this figure, we can find that the energy map is different in shape and size. Table 1 gives the distributions of maximum energy, minimum energy and max/min ratio. Each feature's mean and variance are calculated and showed in Table 1. From the table it is clear that these features are helpful in improving the classification accuracy. For example, the lower value of maximum energy gives a hand to separate inter phase cells from others, so does the higher value of max/min ratio for metaphase.

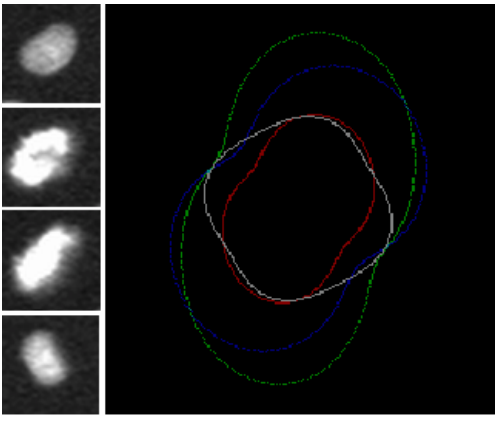


Fig. 3. Energy maps of cells in different phases. Four cells from top to bottom in the left: inter phase cell, prophase cell, metaphase cell, and anaphase cell. Red line represents the inter phase cell, green represents the prophase, blue represents the metaphase, and white represents the anaphase.

5.2 Classification with Neural Network and KNN

Neural network and KNN are used to classify cells with the proposed features. Before classification, the training cells and test cells are extracted from the video sequence using our methods showed in [2]. Then, feature vectors are extracted from all cells to be classified. Since feature values have different ranges, normalization on these feature vectors is necessary.

Table 1. Steerable filter feature distributions of different phases. Mean and variance of maximum energy, minimum energy, and ratio of max/min are listed.

Phase	Mean of maximum energy	Variance of maximum energy	Mean of minimum energy	Variance of minimum energy	Mean of the max/min ratio	Variance of the max/min ratio
Inter phase	2483	883	1104	419	2.320	0.391
Prophase	5997	1868	2753	914	2.254	0.487
Metaphase	8266	1894	1741	600	5.374	2.097
Anaphase	4875	1993	2155	929	2.466	0.980

Four groups of experiments are performed on the dataset. They are KNN with regular features, KNN with regular features plus steerable filter features, neural network with regular features, and neural network with regular features plus steerable filter features. All the experiments are performed with the same training set and test set and experimental results are given in the Table 2.

We analyze these experimental results from two aspects: the performance of two classifiers and the effectiveness of steerable filters features. Nearly equal classification accuracy can be obtained with KNN and neural network, but it is clear that neural network will make the errors distribute in different classes, while KNN makes the

Table 2. Classification accuracies using different classifiers and different feature sets. RFs represents regular features, and SFs represents steerable filter features.

Classifier and features	Inter phase	Pro-phase	Meta-phase	Ana-phase	Overall classification accuracy
KNN with RFs	94.4%	83.3%	93.8%	78.1%	87.4%
KNN with RFs plus SFs	97.2%	88.9%	96.9%	78.1%	89.9%
ANN with RFs	94.4%	77.8%	84.4%	81.3%	84.8%
ANN with RFs plus SFs	97.2%	94.4%	90.6%	90.1%	92.4%

errors assemble in the same class: anaphase class. For this reason, we choose neural network other than KNN as classifier adding to our cell analysis system. If we only use regular features, the classification accuracies are 87.4% and 84.8% for KNN and neural network, respectively. Plus steerable filter features, the classificationaccuracies will increase to 89.9% and 92.4%. Within each class, the performance will also be improved especially for the inter phase class.

In conclusion, steerable filter features are very effective for classifying cells into different phases. The advantages of steerable filters make them robust to noise and rotation invariant. We can extend it from cell classification to other applications, and conclude that the steerable filter features can play an important role in general classifications.

5.3 Feature Reduction with Principal Component Analysis (PCA)

Although we only extracted 44 features to test the new steerable filter features, there are hundreds of existing features which can be directly used in classification and many potential features which may be exploited in the future. Each of these features intends to capture the differences between different image types. However, they might contain redundant information and some of the features may not provide any useful information. In view of computing complexity, it is also improper to use too many features. To solve this problem, two types of feature reduction methods exist, namely feature selection and feature recombination. Feature recombination is more effective than feature selection because it only discards the redundant information and retains the useful information, while feature selection method misses the useful information when discarding the less effective features.

PCA is such a feature recombination method that it can reduce feature number, remove feature “noise”, and extract the “right” features, so we adopt it to reduce computing complexity and remove feature noise in this paper. As discussed in section 5.2, the new steerable filter features are effective for cell classification, and BP neural network has better performance than KNN. Therefore, the features used in our system contain regular features plus steerable filter features, a total of 44 features. 18 features can be obtained by PCA preserving 95% of the eigenvector energy. One-layer neural network with 8 hidden units is used to classify the reduced features. By combining BP neural network and PCA, the system successfully identifies 97.2% of inter phase cells, 77.8% prophase cells, 81.8% metaphase cells, and 90.6% anaphase cells. The total classification accuracy is 88.2%, which demonstrates that PCA can reach a good balance between the feature number and classification accuracy.

6 Conclusion

In this paper, we propose a new feature set based on steerable filters and prove its effectiveness by applying it to classify cells in different phases with KNN and neural network. Comparative experimental results show that the new feature set can provide better classification accuracy than regular feature sets. Neural network and steerable filter features plus regular features can give acceptable classification results, which

makes our automated cell analysis system workable. In the near future, we will evaluate other classifiers such as SVM and resemble methods and exploit new effective features to improve the accuracy of the classification. An automated, accurate and fast cellular analysis system will also be designed for clinical purpose.

Acknowledgement. The funding support of Dr. Xiaobo Zhou and Dr. Stephen Wong is from HCNR Center for Bioinformatics Research Program and the NIH R01 LM008696.

References

1. Yang, X., Li, H., Zhou, X., and Wong, S.T.C.: Automated Segmentation and Tracking of Cells in Time-Lapse Microscopy Using Watershed and Mean Shift. International Symposium on Intelligent Signal Processing and Communication Systems. Hong Kong. In Press
2. Yang, X., Li, H., Zhou, X., and Wong, S.T.C.: Nuclei Segmentation Using Marker-controlled Watershed, Tracking Using Mean-shift and Kalman Filter in Time-Lapse Microscopy. Submitted to IEEE Transactions on Circuits and Systems (2006)
3. Chen, X., Zhou, X., Wong, S.T.C.: Automated Segmentation, Classification, and Tracking of Cancer Cell Nuclei in Time-Lapse Microscopy. IEEE Transactions on Biomedical Engineering. Accepted for Publication (2006)
4. Duda, R.O., Hart, P.E., and Stork, D.G.: Pattern Classification. Second Edition. John Wiley & Sons, Inc, New York (2001)
5. Huang, K., Murphy, R.F.: Boosting Accuracy of Automated Classification of Fluorescence Microscope Images for Location Proteomics. BMC Bioinformatics 78(5) (2004)
6. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 2nd ed. Prentice Hall. (2002)
7. Freeman, W.T., Adelson, E.H.: The Design and Use of Steerable Filters. IEEE Transactions on Pattern Analysis and Machine Intelligence 13(9) (1991) 891-906
8. Manjunath, B.S., Ma, W.Y.: Texture Features for Browsing and Retrieval of Image Data. IEEE Transactions on Pattern Analysis and Machine Intelligence 18(8) (1996) 837-842

Prediction of the Human Papillomavirus Risk Types Using Gap-Spectrum Kernels

Sun Kim and Jae-Hong Eom

Biointelligence Laboratory,
School of Computer Science and Engineering,
Seoul National University, Seoul 151-744, South Korea
{skim, jheom}@bi.snu.ac.kr

Abstract. Human Papillomavirus (HPV) is known as the main cause of cervical cancer and classified to low- or high-risk type by its malignant potential. Detection of high-risk HPVs is critical to understand the mechanisms and recognize potential patients in medical judgments. In this paper, we present a simple kernel approach to classify HPV risk types from E6 protein sequences. Our method uses support vector machines combined with gap-spectrum kernels. The gap-spectrum kernel is introduced to compute the similarity between amino acids pairs with a fixed distance, which can be useful for the helical structure of proteins. In the experiments, the proposed method is compared with a mismatch kernel approach in accuracy and F1-score, and the predictions for unknown types are presented.

1 Introduction

Infection of the Human Papillomavirus (HPV) is known as the major factor of cervical cancer because certain high-risk HPVs develop into cancer. This is because some types of HPV can cause abnormal tissue growth in the form of warts (papillomas) and some HPVs are associated with certain cancers and precancerous conditions. The HPV is a double-strand DNA tumor virus that belongs to the papovavirus family (papilloma, polyoma, and simian vacuolating viruses). More than 100 human types are specific for epithelial cells including skin, respiratory mucosa, or the genital tract. The genital tract HPV types are classified into two or three types by their malignant potential as low-, intermediate-, and high-risk types [1].

Detection of HPV risk types can be seen as a protein function prediction even though functions can be defined at many levels, ranging from biochemical function to biological processes and pathways, all the way up to the organ or organism level [2]. Many approaches for the protein function prediction are based on similarity search between proteins, and the similarity can be defined in multiple ways. For instance, two proteins are regarded to be similar based on the sequence alignment, the structure match, or the motifs comparison. But, these current prediction methods cannot guarantee good performance for all domains. Therefore, it is necessary to develop classification methods which fit in with HPV

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 710–715, 2006.

risk type classification. Eom et al. [3] presented a sequence-based method. They use genetic algorithms to predict HPV risk types using DNA sequences. Joung et al. [4] have proposed a support vector machine (SVM) approach using protein sequences. HPV sequences are first aligned, and a subsequence in high-risk HPVs is selected by hidden Markov models. Then a mismatch kernel method is used to determine the risk types for the selected subsequence. However, this approach has the limitation by exploiting one specific pattern. A text mining technique for HPV classification has been proposed by Park et al. [5] But, the text mining method has the limit for HPV classification because it only uses text material to capture discrimination evidence, and the keyword such as 'high' tends to be appeared in the data set explicitly.

The most important aspect of cervical cancer diagnosis is to determine which HPVs are highly risky or not, whereas the risk types of HPVs are still manually classified by experts, and there is no method to expect the risk types for unknown or new HPVs. In this paper, we present a kernel-based approach to classify HPV risk types from protein sequences. To discriminate low- and high-risk type, we use a SVM classifier with the string kernel that is introduced to deal with protein sequences. It maps protein sequences into a high-dimensional space and computes the similarity between amino acids pairs in the space. The feature of the string kernel is to consider a fixed gap between amino acids for simplified classification and performance improvement, which is based on the assumption motivated by protein's properties.

The remainder of the paper is organized as follows. In Section 2, we describe the SVM classification using gap-spectrum kernels. In Section 3, experimental data and results are presented and conclusions are drawn in Section 4.

2 SVM Classification Using Gap-Spectrum Kernels

Here, we use support vector machines to discriminate HPV risk types. A string kernel-based SVM is trained on given protein sequences and tested on unknown sequences. When the SVM is used for classification problem, a kernel and a set of labeled vectors, which is marked to positive or negative class are given. The kernel functions introduce nonlinear features in hypothesis space without explicitly requiring nonlinear algorithms. SVMs find a linear decision boundary in the feature space mapped by the kernel in order to separate the data into two classes.

The recently presented string kernels such as the mismatch kernels, restricted gappy kernels, substitution kernels, and wildcard kernels provide good insights for SVM protein classification [6]. However, we use a simple string kernel reflecting HPV protein's properties, which is based on the spectrum kernel methods. The spectrum kernel has been used to detect remote homology detection in protein sequences [7][8].

The input space $\mathcal{X}$ consists of all finite length sequences of characters from an alphabet $\mathcal{A}$ of size $|\mathcal{A}| = l$ ($l = 20$ for amino acids in protein). Given a number $k \geq 1$, the k-spectrum of a protein sequence is the set of all possible

k-length subsequences (k-mers) that it contains. The feature map is indexed by all possible subsequences a of length k from $\mathcal{A}$. The k-spectrum feature map $\Phi_k(x)$ from $\mathcal{X}$ to $\mathbb{R}^{l^k}$ can be defined as:

$$\Phi_k(x) = (\phi_a(x))_{a \in \mathcal{A}^k}. \tag{1}$$

where $\phi_a(x)$ = number of occurrences of a occurs in x. Thus the k-spectrum kernel function $K^s(x_i, x_j)$ for two sequences, x_i and x_j is obtained by taking the inner product in the feature space:

$$K_k^s(x_i, x_j) = \langle \Phi_k(x_i), \Phi_k(x_j) \rangle. \tag{2}$$

Now, we want to modify the spectrum kernel to fit in with HPV risk type classification. Proteins are linear chains of amino acids, which are made during the process of translation. But, the natural shape of proteins are not such as straight lines, rather three-dimensional structures formed by protein folding. Secondary structure of a protein defines as the general three-dimensional form of local regions and may include regions of α helices, β sheets, or other segments. The α helices and β sheets are major features of protein architecture. The amino acids in an α helix are arranged in a helical structure. A β sheet consists of two or more amino acid sequences within the same protein that are arranged adjacently and in parallel. The mechanism of protein folding is so complex and not entirely understood, but the structure of a similar homologous sequence can be helpful to identify the tertiary structure of the given sequence. Especially, HPVs are relatively short and have relatively small region of β sheets in some gene products, predicted by secondary structure systems. Here, we meet the following assumptions:

1. The amino acids pair with certain distance affect HPV's risk type function more than consecutive amino acids (k-mers) according to its 3-dimensional structure property.
2. HPV risk types can be identified by the amino acids pair with the fixed distance, which mostly influence on risk type decision.

Based on the assumptions, we want to define a string kernel, the gap-spectrum kernel based on the k-spectrum. For a fixed k-mer $a = a_1 a_2 \dots a_k,\ a_i \in \mathcal{A}$, 2-length sequence $\beta = a_1 a_k,\ \beta \in \mathcal{A}^2$. The β indicates the amino acids pair with (k-2) gap. The feature map $\Psi_k(x)$ is defined as:

$$\Psi_k(x) = (\phi_\beta(x))_{\beta \in \mathcal{A}^2}. \tag{3}$$

where $\phi_\beta(x)$ = number of occurrences of β occurs in x.

Furthermore, RBF kernel is appended to increase the discrimination performance between the HPV sequences. By closure properties of kernels, the gap-spectrum kernel $K_k(x_i, x_j)$ is defined as follows:

$$K_k(x_i, x_j) = K'(\Psi_k(x_i), \Psi_k(x_j)) \tag{4}$$

$$= \exp\left(-\gamma \|\Psi_k(x_i) - \Psi_k(x_j)\|^2\right). \tag{5}$$

where $\gamma > 0$.

3 Experimental Results

In this paper, we use the HPV database in Los Alamos National Laboratory (LANL, http://hpv-web.lanl.gov/stdgen/virus/hpv) to get protein sequences. Viral early protein E6 is known for inducing immortalization and transformation in rodent and human cell types. The E6 gene product produced by the high-risk HPV type can bind to and inactivate the tumor suppressor protein, thus facilitating tumor progression [9]. Thus we have selected the E6 sequences for HPV risk type classification from the LANL database.

The risk types of HPVs have been determined based on the HPV compendium (1997) in LANL. If a HPV belongs to skin-related or cutaneous groups, the HPV is classified into low-risk. On the other hand, a HPV is classified as high-risk if it is known to be high-risk type for cervical cancer. The comments in LANL database are used to decide risk types for some HPVs, which are difficult to be classified. Seventeen sequences out of 72 HPVs were classified as high-risk types (16, 18, 31, 33, 35, 39, 45, 51, 52, 56, 58, 59, 61, 66, 67, 68, and 72), and others were classified as low-risk types. Four HPVs (26, 54, 57, and 70) were marked as 'unknown' because their risk type could not be determined.

For experiments, leave-one-out cross-validation is used to determine the classification performance. Table 1 presents the accuracy changes with given k in Equation (5). It shows the accuracy has the highest performance (97.22%) when $k = 4$. $k = 4$ means that we only consider the amino acids pairs that have two gaps between amino acids in the SVM classification. When $k = 2$, the gap-spectrum kernel does not allow a gap between amino acids. The accuracy for $k = 2$ shows 94.44% as shown in Figure 1. Here we can conclude the amino acids pair with certain distance can provide more evidence than consecutive amino acids to discriminate low- and high-risk HPV proteins. In other words, it is possible that the α helix structure of E6 proteins has an important role to result in the high-risk type.

Our approach is also compared with the mismatch kernel method in classification performance. Figure 1 shows the comparison of the gap-kernel method and the mismatch kernel method with sequence alignment, which has been proposed by Joung et al. [4] The F1-score is computed for the precision p and recall r as F1-score = (2pr)/(p + r). F1-score is helpful to evaluate how well the classifier did for the assigned high-risk HPV types. The gap-spectrum kernel with $k = 4$ shows 97.22% of accuracy and 95.00% of F1-score, and the gap-spectrum kernel with $k = 2$ shows 94.44% of accuracy and 90.23% of F1-score. $k = 2$ is exactly same as the SVM classification using RBF kernel with 2-spectrum method. But, the mismatch kernel method shows 93.15% of accuracy and 85.71% of F1-score.

Table 2 shows the prediction results for four HPVs marked as unknown risk type. HPV26, HPV54, HPV57, and HPV70 are predicted as high-, low-, low-,

Table 1. Accuracy changes by the gap-spectrum kernel

k	3	4	5	6	7
Accuracy	94.44	97.22	95.83	95.83	94.44

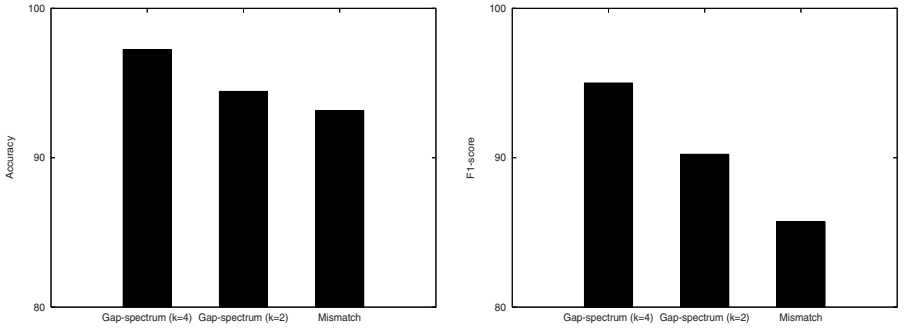


Fig. 1. The performance comparison of gap-spectrum and mismatch kernel methods

Table 2. Predicted risk type for the HPVs which risk types are unknown

Type	HPV26	HPV54	HPV57	HPV70
Risk	High	Low	Low	High

and high-risk, respectively. The prediction results for HPV26 and HPV54 are identical to the one in Muñoz et al. [10], and we assume that their results are reliable because it is based on epidemiologic classification from over 1,900 patients. For HPV70, there are different results for HPV risk type by previous research [10][11][12], and the risk type of HPV57 cannot be decided as yet because of the insufficient published works.

4 Conclusion

In this paper, we have proposed a kernel method to classify the HPV risk types, which are closely related to cervical caner. The proposed approach uses E6 gene product, and the SVM classifier with a simple string kernel is used to discriminate HPV risk types from the protein sequences. The gap-spectrum kernel is defined to consider amino acids pair with a fixed distance, which can be useful for α helix-dominant proteins. For experimental results, the classification performance was measured based on leave-one-out cross-validation for 72 HPV sequences, and it showed our approach provides better performance than mismatch kernel methods combined with the sequence alignment. The prediction performance for the unknown types was also promising.

References

[1] Janicek, M.F. and Averette, H.E.: Cervical Cancer: Prevention, Diagnosis, and Therapeutics. Cancer Journals for Clinicians **51** (2001) 92–114

[2] Watson, J.D., Laskowski, R.A., and Thornton, J.M.: Predicting Protein Function from Sequence and Structural Data. Current Opinion in Structural Biology **15** (2005) 275–284

[3] Eom, J.H., Park, S.B., and Zhang, B.T.: Genetic Mining of DNA Sequence Structures for Effective Classification of the Risk Types of Human Papillomavirus (HPV). In: Proceedings of the 11th International Conference on Neural Information Processing (2004) 1334–1343
[4] Joung, J.G., O, S.J, and Zhang, B.T.: Prediction of the Risk Types of Human Papillomaviruses by Support Vector Machines. In: Proceedings of the 8th Pacific Rim International Conference on Artificial Intelligence (2004) 723–731
[5] Park, S.B., Hwang, S., and Zhang, B.-T.: Mining the Risk Types of Human Papillomavirus (HPV) by AdaCost. In: Proceedings of the 14th International Conference on Database and Expert Systems Applications (2003) 403–412
[6] Leslie, C. and Kuang, R.: Fast String Kernels using Inexact Matching for Protein Sequences. Journal of Machine Learning Research **5** (2004) 1435–1455
[7] Leslie, C., Eskin, E., and Noble, W. S.: The Spectrum Kernel: A String Kernel for SVM Protein Classification. In: Proceedings of the Pacific Symposium on Biocomputing (2002) 564–575
[8] Leslie, C., Eskin, E., et al.: Mismatch String Kernels for Discriminative Protein Classification. Bioinformatics **20**(4) (2004) 467–476
[9] Pillai, M., Lakshmi, S., et al.: High-Risk Human Papillomavirus Infection and E6 Protein Expression in Lesions of the Uterine Cervix. Pathobiology **66** (1998) 240–246
[10] Muñoz, N., Bosch, F.X., et al.: Epidemiologic Classification of Human Papillomavirus Types Associated with Cervical Cancer. New England Journal of Medicine **348** (2003) 518–527
[11] Longuet, M., Beaudenon, S., and Orth, G.: Two Novel Genital Human Papillomavirus (HPV) Types, HPV68 and HPV70, Related to the Potentially Oncogenic HPV39. Journal of Clinical Microbiology **34** (1996) 738–744
[12] Meyer, T., Arndt, R., et al.: Association of Rare Human Papillomavirus Types with Genital Premalignant and Malignant Lesions. Journal of Infectious Diseases **178** (1998) 252–255

Extreme Learning Machine for Predicting HLA-Peptide Binding

Stephanus Daniel Handoko[1], Kwoh Chee Keong[1], Ong Yew Soon[1], Guang Lan Zhang[1,2], and Vladimir Brusic[3]

[1] School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, 639798 Singapore
[2] Institute for Infocomm Research, 21 Heng Mui Keng Terrace, 1196131 Singapore
[3] School of Land and Food Sciences and Institute for Molecular Bioscience, University of Queensland, Brisbane QLD 4072, Australia

Abstract. Machine learning techniques have been recognized as powerful tools for learning from data. One of the most popular learning techniques, the Back-Propagation (BP) Artificial Neural Networks, can be used as a computer model to predict peptides binding to the Human Leukocyte Antigens (HLA). The major advantage of computational screening is that it reduces the number of wet-lab experiments that need to be performed, significantly reducing the cost and time. A recently developed method, Extreme Learning Machine (ELM), which has superior properties over BP has been investigated to accomplish such tasks. In our work, we found that the ELM is as good as, if not better than, the BP in term of time complexity, accuracy deviations across experiments, and – most importantly – prevention from over-fitting for prediction of peptide binding to HLA.

1 Introduction

Major Histocompatibility Complex (MHC) molecules play an important role in immune system [1]. In humans, the MHC molecules are known as Human Leukocyte Antigens (HLA). They bind short peptides and present them on the cell surface for recognition by the T-cells of the immune system. HLA genes show high polymorphism; in fact, the HLA genes are the most polymorphic human genes [2]. There are more than 2000 known HLA variants characterized to date [3]. Due to this high variability, it would be impossible to carry out wet-lab experiments for each and every possible HLA-peptide pair. Hence, computer models are built and used for predicting whether a given peptide is likely to bind a particular HLA molecule. This allows us to carry out wet-lab experiments only for those peptides that are pre-selected by computational screening and therefore have high probability of being binders. Brusic and his co-workers earlier implemented the Back-Propagation (BP) Artificial Neural Networks [4][5][6] and the Hidden Markov Model (HMM) [7] for high-accuracy prediction of HLA-peptide binding. In this work, we have investigated

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 716 – 721, 2006.

the BP and the Extreme Learning Machine (ELM) [8] and compared their predictive and technical performances in prediction of peptide binding to HLA molecules.

2 Review of Current Machine Learning Method in Predicting HLA-Peptide Binding

The problem of predicting peptides that bind to MHC molecules has been extensively investigated since late 1980s. A number of predictive methods for MHC classes I and II binding peptides are available, including those based on structural modeling and free energy calculations of peptide/MHC complexes and sequence-based methods. Structural modeling methods [9] are computationally intensive, and have mainly been applied to MHC molecules with known crystal structures. Sequence-based statistical methods can be classified into two groups. The AIB methods [10], such as binding motifs and quantitative matrices, assume independent contributions of peptide residues to peptide/MHC binding. The general methods, such as classification and regression trees, Neural Networks (NN), Hidden Markov Models (HMM), or Support Vector Machines (SVM), multivariate statistical approaches and decision trees, take into account possible combined influences of multiple residues in different positions in a peptide. Decision trees [11] were used to predict MHC-peptide binding. Decision trees use a natural and intuitive way to classify a pattern through a sequence of questions in which the next question asked depends on the answer to the current question [12]. In Segal's work, it was shown that 70% of the positive T-cell responses and 84% for the negative ones can be identified correctly. NNs are a class of machine learning techniques that are biologically inspired and have shown great success in many real world applications, and particularly well suited to perform classification and complex pattern recognition tasks [13]. Various NNs have been used for prediction of peptide bindings to MHC [4][5] and their prediction performance compare favorably to other existing prediction methods given sufficient training dataset [14]. HMMs are a class of probabilistic discrete models. A HMM is defined by a finite set of states representing the modeled system and have been demonstrated to be effective for MHC binders prediction [7]. Support Vector Machines (SVM) are discriminative supervised machine learning methods. They have been used for predicting MHC class I binding peptides and the results in [15] indicate that they predict well even on relatively small peptide datasets. Multivariate statistical approaches [16] are based on partial least squares. The prediction is obtained from a combination of individual amino acid contributions at each position of the peptide and contributions from side chain side to chain interactions. A more detailed review on computational methods for prediction of MHC binding peptides is given in [17].

3 Data and Methods

In this work, the dataset consists of 3,050 9-mer peptides related to 15 variants of the HLA-A2 supertype. It is similar to the dataset used in [6], except that the current set contains 88 new peptides. The sequences of HLA-A2 variants are obtained from the IMGT/HLA Sequence Database [18]. The sequence encoding scheme is the same as

that described in [6]. This dataset was used to train and test the Back-Propagation (BP) Neural Networks and the Extreme Learning Machine (ELM). BP are very cost intensive in terms of time and memory requirements. In term of the performance, high accuracy will normally be achieved as this method iteratively adjusts the weight of each connection such that the outputs are close to those desired. Different sets of initial weights result in different adjustments and hence, different outputs. Normally, a series of trials are carried out, each with randomly assigned initial weight. Recently, a new method called ELM has been introduced. ELM is quite similar to BP, except that ELM employs significantly faster algorithm than the BP resulting in much shorter computation time. The ELM consists of three layers: input, hidden, and output layers. The weights of the connections between the input and the hidden layer (input weights) are assigned randomly. The weights of the connections between the hidden and the output layer (output weights) are calculated by using Moore-Penrose pseudoinverse [19]. Since no iterative computations are needed, ELM trains very fast. However, large memory is required for the calculation of the pseudoinverse, which is a tradeoff for fast training. In this work, a series of 25 experiments were carried out, each with sigmoid activation function for the hidden layer and linear activation function for the output layer.

4 Result

In this section, we discuss the results of an empirical study on using BP and ELM for predicting peptide binders to HLA-A2. In our experiments, we used the 5-fold cross validation scheme. Each time, one subset is selected as the testing data while the other four subsets are used as the training data. Various architectures (numbers of hidden neurons) were studied for both BP and ELM. In particular, the number of hidden neurons for studied ELMs ranged from 2 to 500. On the other hand, we consider only 2 to 11 hidden neurons when building BP models due to large training time and rapidly increasing memory requirements with increasing number of hidden neurons. The results demonstrate that BP tends to outperform ELM for small number of hidden neurons.

The average computational time, *i.e.* for 25 repeated experiments, to train BP and ELM models for various numbers of hidden neurons are plotted in Fig. 1 and 2,

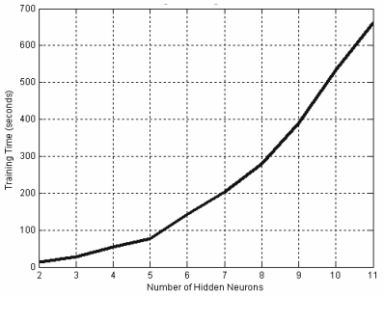


Fig. 1. Average training time of the BP

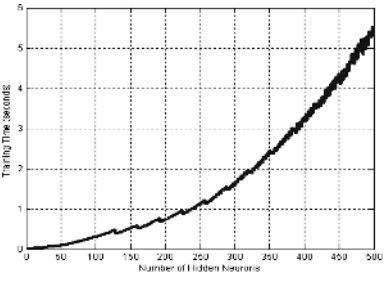


Fig. 2. Average training time of the ELM

respectively. For 500 hidden neurons, ELM takes less than six seconds for training while BP with only two hidden neurons consumes more than 10 seconds on average. It is clear from these results that the training time of BP is significantly higher than that of ELM.

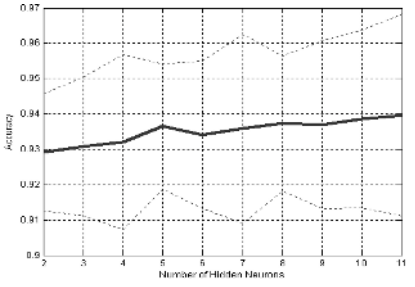


Fig. 3. Average training accuracy of the BP

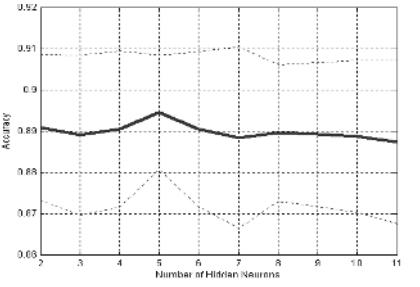


Fig. 4. Average testing accuracy of the BP

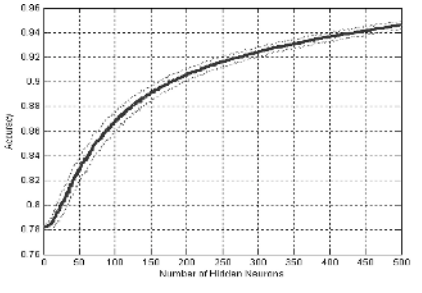


Fig. 5. Average training accuracy of the ELM

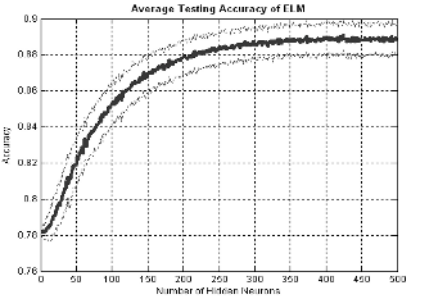


Fig. 6. Average testing accuracy of the ELM

The average prediction accuracy across 25 repeated experiments for both methods are given in Fig. 3 to 6. The solid and dotted curves in these figures represent the average prediction accuracies and standard deviations, respectively. In Fig. 3, the training accuracy of the BP is given. As we increase the number of the hidden neurons, the training accuracy increases. From the graph, it can be deduced that the standard deviations across experiments are approximately two percents. In Fig. 5, the training accuracy of the ELM is given. As we increase the number of the hidden neurons, the training accuracy also increases. However, the ELM model produces less accuracy deviations as compared to those of the BP model. The ELM, nevertheless, performs poorly with small number of hidden neurons while this is not the case for the BP. Only with large number of hidden neurons can the performance of the ELM be compared to that of the BP. As the ELM requires few seconds only for training, introducing more hidden neurons is not a practical constraint here. Fig. 4 provides the testing accuracy plot of the BP. It can be clearly seen that as we increase the number of the hidden neurons beyond five, the accuracy deteriorates. Fig. 5 provides the testing accuracy plot of the ELM. As the number of the hidden neurons is increased,

the accuracy is initially increased but stabilizes after around 300 hidden neurons are used. With large number of hidden neurons, the performance of the ELM is really comparable to that of the BP. In term of standard deviations, it is very much similar to the one discussed above.

Another important issue is the over-fitting problem that troubles many NN implementations. Increasing the number of hidden neurons of a model increases its capability of memorizing. However, this would normally decrease its capability of generalizing. Thus, the over-fitting occurs. From Fig. 3 and Fig. 4, we can see that there is an indication of over-fitting for BP. The model with more hidden neurons has greater capability of memorizing the variability of data during the training phase and that has resulted in poorer capability of generalizing its prediction for the data points it has not seen before. This is, however, not shown in Fig. 5 and Fig. 6 for ELM. As the capability of an ELM model is getting better and better with more hidden neurons, it does not lose its capability of generalizing its prediction about the data points it has not yet seen. These results provide evidence that the ELM has the intrinsic resistance to over-fitting.

5 Conclusions

From our experiments, we found out that the ELM has several advantages as compared to the BP. The most obvious one is the computation time required to train the model. Using 2,440 data points, the ELM only needs less than six seconds to train a model with 500 hidden neurons while the BP requires approximately ten seconds to train a model with only two hidden neurons. Hence, introducing larger number of hidden neurons to facilitate larger amount of data points shall not be a problem in the ELM, as long as there is enough memory space to facilitate the calculations. In term of accuracy, the ELM is comparable to the BP even though the ELM needs much more hidden neurons to achieve similar performance to the BP. As the time complexity is not an issue in the ELM, this trade-off is worthwhile. Another point worth noting is that the ELM produces smaller variations of accuracy with respect to the training and the testing datasets. Lastly, an important point observed in the experiments is that the ELM is resistant to over-fitting. While the training accuracy increases as more and more hidden neurons are introduced, the testing accuracy rises steadily and remains stable after a certain number of hidden neurons were introduced. This indicates that the learning of the ELM improves with the increasing number of hidden neurons, without the loss of generalization capability.

Acknowledgement

This work has been funded in part by Nanyang Technological University and (ZGL and VB) with the USA Federal funds from the National Institute of Allergy and Infectious Diseases, National Institutes of Health, Department of Health and Human Services, under Grant No. 5 U19 AI56541 and Contract No. HHSN 266200400085C.

References

1. Janeway, C. A., Travers, P., Walport, M., Shlomchik, M.: Immunobiology. Garland Publishing, New York (2001)
2. Williams, T. M.: Human Leukocyte Antigen Gene Polymorphism and the Histocompatibility Laboratory. J. Mol. Diagn 3 (2001) 98-104
3. Marsh, S. G. E., et al.: Nomenclature for Factors of the HLA System. Tissue Antigens 65 (2005) 301-369
4. Brusic, V., Rudy, G., Honeyman, G., Hammer, J., Harrison, L. C.: Prediction of MHC Class II-binding Peptides Using an Evolutionary Algorithm and Artificial Neural Network. Bioinformatics 14 (1998) 121-130
5. Honeyman, M. C., Brusic, V., Stone, N. L., Harrison, L. C.: Neural Network-based Prediction of Candidate T-cell Epitopes. Nat. Biotech. 16 (1998) 966-969
6. Zhang, G. L., Khan, A. M., Srinivasan, K. N., August, J. T., Brusic, V.: Neural Models for Predicting Viral Vaccine Targets. J. Bioinform. Comput. Biol. 3(5) (2005) 1207-1225
7. Brusic, V., Petrovsky, N., Zhang, G. L., Bajic, V. B.: Prediction of Promiscuous Peptides That Bind HLA Class I Molecules. Immu. Cell. Biol. 80 (2002) 280-285
8. Huang, G. B., Zhu, Q. Y., Siew, C. K.: Extreme Learning Machine: A New Learning Scheme of Feedforward Neural Networks. IJCNN (2004).
9. Schueler-Furman, O., Altuvia, Y., Sette, A., and Margalit, H.: Structure-based Prediction of Binding Peptides to MHC Class I Molecules: Application to a Broad Range of MHC Alleles. Protein Sci. 9 (2000) 1838-1846
10. Peters, B., Tong, W., Sidney, J., Sette, A., and Weng, Z.: Examining the Independent Binding Assumption for Binding of Peptide Epitopes to MHC-I Molecules. Bioinformatics 19 (2003) 1765-1772
11. Savoie, C. J., Kamikawaji, N., Sasazuki, T., and Kuhara, S.: Use of BONSAI Decision Trees for the Identification of Potential MHC Class I Peptide Epitope Motifs. Pac. Symp. Biocomput. (1999) 182-189
12. Duda, R. O., Hart, P. E., and Stork, D. G.: Pattern Classification. Wiley-Interscience (2001)
13. Zurada, J. M.: Introduction to Artificial Neural Systems. PWS Publishing Co, St. Paul MN USA (1999)
14. Yu, K., Petrovsky, N., Schönbach, C., Koh, J. Y., Brusic, V.: Methods for Prediction of Peptide Binding to MHC Molecules: A Comparative Study. Mol. Med. 8 (2002) 137-148
15. Donnes, P. and Elofsson, A.: Prediction of MHC Class I Binding Peptides Using SVMHC. BMC Bioinformatics 3 (2002) 25-38
16. Hattotuwagama, C. K., Guan, P., Doytchinova, I. A., Zygouri, C., Flower, D.R.: Quantitative Online Prediction of Peptide Binding to the Major Hstocompatibility Complex. J. Mol. Graph Model 22 (2004) 195-207
17. Brusic, V., Bajic, V. B., Petrovsky, N.: Computational Methods for Prediction of T-cell Epitopes—A Framework for Modelling, Testing, and Applications. Methods 34 (2004) 436-443
18. IMGT/HLA Sequence Database Rel. 2.11.0. URL: www.ebi.ac.uk/imgt/hla/.
19. Albert, A. E.: Regression and the Moore-Penrose Pseudoinverse. Academic Press, New York (1972)

Identifying Transcription Factor Binding Sites Based on a Neural Network

Zhiming Dai, Xianhua Dai, and Jiang Wang

School of Information Science and Technology, Sun Yat-sen University,
Guangzhou 510275, P.R. China
issdxh@zsu.edu.cn

Abstract. The identification of regulatory motifs (transcription factor binding sites) in DNA sequences is a difficult pattern recognition problem. Many methods have been developed in the past few years. Although some are better than the others in a sense, yet not a single one is recognized to be the best. Generally, in the case of long and subtle motifs, exhaustive enumeration becomes problematic. In this paper, we present a new method which improves exhaustive enumeration based on a neural network. We test its performance on both synthetic data and realistic biological data. It proved to be successful in identifying very subtle motifs. Experiments also show our method outperforms some popular methods in terms of identifying subtle motifs. We refer to the new method as IMNN (Identifying Motifs based on a Neural Network).

1 Introduction

Regulatory motifs (transcription factor binding sites) are conserved short subsequences (6-20 nucleotides in length) in the upstream non-coding DNA sequences, shared by a family of co-regulated genes, which are bound by the transcription factors to activate or repress gene expression. The identification of a motif can be defined to look for a short common pattern in the sample (a collection of sequences). Although it might seem simple at first, it is very complex.

There were many methods proposed to identify motifs in the past, including GibbsDNA [1], CONSENSUS [2], MEME [3], ANN-Spec [4] and other methods. However, Pevzner and Sze [5] defined a "Challenge Problem" for motifs identification, which most of the above methods fail to solve.

***Challenge Problem*:** Find a signal in a sample of sequences, each 600 nucleotides long and each containing an unknown signal (pattern) of length 15 with 4 mismatches. We also refer to it as planted (*15, 4*)-motif problem. Note that the planted (*15, 4*)-motif problem is a particular parameterization of the planted (*l, d*)-motif problem.

To solve the "Challenge Problem", researchers proposed some new methods, such as WINNOWER [5], Graphical Approach, CMF, PATTERNBRANCHING, Random Projection and Uniform Projection. Although these methods can identify subtle motifs at a high success rate, the computation cost is still a big problem. In [6], we present a new method to reduce the computation cost. In this paper, we present a novel method

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 722 – 727, 2006.

named IMNN for the motifs identification problem. In our opinion, the best methods for identifying motifs are those which rely on exhaustive enumeration. However, it becomes impractical for large l and subtle motifs. A natural idea is to make use of a machine learning technique to improve exhaustive enumeration so that we can get the best tradeoff between sensitivity and efficiency. IMNN is such a method based on a neural network. IMNN searches for the parameters of a neural network that will maximize the scoring scheme of alignment. It has the advantages of the high sensitivity of exhaustive enumeration and the efficiency of our constructed network. We test its performance on both synthetic data and realistic biological data. It proved successful in identifying very subtle motifs.

2 IMNN

2.1 Preliminaries

Definition 1. For an l-mer A and each sequence s_i in the sample $S = \{s_1, s_2, ..., s_n\}$, let $dis(A, s_i) = \min\{dis(A, P) \mid P \in s_i\}$, where P denotes an l-mer and $dis(A, P)$ denotes the Hamming distance between A and P, that is, the number of mismatches between A and P. Then the total distance of A from the sample is $dis(A, S) = \sum_{s_i \in S} dis(A, s_i)$.

Definition 2. Each element in a sequence is encoded for presentation to the network. We use unique one-hot codes (one bit on, the rest are off) for the four symbols in the sequence.

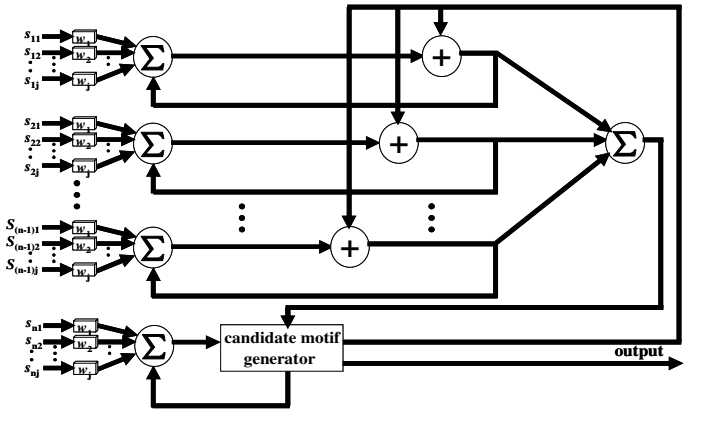


Fig. 1. The network of IMNN. Given a collection $S = \{s_1, s_2, ..., s_n\}$ of n DNA sequences as input, the network identifies the consensus motif for output. s_{ik} denotes the nucleotide on the kth position of the ith sequence.

2.2 The Network

The network of IMNN is depicted schematically in Fig.1. At the input layer, every sequence of length j is denoted by a char vector, which times the weight vector. In

this architecture, the weight value is either 1 (the connection between the corresponding symbol and the input layer is on) or 0 (the connection is off). The network accepts its input in one time step according to the size l of the motif. It uses an input window of length l and process the sequence by moving this window across the sequence until all of the input has been exposed to the network. In this way, all the l-mers in the sequence are exposed to the network one after another. The network works its way across the sequence iteratively, continually modifying its internal state in light of the Hamming distance between the current input and the current state of the network (the current candidate motif). So, the input sequence can be broken into (j-l+1) windows W, then the processing can be defined as taking one window at a time, working recursively from 1 to (j-l+1) (the second index on the inputs in (1)). A null vector terminates the recursion.

$$x_i = \min(dis(\phi(W_{i1}), \phi(CM)), dis(\phi(W_{i2}), \phi(CM)), ...) \tag{1}$$

where CM signifies the current Candidate Motif. The next process is

$$f(x_1, ..., x_{n-1}) = \sum_{i=1}^{n-1} x_i \,. \tag{2}$$

According to the sum value, the candidate motif generator determines whether to take the current candidate motif for output or to generate a new candidate motif for the hidden layer. The candidate motif generator, which is the keystone of IMNN, will be introduced in detail in the next subsection.

2.3 Implementation Issues

2.3.1 The Candidate Motif Generator

We choose a sequence s_n from the input sample as a reference sequence at random. The candidate motif generator accepts an l-mer in one time step. Like the other sequences, we use an input window of length l and process the sequence by moving this window across the sequence until the candidate motif generator has identified a consensus motif for output or all the input has been exposed to the network. We assume that there is at least one instance of the consensus motif in every input sequence, so this instance will certainly be transmitted to the candidate motif generator, where several positions of the instance are mutated in light of the length l so that it will become the consensus motif for output. Note that the planted (l, d)-motif problem defines nearly the worst case for subtle motifs, but in realistic biological data, motifs are often much better conserved than these subtle motifs. Therefore, when we develop a method to identify motifs, we must consider both well conserved motifs and subtle motifs.

Let M be the consensus motif, viewed as an l-mer, and let A be an instance of M in the reference sequence with at most d mismatches. Given A, how do we generate M? It means that we exhaustively enumerate every potential l-mer CM (Candidate Motif) that satisfies $dis(A, CM) \le d$, and then score each pattern according to some metrics (The pattern with the highest score is the consensus motif). But the computation cost of this method is very large, and therefore we propose another method to reduce the computation cost as follows. For convenience, we take the planted (8, 2)-motif

problem for example. In this case, it satisfies $dis(M, s_i) \le 2, (i = 1,...,n)$, which means that if an *8*-mer *CM* satisfies $dis(CM, s_i) \le 2, (i = 1,...,n)$ and $dis(A, CM) \le 2$, this *CM* will be considered to be *M*. First of all, if the *A* satisfies $dis(A, s_i) \le 2, (i = 1,...,n)$, this *A* will be considered to be *M* for output. Secondly, we will replace the original symbol with the other three symbols on each position of *A* respectively, so we can generate $8 \times 3 = 24$ *CMs*. If one of *CMs* satisfies $dis(CM, s_i) \le 2, (i = 1,...,n)$, this *CM* will be considered to be *M* for output. Thirdly, note that if one *CM* is an instance of *M*, it will satisfy $dis(CM, M) \le 1$ and $dis(CM, s_i) \le 3, (i = 1,...,n)$. In other words, if one *CM* satisfies $\max(dis(CM, s_i)) > 3, (i = 1,...,n)$, this *CM* will not be an instance of *M* and must be washed out. For the remainder, we will replace the original symbol with the other three symbols on each position of *CM* respectively, except the position on which there is a mismatch between *CM* and *A*, in this way, we can generate several new *CMs*. If one new *CM* satisfies $dis(CM, s_i) \le 2, (i = 1,...,n)$, it will be considered to be *M* for output; otherwise, it will be washed out. In this way, we can greatly reduce the number of *CMs* to analyze. More generally, we can adjust the maximal number of steps in light of the value *l*.

The candidate motif generator processes every input based on the above method. According to the sum value from the hidden layer, it determines whether to send a signal to the input layer to ask for a new input or not, whether to generate a new candidate motif for the hidden layer or not, whether to take the current candidate motif for output or not.

2.3.2 Speedups for IMNN

At the hidden layer, it is unnecessary to compute the Hamming distance value between the current candidate motif and every subsequence in the input sequence. The recursion can be terminated if the Hamming distance value between the current candidate motif and any subsequence is less than the threshold. This threshold is set dynamically by the candidate motif generator. Take the planted (*8, 2*)-motif problem for example, the threshold can be set at three in the first step.

Table 1. Results of four popular methods on our experiment data

Method	Success rate	Method	Success rate
CONSENSUS	5%	WINNOWER	93%
Gibbs DNA	9%	SP-STAR	81%

Table 2. The realistic biological motifs in our experiment data

Name	Species	Class	Consensus motif
Pbx	Homo sapiens	HOMEO	ATCAATCAA
SAP-1	Homo sapiens	ETS	ACCGGAAGT
P50	Homo sapiens	REL	GGGGATTCCCC
P53	Homo sapiens	REL	GGACATGCCCGGGCATGT

Note that A is an l-mer in the sample, which is only one symbol different from the two adjacent As. We can compute $dis(A, s_i), (i = 1,...,n)$ at the hidden layer by sharing computations across different l-mer As, reducing the computation cost to one lth of the original cost.

3 Results

IMNN is implemented on Matlab. We test the performance of IMNN on both synthetic data and realistic biological data. It proved to be successful in identifying very subtle motifs.

3.1 Results on Synthetic Data

Our experiment data is a number of real upstream non-coding DNA sequences sets of a variety of lengths, which are obtained from [7] and implanted with (l, d)-motifs. IMNN can identify the implanted motifs at the success rate of approximately 100%. Then we experiment with other popular methods on our experiment data. The results are shown in Table 1. We can learn from Table 1 that IMNN outperforms these popular methods in terms of identifying subtle motifs.

In the planted (l, d)-motif identification, the motif M is known, and thus, we can run the method until the consensus motif is identified or all the input has been exposed to the network. However, in a real biological motif identification problem, the motif M is unknown, we have to run the method until all the input has been exposed to the network.

3.2 Results on Realistic Biological Data

We test IMNN on the following biological data with known motifs from the database [8]: four samples containing Pbx, SAP-1, p50 and p53 binding sites respectively (see Table 2). IMNN can also identify the known motifs in the samples. The motifs in these data sets are much better conserved than those in the synthetic data sets, with little variation. IMNN can identify both well conserved motifs and subtle motifs successfully. In general, we have been unable to locate published examples of biological motifs as subtle as those in the planted (l, d)-motif problem, but the dearth of such examples does not imply that subtle motifs do not exist biologically.

4 Conclusions

We have presented a new method IMNN which improves exhaustive enumeration based on a neural network. IMNN searches for the parameters of a neural network that will minimize the Hamming distance between the output motif and the input sample. It has the advantages of the high sensitivity of exhaustive enumeration and the efficiency of our constructed network. It can identify subtle motifs successfully. Experiments on synthetic data show IMNN outperforms other popular methods in terms of identifying subtle motifs. To our knowledge, there is only one existing method to identify motifs based on neural networks [4], which IMNN is different

from. In summary, we have only begun to explore the potential of the application of neural networks to the motifs identification. To this end, we have shown in experiments that our constructed network can identify subtle motifs at a higher success rate than traditional methods. As for the future work, we intend to improve IMNN by reducing the computation cost and identifying motifs of unknown size.

Acknowledgments

This work is supported by the National Science Foundation of China (NSFC), Grant 60474075.

References

1. Lawrence, C.E., Altschul, S.F., Boguski, M.S., Liu, J.S., Neuwald, A.F., Wootton, J.C.: Detecting Subtle Sequence Signals: a Gibbs Sampling Strategy for Multiple Alignment. Science 262 (1993) 208-214
2. Hertz, G.Z., Stormo, G.D.: Identifying DNA and Protein Patterns with Statistically Significant Alignment of Multiple Sequences. Bioinformatics 15 (1999) 563-577
3. Bailey, T.L., Elkan, C.: The Value of Prior Knowledge in Discovering Motifs with MEME. Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology. AAAI Press, Menlo Park, CA (1995) 21-29
4. Workman, C.T., Stormo, G.D: ANN-Spec: a Method for Discovering Transcription Factor Binding Sites with Improved Specificity. Pacific Symposium on Biocomputing. Stanford University, Stanford, CA (2000) 467-478
5. Pevzner, P.A., Sze, S.-H.: Combinatorial Approaches to Finding Subtle Signals in DNA Sequences. Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology. AAAI Press, Menlo Park, CA (2000) 269-278
6. Dai, X., Dai, Z.: Finding Transcription Factor Binding Sites Based on Multiple Pairwise Alignment. Proceedings of 2005 International Symposium on Intelligence Signal Processing and Communication Systems 577-580
7. Tompa, M., *et al*.: Assessing Computational Tools for the Discovery of Transcription Factor Binding Sites. Nature Biotechnology 23 (2005) 137-144
8. Sandelin, A., Alkema, W., Engstrom, P., Wasserman, W.W., Lenhard, B.: JASPAR: an Open-access Database for Eukaryotic Transcription Factor Binding Profiles. Nucleic Acids Research 32 (2004) 91-94

TSFSOM: Transmembrane Segments Prediction by Fuzzy Self-Organizing Map

Yong Deng

Zhejiang Police Vocational Academy, Hangzhou, China, 310018

Abstract. A novel method based on fuzzy Self-Organizing Map to detect the transmembrane segments, called TSFSOM, is presented in the paper. The multivariate "time" series of transmembrane proteins are classified by fuzzy Self-Organizing Map into five classes. Through the analysis of resulting trajectories on the map, frequent patterns of transmembrane segments are detected and even some kind of "new" knowledge about membrane insertion mechanism is obtained. The discovered patterns and the knowledge are then used to predict transmembrane segments for query sequence. The prediction results not only show that the method is powerful, but also prove that the patterns and the knowledge about the interaction between the patterns are effective and acceptable.

1 Introduction

Transmembrane proteins are important for a broad range of processes and functions in all biological systems and of the genomes sequenced thus far, about 20 percent to 30 percent of the gene products are predicted to be polytopic transmembrane proteins [1]. However due to their property unsuitable for X-ray diffraction or NMR studies, there is a large gap between the number of known sequences and the number of known structures. Efficient methods for membrane protein structure prediction are therefore needed. To solve this problem, a number of methods designed to locate the transmembrane regions and find the transmembrane orientation have been developed. The simplest and earliest scheme was based upon hydropathy analysis [2]. Hydrophobic segments with a score above a threshold value are considered as highly probable to be membrane-spanning. After the "positive-inside" rule was discovered [3], it was exploited to improve the prediction accuracy. Many methods [4, 5, 6, 7, 8, 9, 10, 11] have been proposed for transmembrane prediction.

Recently, soft computing technologies, such as fuzzy sets theory, and neural networks, are introduced to solve many problems such as target recognition and decision making. Researchers also find that soft computing technologies can also deal with transmembrane topology structure prediction due to its ability to model uncertain information [12, 13, 14, 15, 16].

The Self-Organizing Map (SOM) is a neural network algorithm based on unsupervised learning, which means that no previous knowledge about the characteristics of the input data is needed [17]. In our proposed TSFSOM, five primitive classes are visualized on the map and three frequent patterns of transmembrane segments are detected. The relationship between different patterns is explored. Subsequently, the patterns and the knowledge about the relationship between the patterns are used to predict the transmembrane segments. The prediction accuracy suggests that the patterns are effective

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 728–733, 2006.

and acceptable and TSFSOM is powerful to predict the topology of transmembrane proteins.

2 Structure of TSFSOM

TSFSOM has four steps. First, the multivariate "time" series for transmembrane proteins is generated based on different physiochemical characteristics of amino acids. Second, FSOM is trained and five primitive classes are determined by the U-matrix method. The amino acid sequences of membrane proteins are then converted to sequence of these five classes. Third, frequent patterns of transmembrane segments are detected and the interactions between the patterns are also studied. Finally, the discovered patterns and the knowledge about the interactions are used to predict transmembrane segments for query sequences fig (1).

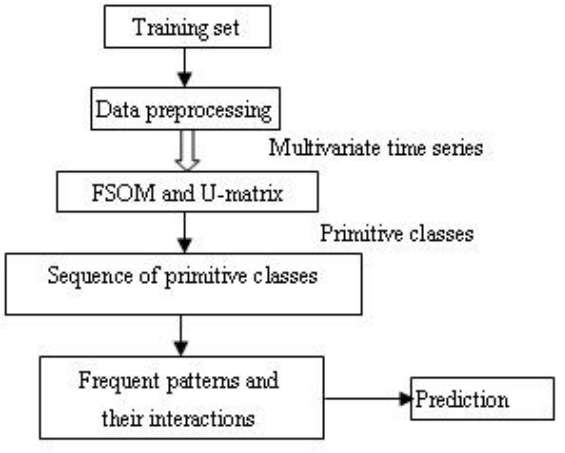


Fig. 1. The structure of TSFSOM

2.1 Data Processing

First, transmembrane protein sequences are converted to multivariate "time" series according to the amino acids' physiochemical properties. It is well known that transmembrane proteins have two common basic architectural principles due to the lipid environment in which they are embedded: 1) transmembrane -helices have a high overall hydrophobicity; 2) The charge distribution of the hydrophilic loops that connect the transmembrane segments follows the "positive-inside" rule. A set of 4 property scales is selected for data representation: 1) hydrophobicity; 2) polarity; 3) isoelectric point; 4) hydrophilicity. Then, four-dimensional "time" series $X_t = \{X_{1t}, X_{2t}, X_{3t}, X_{4t}\}$ is obtained for a transmembrane protein.

Additionally, to reduce the noise introduced by a single sequence, each data vector is computed by calculating the average property value of a continuous stretch of residues

(the window), plotting the value at the position of the center residue and subsequently, sliding the window over the sequence

$$x_{mt} = \sum_{j=t-(w-1)/2}^{t+(w-1)/2} p_m(j) \Big/ w$$

where w is the window size and $p_m(j)$ denotes the $m$here m=1,2,3,4 property value of the *jth* residue.A window of 11 residues is used here. Finally, the generated "time" series are sampled every 3 residues. To avoid the case that components with large variance shadow those with small variance, the components of the input data are scaled to have unit variance.

2.2 Primitive Classes Visualized from FSOM and U-Matrix

The traditional SOM basic algorithm belongs to a kind of hard partition methods. The aim of the algorithm is that the sets of objects are strictly grouped into clusters. However, all objects have not strict attributes and both attributes and characters are always fuzzy. Due to the reason that soft partition is proper in many real application systems, many fuzzy SOM (FSOM)have been proposed. In our proposed TSFSOM, the FSOM presented in [18] is used to find frequent patterns of transmembrane segments. Let M denote the number of input samples, N the number of input vector components, and K the number of output neurons. The learning algorithm consists of the following steps.

Step 1) Randomize the initial values of the components of the weight vectors.

Step 2) Input all samples $X_l = [X_{l,1}, X_{l,2}, \ldots, X_{l,n}]$, where $l = 1, 2, \ldots, M$

Step 3) Calculate the Euclidean distances from each sample X_l to all output neurons

$$d_{lj}(t) = \sqrt{\sum_{i=1}^{N} (X_{li} - W_{ij}(t))^2}$$

where $l = 1, 2, \ldots, M$, $j = 1, 2, \ldots, K$

Step 4) Compute the memberships of each sample to all neurons

$$R_{lj}(t) = \frac{\frac{1}{d_{lj}^2(t)}}{\sum_{i=1}^{K} \left(\frac{1}{d_{lm}^2(t)}\right)}$$

where $l = 1, 2, \ldots, M$, $j = 1, 2, \ldots, K$

Step 5) Adjust the weights of each neuron according to the computed memberships

$$W_{ij}(t+1) = W_{ij}(t) + \frac{\sum_{l=1}^{M} R_{lj}(t) \cdot (X_{li} - W_{ij}(t))}{\sum_{l=1}^{M} R_{lj}(t)}$$

Step 6) Determine the stability condition of the network

$$\max_{\substack{1 \le i \le N \\ 1 \le j \le K}} \{|W_{ij}(t+1) - W_{ij}(t)|\} < \varepsilon$$

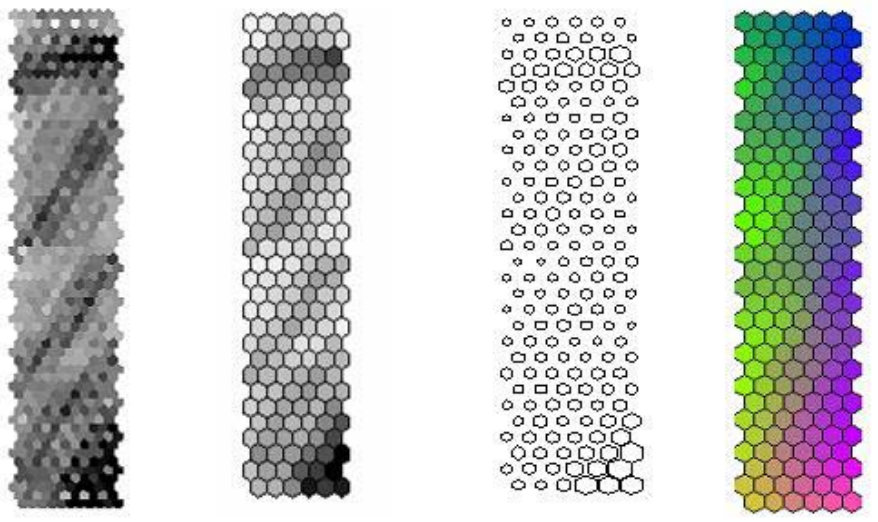


Fig. 2. The cluster structure of the FSOM is typically visualized using distance matrix techniques. (a)U-matrix visualization. (b)median distance matrix plotted by gray scale. (c) median distance matrix plotted by marker size.

3 Prediction

The prediction process has three steps. First, the query sequence is mapped into the trained SOM grid and the whole sequence is represented by the sequence of the five primitive classes. Second, the strong and middle hydrophobic patterns are detected and their distributions along the sequence are analyzed. If there is only one strong hydrophobic pattern and this pattern has a long distance from other detected middle patterns, the sequence is predicted to be a single transmembrane protein with strong hydrophobic pattern corresponding to transmembrane region. Otherwise, the sequence is considered to be a polytopic transmembrane protein with all detected strong and middle hydrophobic pattern corresponding to transmembrane segments. Finally, the trajectories around each assured transmembrane segment is scanned to find neighboring weak hydrophobic patterns (distance¡18) and these neighboring patterns are further affirmed to be transmembrane segments or not.

The prediction accuracy of TSFSOM is evaluated on the data set A created by European Bioinformatics Institute (EBI). Since the data set A contains membrane proteins with known three-dimensional structures, it is established to benchmark the performance of various methods. Sequences above 25 % identity are eliminated to perform an unbiased evaluation of the prediction accuracy. Therefore, the test set contains 30 membrane proteins with 91 transmembrane segments.

The efficiency of the transmembrane helix prediction was measured in terms of the following ratios:

M=Ncor/Nobs(single transmembrane helix sensitivity)
C=Ncor/Nprd(single transmembrane helix specificity)

where Nobs, Nprd and Ncor are the number of observed, predicted and correctly predicted transmembrane helices, respectively. The overall prediction power can be measured as the geometric mean of these ratios: . The prediction accuracy of TSFSOM is shown in Table 1, along with the results of other prediction methods. It can be seen that overall prediction power of TSSOM is as high as 95.4 %. It outperforms other prediction methods using single sequence information (TMHMM 92.2 %, HMMTOP 93.6 %, MEMSAT 92.2 %) and gains almost the same prediction accuracy as the methods using multiple sequence alignments information (TMAP 95.6 percent, PHDhtm 96.7 percent). However, the specificity of TSSOM is relatively low (91.9 %), which probably result from the incomplete discovery of the relationship between different patterns.

Table 1. Prediction accuracy of various methods on the data set A created by EBI

Data set	**Method**	**No of transmembrane helices**					
		$\mathbf{N}_{obs}$	$\mathbf{N}_{prd}$	$\mathbf{N}_{cor}$	**Qp(%)**	**M**	**C**
A	**TMHMM**	91	87	82	92.2%	90.1%	94.3%
	HMMTOP		95	87	93.6%	95.6%	91.6%
	MEMSAT		87	82	92.2%	90.1%	94.3%
	TMAP		91	87	95.6%	95.6%	95.6%
	PHDhtm		91	88	96.7%	96.7%	96.7%
	TSFSOM		99	93	95.4%	98.7%	91.9%

4 Conclusions

We suppose that the promising result of TSFSOM is due to: 1) Physiochemical properties of amino acids are successfully selected to describe the transmembrane protein sequence in terms of multivariate ”time” series. 2) SOM together with special visualization techniques is used to explore the elementary structures of multivariate ”time” series. 3) Frequent patterns of transmembrane segments are detected and relationships between different patterns are considered.

Acknowledgement

This work was supported by National Natural Science Foundation of China grant no. 30400067, Shanghai Nature Foundation grant no. 03ZR14065

References

1. Liu, Q., Chen, Z. Q., Wang, B.H., Zhu, Y.S., Li, Y.X.: Research on Several Prediction Methods of Membrane Protein Structure and Topology. High Tech. Lett. **7** (2001) 1-4
2. Kyte, J., Doolittle, R.F.: A Simple Method For Displaying the Hydropathic Character of a Protein. J. Mol. Biol. **157** (1982) 105-132
3. Von Heijine, G.: The Distribution of Positively Charged Residues in Bacterial Inner Membrane Proteins Correlates with the Transmembrane Topology. EMBO. J. **5** (1986) 3021-3027

4. von Heijine, G.: Membrane Protein Structure Prediction: Hydrophobic Analysis and the Positive-inside Rule. J. Mol Biol **225** (1992) 487-494
5. Persson, B., Argos, P.: Prediction of Transmembrane Segments in Proteins Utilizing Multiple Sequence Alignments. J. Mol. Bio. **273** (1994) 182-192
6. Persson, B., Argos, P.: Topology Prediction of Membrane Proteins. Protein Sci. **273** (1994) 182-192
7. Jones, D. T., Taylor, W. R., Thornton, J. M.: A Model Recognition Approach to the Prediction Of All-helical Membrane Protein Structure and Topology. Biochemistry **33** (1994) 3038-3049
8. Deng, Y., Liu, Q.: An Improved Optimal Fuzzy Information Fusion Method and Its Application in Group Decision. J Comp. Sys. Sci. Int. **44** (2005) 531-541
9. Deng, Y., Liu, Q.: A TOPSIS-based Centroid-index Ranking Method of Fuzzy Numbers and Its Application in Decision-making. Cyber. Sys. **36** (2005) 581-595
10. Liu, Q., Zhu, Y.S., Wang, B.H., Li, Y.X.: A HMM-based Method to Predict the Transmembrane Regions of Beta-barrel Membrane Proteins. Computational Bio. and Chemistry **27** (2003) 69-76
11. Deng, Y., Liu, Q., Li, Y.X.: Scoring Hidden Markov Models to Discriminate Beta-baffel Membrane Proteins Volume. Computational Bio. and Chemistry **28** (2004) 189-194
12. Rost, B., Casadio, R., Fariselli, P., Sander C.: Transmembrane Helices Predicted at 95 Percent Accuracy. Protein Sci. **4** (1995) 521-533
13. Rost, B., Casadio, R., Fariselli, P., Sander C.: Prediction For Helical Transmembrane Proteins at 86 Percent Accuracy. Protein Sci. **5** (1996) 1704-1718
14. Deng, Y., Liu, Q., Li, Y.X.: Prediction Of Transmembrane Segments Based On Fuzzy Cluster Analysis Of Amino Acids. Acta Chimica Sinica **62** (2004) 1968-1976
15. Jacoboni, I., Martelli, P.L., Fariselli, P., De, Pinto, V., Casadio, R.,: Prediction of the Transmembrane Regions of Beta-barrel Membrane Proteins with a Neural Network- Predictoremph. Protein Sci. **10** (2001) 779-787
16. Gromiha, M.M., Ahmad, S., Suwa, M.: Neural Network-based Prediction of Transmembrane Beta-strand Segments in Outer Mmembrane Proteins. J.Comp. Chem.**25** (2004) 762-767
17. Kohonen, T.: Self-Organizing Maps. Springer, Berlin Heidelberg (1995)
18. Hu, W.M., Xie, D., Tan, T.L., Steve, M.: Learning Activity Patterns Using Fuzzy Self-Organizing Neural Network. IEEE Trans. on Sys. Man Cyber. B **34** (2004) 1618-1626

Analysis of Multifibre Renal Sympathetic Nerve Recordings

Dong Li[1], Yingxiong Jin[1], Zhuo Yang[2], and Tao Zhang[1]

[1] Key Lab of Bioactive Materials,
Ministry of Education and College of Life Science,
Nankai University, Tianjin 300071, P.R. China
zhangtao@nankai.edu.cn
[2] College of Medicine Science, Nankai University,
Tianjin 300071, P.R. China
zhuoyang@nankai.edu.cn

Abstract. Multifibre renal sympathetic nerve activity (RSNA) recordings represent a nonlinear dynamic system with high dimensionality. In this paper, an effort has been made to effectively remove noises and reduce the dynamics of the multifibre RSNA signals to a simpler form. For this purpose, an improved cluster method combined with the wavelet-transform-based denoising approach is proposed. The outcomes of the present work show that wavelet denoising approach is a useful tool for analyzing multifibre RSNA in rats. Furthermore, compared to the original algorithm of the cluster method, the improved one reduces some aspects of bias.

1 Introduction

Activity within the sympathetic nerves to the kidney is determined by integration within the hypothalamic areas of the brain of sensory information, arising mainly from the high-and low-pressure cardiovascular baroreceptors with contributions from the somatosensory and visceral systems and the higher cortical areas of the brain [1]. Moreover, the basal level of activity in any particular multifibre nerve preparation varies from moment to moment, reflecting the ability of the central nervous system to formulate complex patterns of sympathetic outflow.

It has become apparent that the impact of renal sympathetic nerve activity (RSNA) on the kidney function may not be determined solely by the absolute level of activity within the nerve signal, but more by how the pattern of the energy is distributed across the signal and how it changes dynamically [2]. Studies have shown that, in the renal sympathetic nerve, major energy peaks occur at heart rate (HR) frequency, respiratory frequency and also at very low frequencies [3, 4, 5]. Furthermore, most frequently the discharges of postganglionic sympathetic fibres are synchronized into bursts related to the cardiac cycle. However theoretically, power spectral analysis in the frequency domain fails to provide useful information about nonlinear characteristics of the time series, especially for detecting any nonlinear changes of the underlying network, which may be activated.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 734 – 739, 2006.

An alternative approach is that of time domain analysis, in which fluctuations in energy over time are assessed by the application of nonlinear dynamical theory. However, our previous study [1] suggested that the multifibre RSNA may represent a nonlinear dynamical system with high-dimensionality. Estimating the nonlinear features of such a high dimensional dynamical system is extremely difficult. Moreover, the multifibre renal nerve signals inevitably contain noise components, which sometimes lead to incorrect estimation when using nonlinear analysis. Therefore, heavily contaminated multifibre RSNA are hard to analyze quantitatively. Recently, a method based on the wavelet transform, namely wavelet denoising, has been developed and successfully applied to the analysis of various physiological time series. It allowed the visualization of multifibre sympathetic nerve activity and a further quantitative analysis of their variability. In the present study we introduced an improved cluster method combined with the wavelet-transform-based denoising approach, by which the dynamics of the multifibre sympathetic traffic signal was reasonably reduced to a simpler form that makes it possible to examine some essential aspects of the overall behaviour of the underlying dynamic system.

Furthermore, the sample entropy (SampEn) [6] method has been applied in examining the changes of complexity or irregularity of time series; including raw and denoised RSNA. The SampEn provides an alternative approach for measuring nonlinearity of the dynamical system based on multifibre nerve recording, especially for short-term signals. A lower entropy value always indicates more regularity or less complexity. If this is the case then it is expected that the entropy values should be changed before and after filtered multifibre renal nerve recordings.

2 Method

The experiments were approved by the local ethical committee of Nankai University.

2.1 Animal Experiment

The experiments were performed on 14 male Wistar rats, 285±3.25g, anaesthetized with a mixture of urethane and chloralose. The left kidney was exposed retroperitoneally and a branch of the nerve to the kidney was dissected free and placed on a bipolar electrode made from silver wire. The nerve signals were amplified, filtered and displayed on an oscilloscope and stored for later analysis, using a PowerLab data acquisition system. A 40s high frequency (1 KHz) sampling of RSNA was taken at baseline condition.

2.2 Wavelet-Transform-Based Denoising Approach

Wavelet transform (WT) has been successfully applied to denoising since it can decorrelate random processes into nearly independent coefficients. The complete signal representation by wavelet transform domain maxima was introduced [7]. It was able to distinguish edge maxima from noise maxima by analyzing the singularity properties of wavelet transform domain maxima of a signal across the various scales [8].

Furthermore, WT can pack most of the signal energy into a few significant coefficients and relates the insignificant coefficients to the signal-independent additive noise. A threshold is set to distinguish noise from the structure information in threshold-based denoising schemes. It can be classified into soft and hard ones, in which coefficients less than the threshold will be set to 0 but those above the threshold will be preserved or shrunk. Donoho [9] developed the approach named the *WaveletShrinkage* scheme with a universal threshold based on orthonormal wavelet bases. The method, developed by Xu *et al*, was applied in the present study [10].

2.3 An Improved Cluster Method

The Cluster algorithm examines sequential groups of samples (clusters) of synchronized RSNA voltages. The first cluster of samples is defined as a possible nadir and the second as a possible peak. After all significant increases were marked; the series was rescanned in consecutive order to search for significant decreases. The occurrence of a peak in synchronized RSNA is defined as a significant increase followed by a significant decrease with a nadir on each side.

Two improvements of the computerized peak detection algorithm, based on the cluster method, have been made. Firstly, the Student t-test was appropriately applied in searching for significant increases by comparing two sequential clusters. However, in the original program the value of standard deviation for the t-test was improperly fixed to 0.0001 for every comparison. Secondly, the present program was set to read into all input data at once rather than part of the data each time. Such an improvement saved program-running time and increased the efficiency of program performing.

2.4 Sample Entropy Analysis and Statistics

The improved algorithm of ApEn, SampEn statistics, agree much better than ApEn statistics with the theory of random numbers with known probability characteristics, over a broad range of operating conditions and maintain relative consistency where ApEn statistics do not. The mathematical details of the SampEn are referred to the reference [6].

All the data are expressed as the mean ± SEM. Analysis of variance (ANOVA) was used for statistical analysis of the data, allowing within and between task comparisons to be made and significant differences are presented when $P < 0.05$.

3 Results

Figure 1a presents an original neurogram obtained from a multifibre recording of the RSNA made from one Wistar rat with a 1 KHz sampling frequency and a 2 second sampling time; Fig. 1b shows a denoised neurogram of the RSNA made from the same a Wistar rat and the wavelet-transform-based denoising approach has been employed for removing noise from the original signals; Fig. 1c gives the integrated (with 20 ms time constant) RSNA and the improved Cluster method has been used for simplifying high-dimensional chaotic dynamical system.

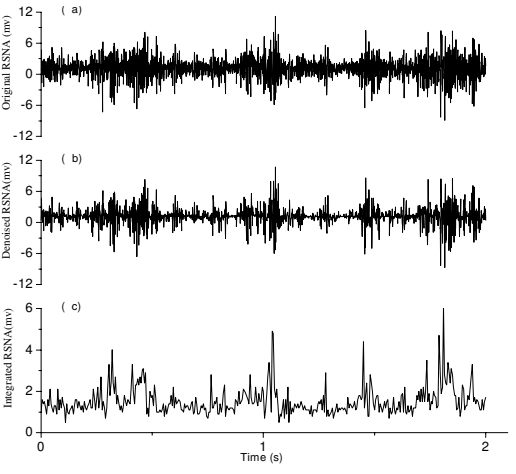


Fig. 1. (a) Original neurogram of renal sympathetic nerve activity (RSNA) from one Wistar rat using 1kHz-sampling frequency with a 2 s time period. (b) Denoised neurogram of the RSNA using wavelet-transform-based denoising approach. (c) Integrated (with a 20 ms time constant) RSNA.

Table 1. Average value of peak characteristics between original and Denoised RSNA

n =14	No peaks	Mean PI	Mean PA	Mean PW
Original RSNA	191±13	220±17	6.48±1.0	133±10
Denoised RSNA	152±13**	286±26**	6.20±1.1	162±11

Values are mean±S.E.M., (n=number of rats). No peaks: Number of Peak; Mean PI: Mean of peak to peak interval（ms）; Mean PA: Mean of peak amplitude（mv）; Mean PW: Mean of peak width（ms）. Statistical comparison is within groups comparing the raw renal nerve activities to filtered RSNA using wavelet-based transform denoised approach (**$p<0.01$). Signal sample frequency is 1 KHz and the length of each data file is about 40 s.

The total number of original RSNA synchronized peaks detected by the improved Cluster method ranged from 267 to 109 in individual animals, with a mean value of 191±13 (Table 1, n=14) for the sampling time of 40 s. After the noise of multifibre recordings was efficiently removed by the wavelet-based transform denoised approach the RSNA synchronized peaks averaged 152±13, which was significantly less than that obtained in the raw data ($p<0.01$). In contrast to the peak number, it can be seen that the averaged value of peak-to-peak intervals for the denoised RSNA (286±26) was statistically ($p<0.01$) higher than that (220±17) of the raw renal signals ($p<0.01$) in Table 1. However, there was no significant difference between the original and denoised RSNA for either mean peak amplitude (6.48±1.0 *vs* 6.20±1.1) or mean peak width (133±10 *vs* 162±11).

Complex analysis of the oscillators of the RSNA signal using sample entropy measurement provides the evidence that the values did significantly change after filtering the multifibre recordings of RSNA by wavelet-based denoised approach. Figure 2 presents the curves of the sample entropy values obtained from renal sympathetic nerve signals in fourteen rats. Each sample on the X-axis shows the values of entropy measurement in both original and denoised RSNA. It can be seen that the original data curve (Fig 2, line with square symbol) was significantly higher ($p<0.001$) in all samples compared with those obtained after denoised signals (Fig 2, line with circle symbol).

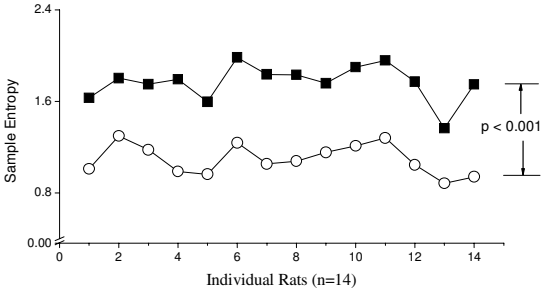


Fig. 2. Individual values for the sample entropy (vertical axis) obtained from 14 rats (each shown on horizontal axis). Series A, original RSNA (■); series B denoised RSNA (○). m=2 (value of embedding dimension) and r=0.25 (value of band width) were used during entropy measurement.

4 Discussions

Little attention has been paid to the multifibre recordings of sympathetic outflow. Since the computerized peak detection algorithm was proposed by Malpas and Ninomiya [11], several papers have reported on the application of the algorithm to synchronize RSNA. Compared to the original one, the improved algorithm reduces some aspects of bias, which make it much smoothly and efficiently.

The main advantage of the wavelet transform is its variable window size, thus leading to an optimal time-frequency resolution adapted to each frequency range. Based on the wavelet transform, this denoising method allowed a better visualization of synchronized sympathetic discharges (Fig.1b) compared to that of the original traces of nerve activity (Fig.1a). We should remark that this is usually difficult to achieve with standard approaches due to the fact that multifibre RSNA represent non-stationary signals with noise. It was evident from Fig.2 that the entropy values did reduce after applying the wavelet-based denoised filter, indicating the noisy components within the nerve activity were effectively removed.

Finally, the SampEn measurement has been proposed at the present study. It was evident that the length of peak numbers was greatly dependent on the sampling period of sympathetic nerve signals. It is a fact that less than one thousand data points of the above peak sequences are sometimes available for further nonlinear analysis after applying the peak detection algorithm, as a consequence the application of chaotic analysis programs has been questioned its validity debated because of the limited

length of data points. Therefore, SampEn could become an alternative nonlinear approach for analyzing short-term peak sequences.

In conclusion, an improved cluster method combined with the wavelet-transform -based denoising approach presented in this study allows for measuring multifibre sympathetic nerve signals with much better accuracy.

Acknowledgements

This work was supported by grants from the NSFC (30370386, 304704530), TMSTC (043611011).

References

1. Zhang, T., Johns, E.J.: Chaotic Characteristics of Renal Nerve Peak Interval Sequence in Normotensive and Hypertensive Rats. Clin. Exp Pharmacol Physiol 25(11) (1998) 896-903
2. Stauss, H.M., Persson, P.B., Johnson, A.K., Kregel, K.C.: Frequency-Response Characteristics of Autonomic Nervous System Function in Conscious Rats. Am. J. Physiol 273 (1997) H786-95
3. Persson, P.B., Ehmke, H., Kirchheim, H.R.: Autoregulation and Nonhomeostatic Behavior of Renal Blood Flow in Conscious Dogs. J. Physiol-London 462 (1993) 261-273
4. Zhang, T., Johns, E.J.: Somatosensory Influence on Renal Sympathetic Nerve Activity in Anaesthetised Wistar and Hypertensive Rats. Am. J. Physiol. 272 (1997) R982 - R990
5. Yang, Z., Zhang, T., Coote, J.H.: Synchrony Analysis between Blood Pressure and Sympathetic Nerve Signal Inhibited by Atrial Receptor Stimulation in Wistar Rats. Exp Physiol 87 (2002) 461-468
6. Richman, J.S., Moorman, J.R.: Physiological Time-Series Analysis Using Approximate Entropy and Sample Entropy. Am J Physiol 278 (2000) H2039-H2049
7. Mallat, S.: A Theory for Multiresolution Signal Decomposing: the Wavelet Representation. IEEE Trans Pattern Anal Machine Intell 11(7) (1989) 674-693
8. Mallat, S., Hwang, W.L.: Singularity Detection and Processing with Wavelet. IEEE Trans Inform Theory 38(2) (1992) 617-643
9. Donoho, D.L.: De-Noising by Soft-Thresholding: IEEE Trans on Inform Theory 41(3) (1995) 613-627
10. Xu, Y., Weaver, J.B., Healy, D.M., Jr and Lu, J.: Wavelet Transforms Domain Filters: A Spatially Selective Noise Filtration Technique. IEEE Transaction on Image Processing 3(6) (1994) 747-758
11. Malpas, S.C., Ninomiya, I.: A New Approach to Analysis of Synchronized Sympathetic Nerve Activity. Am. J. Physiol 263 (1992) H1311-H1317

A New Color Blindness Cure Model Based on BP Neural Network*

Yu Ma, Xiao-Dong Gu, and Yuan-Yuan Wang**

Department of Electronic Engineering, Fudan University, 200433, Shanghai, P.R. China
mayu82@163.com, guxiaodong@263.net, yywang@fudan.edu.cn

Abstract. There is no method of color blindness cure so far although some research on color blindness has been done. In this paper we propose the BP neural network model for Color Blindness Cure (CBC) according to the biological experimental results. This model transforms abnormal output signals of two paths corresponding to two kinds of cones of the dichromats into normal ones of three paths corresponding to three kinds of cones of normal people. Computer simulation results show that this model has good performance and changes the abnormal vision of the dichromats into normal ones. In addition, some improvements of our model are introduced.

1 Introduction

1.1 Human Color Vision and Color Blindness

In human vision, the visual signal is received by rods and cones in the retina [1] [2]. Rods are sensitive to the luminance while cones convey color information. Three types of cones named L cones, M cones, and S cones are identified based on the spectral absorption characteristics [3] in Fig.1. The signals produced by the cones are passed in the retina and then arrive at the lateral geniculate nucleus (LGN).

Color blindness is a kind of vision disease. About 8% of men and 0.5% of women have hereditary color defects and a quarter of them are dichromats (a kind of color blindness sufferers who confuse some colors). Losing or abnormality of cones is the main reason of color blindness. There are three types of dichromats according to the type of the abnormal cones: protanopes, deuteranopes, and tritanopes.

According to three-color theory, some models that focus on the research of dichromats' vision characteristics have been built up, such as "multilayered neural network models" [4] and "fuzzy connectionist model" [5] etc. These models can be used to analyze or simulate the color information processing mechanism of nervous system. However, no valid color blindness cure method has been introduced.

Considering the situation of losing of cones, the dichromats lose one kind of cones but the neurons which receive the output signals from those cones may be normal though they do not receive the correct inputs. If we can simulate the output signals of the lost kind of cones while adjusting the output signals of the other two kinds of

* This work is supported by National Basic Research Program of China (2005CB742303).
** Corresponding author.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 740 – 745, 2006.

cones at the same time so as to convey the similar signals like the normal people's, the dichromats may recover the normal vision. Different kinds of dichromats may have different quantity and proportion of cones [6] so it is not practicable to use the same transfer function for every dichromats.

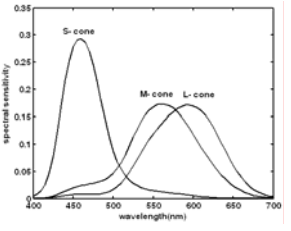


Fig. 1. The spectral sensitivities of three kinds of cones [3]

1.2 BP Neural Networks (BPNN)

The design of artificial neural networks is motivated by the research on the brain, which can process complex information fast and powerfully [7]. Different types of neural networks are widely used in many fields. Research on neural networks also has important significance of discovering how biological vision system operates. BPNN which is a multilayer perceptron with back-propagation learning has good performance in nonlinear mapping and generalization. It has simple structure and is easy to implement by hardware so it could be used in the vision system with huge information. We use the BPNN model to transform the output signals of two kinds of cones into three kinds and supply the probability for the cure of color blindness.

2 Color Blindness Cure (CBC) Model Using BPNN

To simulate the human eyes model, when a picture inputs we change the color of each pixel into output signals of three kinds of cones (for dichromats there are two kinds). This is performed by a transformation matrix from RGB color space to cones'

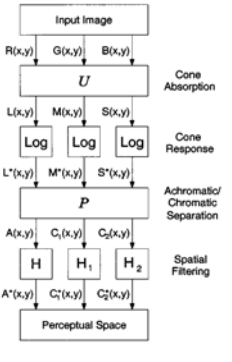


Fig. 2. Faugeras model of the human color vision system [8] [9]

absorption space as the Faugeras model (Fig. 2) of human color vision system [8] did. For the D6500 white light the transformation matrix U [8] is given as

$$U=\begin{bmatrix} 0.3634 & 0.6102 & 0.0264 \\ 0.1246 & 0.8138 & 0.0616 \\ 0.0009 & 0.0602 & 0.9389 \end{bmatrix}. \quad (1)$$

So we can get the output signals of the cones from the RGB values of the images.

2.1 Structure of Our CBC Model

In the simulation, we use the output signals of two kinds of cones transformed from the RGB values of the image as the inputs of the neural network while the training object is the output signals of normal people's three kinds of cones. The BP learning method is chosen and there is one hidden layer in the neural network. The model can be described by three formulas as:

$$L' = F(A, B),\ M' = G(A, B),\ S' = H(A, B). \quad (2)$$

In the formulas, A, B mean the output signals of the two kinds of cones of the dichromats (They are chosen from the L, M, S according to the type of the dichromats). L', M', S' mean the outputs of the neural network which are expected to be similar with the object L, M, S. And F, G, H mean the equipollent transfer functions of the model. The structure of the model is shown in Fig. 3.

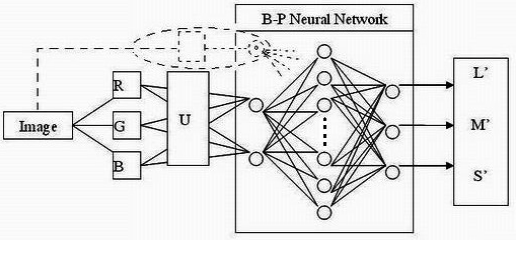


Fig. 3. The structure of the model (The part illustrated by dashed lines is not for this model and will be mentioned afterward)

We choose many pictures which contain enough color information to train the model and some other pictures to test the results. We can not only compare the output of the model but also draw the "output pictures" by doing the inverse transform from LMS space to RGB space. In Fig .4 we can see the testing results.

This model can help the dichromats distinguish some confused colors but there is still some confusion (we can see the color bar in Fig .4). To solve this problem, one probable method is to choose more natural images to train the model so as to make the model more sensitive to the colors which appear more in the nature. The effect is to change the "color blindness" space into smaller or less important space.

Fig. 4. (a) The original image. (b) The simulation image of deuteranopes (All images of dichromats in this paper are obtained by the transform from the outputs of the cones to RGB values for simple without considering the real color perceptron of dichromats. More research on dichromats' actual vision can be seen in [10]). (c) The recovered image using our model**.** (d) The original color bar. (e) The simulation color bar of deuteranopes. (f) The recovered color bar using the model. It shows both the effect and some confusion of the colors.

2.2 Improvements of Our Model

To improve the model, a "man-made cone input" is added. Considering the biological practicability we use the luminance as the third input of the neural network and it is one kind of combination of the RGB value. In human vision system there are rods which transfer the luminance information so their output signals could be used.

The structure of the improved model is similar to the foregoing one and it has one more input in the neural network as shown in the dashed-line part of Fig. 3. The results in Fig .5 show that it has better performance than the first model.

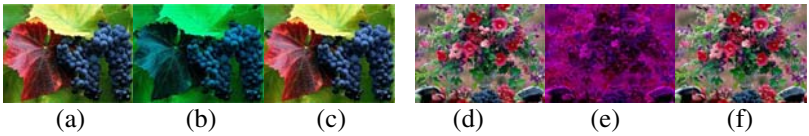


Fig. 5. This figure shows the effect of the improved model for the recovering of the colors for dichromats. (a), (d) are the original images. (b), (e) are the simulation images of protanopes and deuteranopes respectively. (c), (f) are the recovered images by our improved model.

3 Results and Discussion

3.1 The Performance of the Model

After training, the neural network can get the fixed weight vectors to transform the output signals of the two kinds of cones into three kinds. We use more than one hundred nature pictures to test the performance of the models for three different kinds of dichromats. The visualized effect is shown in the figures and we also compare the output of the model with the object by computing the relative error of the output signals to estimate the model. Experimental results show that the model performs well in recovering different kinds of images.

In the experiments we found that the number of the hidden layer neurons of the neural network is an important parameter. In the same training and testing condition, different relative errors according to different number of hidden neurons are listed in Table 1. About 7 to 10 neurons will be enough and satisfying with the differences seen by eyes between original images and recovered ones being acceptable.

The selection of training samples is also important. A small number of samples may not stand for great variety of colors in nature so the neural network may not be trained well and have bad performance in the testing set. However, too many samples may lead to "overfitting" or "overtraining" [7]. It is a worthy task to be studied.

Table 1. The relative errors of the model's output with different number of neurons

Number of hidden neurons	3	4	5	6
Relative error	0.4325	0.1094	0.0928	0.0828
Number of hidden neurons	7	8	9	10
Relative error	0.0564	0.0447	0.0341	0.0338

3.2 Practicability of the Model

Our color blindness cure model has a good effect in the simulation and might be used for the cure of dichromats. But in practice, there are still many problems. We need to know how to stimulate the receivers which accept the output signals of the cones. This is a challenge to the scientists of neurobiology and medicine.

Another problem is that we not only need to stimulate the receiver without input but also need to change the inputs of the other two kinds of receivers. Here we propose some methods. The simplest method is to cut off the connecting paths between the cones and the receivers while adding the output signals of the model directly, but this method is invasive and dangerous. Another method is computing the differences between the output signals we get from the model and the original output signals so that we only need to make up the differences instead of erasing the original output signals of the existing cones. In fact there is a more practical method that we can do some processing before the colors are received by the eyes so that the output signals of the two existing kinds of cones that dichromats have may be similar as those of the normal people. After preprocessing, only one kind of stimulus need to be artificially made and the other two kinds have been adjusted before light comes into the eyes. It may be implemented by a system such as a specific pair of glasses whose parameters are calculated by the transfer functions of the model.

Now we are taking part in the "Basic Principle and Key Problems of Vision Recovering" project supported by "National Basic Research Program". Our work in this paper is also related to vision recovering so it is possible to do more research on the practicability of the model with the good experimental conditions.

4 Conclusions

Losing of cones and abnormality of cones are main reasons of color blindness. From the spectral sensitivity of cones we can see that it is possible to change the two kinds of cones' output signals of dichromats into three kinds as normal people have, and a nonlinear neural network could be used to accomplish the complex transform. In this paper the BP neural network model for the vision recovering of dichromats was constructed. It was shown that the model did well in the simulation for the cure of color blindness and correlated well with the biological characteristics of human

vision. We also further improved our model and listed some useful discussions on the model's practicability for color blindness cure. With the support of the "National Basic Research Program", a lot of further work could be done in the future, such as the study of the structure and performance of the model, and the hardware implementation for practical application.

Acknowledgement

This work is supported by National Basic Research Program of China (which is also called'973'Program). The number is 2005CB724303.

References

1. Wandell, B.A.: Foundations of Vision. Sinauer Associates Inc. Publishers, Sunderland, Massachusetts (1995)
2. Hubel, D.H.: Eye, Brain, and Vision. Scientific American Library Series, New York (1988)
3. Smith, V.C., Pokorny, J.: Spectral Sensitivity of the Foveal Cone Photopigments between 400 and 500 nm. Vision Research 15(2) (1975) 161-171
4. Nakauchi, S., Usui, S.: Multilayered Neural Network Models for Color Blindness. Neural Networks 1991, IEEE International Joint Conference 1 (1991) 473-478
5. Kuo, Y.H., Hsu, J.P.: MCFC-R: A Fuzzy Connectionist Model for Color-Blindness Plate Recognition. Neural Networks 1996, IEEE International Conference 2 (1996), 718-723
6. Vimal, R., Pokorny, J., Smith, V., Shevell, S.: Foveal Cone Threshold. Vision Research 29 (1989) 61-78
7. Simon, H.: Neural Networks-A Comprehensive Foundation. 2nd edn. Prentice Hall (1999)
8. Martin, C.E., Keller, J.G., Rogers, S.K., Kabrisky, M.: Color Blindness and a Color Human Visual System Model. IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans 30(4) (2000) 494-500
9. Faugeras, O.D.: Digital Color Image Processing and Psychophysics within the Framework of a Human Visual Model. Ph.D. dissertation, Univ. Utah, Salt Lake City, (1976)
10. Brettel, H., Vienot, F., Mollon, J.: Computerized Simulation of Color Appearance for Dichromats. Journal Optical Society of America 14 (1997) 2647-2655

Design of RBF Network Based on Fuzzy Clustering Method for Modeling of Respiratory System

Kouji Maeda, Shunshoku Kanae, Zi-Jiang Yang, and Kiyoshi Wada

Department of Electrical and Electronic Systems Engineering,
Graduate School of Information Science and Electrical Engineering,
Kyushu University,
Hakozaki 6-10-1, Higashi-ku, Fukuoka, 812-8581 Japan
jin@ees.kyushu-u.ac.jp

Abstract. Pulmonary elastance provides an important basis for deciding air pressure parameters of mechanical ventilators, and airway resistance is an important parameter in the diagnosis of respiratory diseases. The authors have proposed a second order nonlinear differential equation model of respiratory system whose elastic and resistant coefficients are expressed by RBF networks with the lung volume as the input. When we use RBF networks expression, numerical stability can be expected, because the output of each node is in range of [0,1], the balance between each node is good. However, the problems of deciding the number of nodes and the center/deviation of each node were remained. In this paper, a design method of RBF network based on fuzzy clustering method is proposed to decide center and deviation of each node. By means of fuzzy clustering, the available data set is partitioned into fuzzy subsets so that each RBF works effectively. The proposed method is validated by examples of application to practical clinical data.

1 Introduction

When a patient cannot autonomously breathe due to some accident or disease, the patient has to receive artificial respiration treatment. Artificial respirators are popular and important medical equipment. When the artificial respirator is used, it is necessary to appropriately set the parameters of the respirator such as pressure and period of the respiration to be suitable for each patient. For setting appropriate ventilation conditions fitting to each patient, it is important to establish a mathematical model describing the mechanism of the human respiratory system, and to know the pulmonary characteristic of each patient via identification of the model.

For this purpose, a respiratory system model have been proposed by the authors. This model is expressed as a second order nonlinear differential equation whit volume variant elastic coefficient and volume variant resistant coefficient. In this model, elastic coefficient and resistant coefficient are expressed by RBF networks with the lung volume as the input[1]. Proposal of this model provides

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 746–753, 2006.

a possibility of estimating pulmonary elastance and airway resistance from the measurements of air pressure, flow and volume at ventilator side, no need for any sensor inserted into body inside. When we use RBF networks expression, numerical stability can be expected, because the output of each node is in range of [0,1], the balance between each node is good. However, the problems how to decide the number of nodes and how to decide center and deviation of each node were remained.

In this paper, a design method of RBF network based on fuzzy clustering is proposed to decide center and deviation of each node. The proposed method is validated by some examples of application to practical clinical data.

2 RBF Network Expression Based Model

In this paper, a case of mechanical ventilation is considered in which case spontaneous breathing is absent. In the inspiration phase of mechanical ventilation, the air pressure of ventilator side is higher than the lung inside, so the fresh air is sent to lung by the difference of air pressure, and the lung is expanded. In the expiration phase, the exhaust after gas exchange is excreted naturally by shrinking force of lung.

The expansion and the shrinkage relying on variation of air pressure are characterized by pulmonary elastance and airway resistance. The medical knowledge and clinical data clearly show that there is nonlinearity in elastance of lung and resistance of the airway. And they are important factor in decision of respiratory dynamics.

Comprehensively considering the above each factor, a new respiratory model of lung has been proposed as the following:

$$P(t) + a\dot{P}(t) = E(V)V(t) + R(V)\dot{V}(t) + h\ddot{V}(t) + \epsilon(t), \tag{1}$$

$$E(V) = \sum_{i=1}^{n} b_i \psi_i(V(t)), \tag{2}$$

$$R(V) = \sum_{i=1}^{n} c_i \psi_i(V(t)), \tag{3}$$

where, $V(t)$ is the volume of lung, $\dot{V}(t)$, $\ddot{V}(t)$ the first and second order derivatives of the volume, $P(t)$ the pressure of the airway, $\dot{P}(t)$ the first order derivative of the pressure. a, h are the coefficients of the model. $\epsilon(t)$ contains model errors and measurement noises.

As mentioned above, pulmonary elastance $E(V)$ and airway resistance $R(V)$ are not constant, they are some nonlinear function of volume $V(t)$. In this model, pulmonary elastic coefficient and airway resistance coefficient are described by RBF network expressed by (2) (3) which input is the volume of lung. Here, n is the number of nodes, and b_i, c_i $(i = 1, \cdots, n)$ are weights of i-th node. The $\psi_i(V)$ is Radial Basis Function with center of V_{0i}, deviation of σ_i:

$$\psi_i(V) = \exp(-\frac{(V - V_{0i})^2}{2\pi\sigma_i^2}), \quad i = 1, \cdots, n. \tag{4}$$

In next section, we will present a method how to decide center V_{0i} and deviation σ_i of each node.

Considering the relationship $Q(t) = \dot{V}(t)$ between $Q(t)$ and $V(t)$, the respiratory model (1) can be written as:

$$P(t) + a\dot{P}(t) = E(V)V(t) + R(V)Q(t) + h\dot{Q}(t) + \epsilon(t). \tag{5}$$

Substituting (2) and (3) for (5), we have

$$P(t) = -a\dot{P}(t) + V(t)\sum_{i=1}^{n} b_i\psi_i(V) + Q(t)\sum_{i=1}^{n} c_i\psi_i(V) + h\dot{Q}(t) + \epsilon(t). \tag{6}$$

Setting data vector $\boldsymbol{\varphi}(t)$ and parameter vector $\boldsymbol{\theta}$ in above model as

$$\boldsymbol{\varphi}(t) = [\dot{P}(t)|V(t)\psi_1(V)\cdots V(t)\psi_n(V)|Q(t)\psi_1(V)\cdots Q(t)\psi_n(V)|\dot{Q}(t)]^T,$$
$$\boldsymbol{\theta} = [-a|b_1\cdots b_n|c_1\cdots c_n|h]^T,$$

then a compact expression of the model is obtained:

$$P(t) = \boldsymbol{\varphi}^T(t)\boldsymbol{\theta} + \epsilon(t). \tag{7}$$

3 Design of RBF Network Based on Fuzzy Clustering Method

In RBF network, it is very important to optimize center and deviation of RBF because of the learning accuracy. Data clustering is a usual approach to decide RBF structure, and fuzzy clustering method is considered more effective than crisp clustering method. Here, we adopt GK algorithm[2] that is regarded as fuzzy version of c-means method to classify input data and to optimize the RBF. As the clustering results, we can obtain cluster prototypes and membership functions which are represented in the partition matrix. By using these two clustering results, center and deviation of each node are decided so that each RBF effects only to some limited input space.

The measurement data that are obtained from the mechanical ventilation system are only the sampled data of pressure $\boldsymbol{P}$, flow $\boldsymbol{Q}$, and volume $\boldsymbol{V}$.

$$\boldsymbol{P} = \begin{bmatrix} P(1) \\ P(2) \\ \vdots \\ P(N) \end{bmatrix}, \quad \boldsymbol{Q} = \begin{bmatrix} Q(1) \\ Q(2) \\ \vdots \\ Q(N) \end{bmatrix}, \quad \boldsymbol{V} = \begin{bmatrix} V(1) \\ V(2) \\ \vdots \\ V(N) \end{bmatrix}. \tag{8}$$

To classify the measurement data, we define measurement date vector $\boldsymbol{z}_k$ as $\boldsymbol{z}_k = [P(k), Q(k), V(k)]^T$, $\boldsymbol{z}_k \in \mathbf{R}^3$, k denote the time instant, $1 \leqslant k \leqslant N$. Then, the set of N observations can be denoted by $\boldsymbol{Z} = \{\boldsymbol{z}_k | k = 1, 2, \cdots, N\}$. $\boldsymbol{Z}$ is a $3 \times N$ matrix:

$$\boldsymbol{Z} = [\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_N] = \begin{bmatrix} z_{11}\ z_{12} \cdots z_{1N} \\ z_{21}\ z_{22} \cdots z_{2N} \\ z_{31}\ z_{32} \cdots z_{3N} \end{bmatrix}, \tag{9}$$

$$\boldsymbol{z}_k = [z_{1k},\ z_{2k},\ z_{3k}]^T, \quad (k = 1, 2, \cdots, N).$$

The objective of clustering is to partition the data set $\boldsymbol{Z}$ into c clusters. Fuzzy partition of $\boldsymbol{Z}$ is a family of fuzzy subsets $\{B_i | 1 \leqslant i \leqslant c\}$. The subsets are defined by their membership (characteristic) functions, represented in the partition matrix $\boldsymbol{U} = [u_{ik}]_{c \times N}$. The partition matrix satisfies the following conditions:

$$u_{ik} \in [0, 1], \quad 1 \leqslant i \leqslant c, \ \ 1 \leqslant k \leqslant N, \tag{10}$$

$$\sum_{i=1}^{c} u_{ik} = 1, \quad 1 \leqslant k \leqslant N, \tag{11}$$

$$0 < \sum_{k=1}^{N} u_{ik} < N, \quad 1 \leqslant i \leqslant c. \tag{12}$$

Equation (10) states the well-known fact that the membership degrees are real numbers distributed in the interval [0, 1]. Condition (11) constrains the sum of each column to 1, and thus the total membership of each $\boldsymbol{z}_k$ in all clusters equals one. Equation (12) means that none of fuzzy subsets is empty nor it contains all the data. The fuzzy partition matrix is obtained by applying the Gustafson-Kessel (GK) algorithm, based on the minimization of the well-known fuzzy c-means functional:

$$J(\boldsymbol{Z}; \boldsymbol{U}, \boldsymbol{G}, \{\boldsymbol{A}_i\}) = \sum_{i=1}^{c} \sum_{k=1}^{N} (u_{ik})^m D^2_{ik\boldsymbol{A}_i}, \tag{13}$$

where $\boldsymbol{G} = [\boldsymbol{g}_1, \boldsymbol{g}_2, \cdots, \boldsymbol{g}_c]$, $\boldsymbol{g}_i \in \mathbf{R^3}$ is a vector set of cluster prototypes (centers), which have to be determined, and

$$D^2_{ik\boldsymbol{A}_i} = (\boldsymbol{z}_k - \boldsymbol{g}_i)^T \boldsymbol{A}_i (\boldsymbol{z}_k - \boldsymbol{g}_i) \tag{14}$$

is the squared inner-product distance norm. Matrices $\boldsymbol{A}_i$ are computed in the optimization algorithm using the local covariance of the data around each cluster center. This allows each cluster to adapt the distance norm to the local distribution of the data. If the data samples are distributed along a nonlinear hyper surface, the GK algorithm will classify the data samples into some clusters[3]. The overlap of the clusters is controlled by the user-defined parameter $m \in [1, \infty)$.

Next, we present how to decide center and deviation of RBF. For deciding center of each RBF V_{0i} which is used in Eq. (4), we apply center of each cluster $\boldsymbol{G}$ obtained by GK algorithm:

$$V_{0i} = \boldsymbol{G}_{3i}, \quad (i = 1, 2, \cdots, c). \tag{15}$$

For deciding deviation of each RBF σ_i which is also used in Eq. (4), we apply means of Euclid distance of between center of each cluster and data set which is obtained by applying an α-cut.

$$\boldsymbol{Z}_i = \{\boldsymbol{z}_k | u_{ik} > \alpha, k = 1, 2, \cdots, N\}, \tag{16}$$

$$\sigma_i = \frac{1}{N_i} \sum_{\boldsymbol{z}_k \in \boldsymbol{Z}_i} (z_{3k} - V_{0i})^2, \quad k = 1, 2, \cdots, N, \tag{17}$$

where, $N_i (i = 1, ...c)$ are data samples which include in $\boldsymbol{Z}_i$. α-cut means that only the data samples that belong to the given cluster to a degree greater than a specified threshold $\alpha \in [0, 1]$ are used to decide deviation of each RBF.

4 Parameter Estimation

In the continuous-time model (6) (or (7)), the differential terms are contained. Generally speaking, it is not desirable to calculate the derivatives directly from the measurements, because it may make the noise effect worse. In this section, a discrete-time identification model for pulmonary respiration is derived based on Sagara's numerical integration technique[4].

Denote the sampling period of data collection as T. At time instant $t = kT$, integrate both sides of Eq. (7) over the interval $[(k - \ell)T, kT]$. Let $y(k)$ be the left hand side of the resultant equation. Then $y(k)$ can be calculated as

$$y(k) = \int_{(k-\ell)T}^{kT} P(\tau) d\tau \simeq \sum_{j=0}^{\ell} d_j P(k - j), \tag{18}$$

where, ℓ is a natural number that decides the window size of numerical integration. The coefficients $d_i (i = 0, 1, \cdots, \ell)$ are determined by formulae of numerical integration. For example, when the trapezoidal rule is taken, they are given as follows:

$$\begin{cases} d_0 = d_\ell = T/2, \\ d_j = T, \quad j = 1, 2, \cdots, \ell - 1. \end{cases} \tag{19}$$

The integral of $\boldsymbol{\varphi}(t)$ in the right hand side of Eq. (7) can be calculated by

$$\begin{aligned} \boldsymbol{\phi}(k) &= \sum_{j=0}^{\ell} d_j \boldsymbol{\varphi}(k - j) \\ &= \Big[P(k) - P(k - \ell) \;\Big|\; \sum_{j=0}^{\ell} d_j V(k - j) \psi_1(V(k - j)) \;\cdots \\ &\quad \sum_{j \overline{\ell} 0}^{\ell} d_j V(k - j) \psi_n(V(k - j)) \;\Big|\; \sum_{j=0}^{\ell} d_j Q(k - j) \psi_1(V(k - j)) \cdots \\ &\quad \sum_{j=0} d_j Q(k - j) \psi_n(V(k - j)) \;\Big|\; Q(k) - Q(k - \ell) \Big]. \end{aligned} \tag{20}$$

Here, analytical forms are taken for the terms where the integral can be calculated analytically. Get together the approximation error ∇_e caused by numerical integration and the integral of error term $\epsilon(t)$ of Eq. (7) in $e(t)$. Namely, $e(k)$ is

$$e(k) = \nabla_e + \int_{(k-\ell)T}^{kT} \varepsilon(\tau) d\tau. \tag{21}$$

Consequently, a discrete-time identification model of pulmonary respiration is derived as follows:

$$y(k) = \boldsymbol{\phi}^T(k)\boldsymbol{\theta} + e(k). \tag{22}$$

From the measurements of air pressure $P(k)$, flow $Q(k)$ and volume $V(k)$, it is easy to calculate $y(k)$ by Eq. (18) and $\boldsymbol{\phi}(k)$ by Eq. (20) at each time instant $k = \ell + 1, \cdots, N$, then $N - \ell$ regression equation can be derived as:

$$\boldsymbol{y} = \Phi\boldsymbol{\theta} + \boldsymbol{e}, \tag{23}$$

where, $\boldsymbol{y} = [y(\ell+1) \cdots y(N)]^T$, $\Phi = [\boldsymbol{\phi}(\ell+1) \cdots \boldsymbol{\phi}(N)]^T$, $\boldsymbol{e} = [e(\ell+1) \cdots e(N)]^T$, respectively.

The least squares estimate that minimizes the criterion function J defined as a sum of squared errors

$$J = ||\boldsymbol{y} - \Phi\boldsymbol{\theta}||^2 \tag{24}$$

is given by

$$\hat{\boldsymbol{\theta}} = (\Phi^{\mathbf{T}}\Phi)^{-1}\Phi^{\mathbf{T}}\boldsymbol{y}. \tag{25}$$

Then, the estimate of pulmonary elastance is obtained:

$$\hat{E}(V) = \sum_{i=1}^{n} \hat{b}_i \psi_i(V). \tag{26}$$

The above algorithm is an off-line algorithm in which the calculation is carried out after the data of length N are completely collected. But, in practical clinical cases, the data are recorded successively, and the state of lung may change, so on-line algorithm is desired. The on-line algorithm for calculating the above LS estimate is as the follows[5]:

$$\begin{cases} \hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k-1) + L(k)(y(k) - \boldsymbol{\phi}^T(k)\hat{\boldsymbol{\theta}}(k-1)), \\ L(k) = \dfrac{S(k-1)\boldsymbol{\phi}(k)}{\lambda + \boldsymbol{\phi}^T(k)S(k-1)\boldsymbol{\phi}(k)}, \\ S(k) = \dfrac{1}{\lambda}\Big[S(k-1) - \dfrac{S(k-1)\boldsymbol{\phi}(k)\boldsymbol{\phi}^T(k)S(k-1)}{\lambda + \boldsymbol{\phi}^T(k)S(k-1)\boldsymbol{\phi}(k)}\Big], \end{cases} \tag{27}$$

where, $\lambda(\lambda \leq 1)$ is forgetting factor, and the initial values of $\hat{\boldsymbol{\theta}}$ and S are taken as $\hat{\boldsymbol{\theta}}(0) = 0$, $S(0) = s^2 I$ (s is a sufficiently large real number).

5 Validation by Clinical Examples

In this section, the effectiveness of proposed method for deciding RBF parameter is validated. A case of using fuzzy clustering method and a case of mean location are compared with the results of pulmonary elastance estimation. The sampling period T is $T = 0.005$ second, and the data size N is $N = 600$. These data corresponds to one cycle of breath. The dynamical $P - V$ curve can be plotted

from the data of airway pressure P and volume V (Fig.1). For verification of estimated results of pulmonary elastance, experimental static recoil $P-V$ points during inspiration ($\circ$) and expiration ($+$) are also plotted in Fig.1. The results of estimation are evaluated by degrees how the calculated static elastic $P-V$ curve based on estimated parameter close to the static recoil $P-V$ points. RBF parameters which are decided by using Eq. (15), (16), (17) based on fuzzy clustering and by using mean location are shown in Table.1. In this example, cluster number c (RBF node number n) is $c = n = 6$.

The estimation results of static elastance are shown in Fig.1 and Table.2. In Table.2, J indicates the value of criterion function defined as Eq. (24). $E_{v\ell}$ is

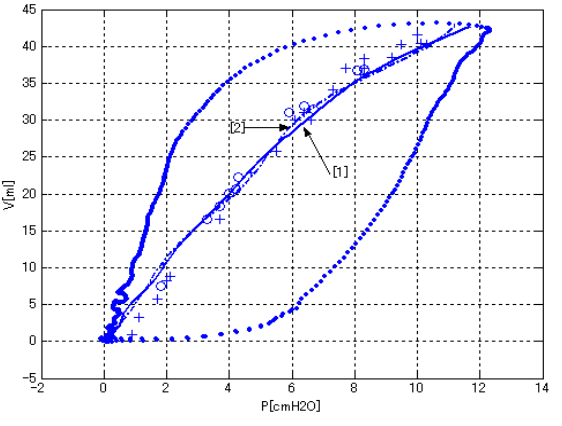


Fig. 1. Dynamical $P-V$ curve (dotted line), experimental static recoil pressure-volume points during inspiration($\circ$) and expiration ($+$), pulmonary elastance estimation results in the case using fuzzy clustering (1) and pulmonary elastance estimation results in the case using mean location (2).

Table 1. Design of RBF Network (node number $n = 6$)

center	$V_{01}[ml]$	$V_{02}[ml]$	$V_{03}[ml]$	$V_{04}[ml]$	$V_{05}[ml]$	$V_{06}[ml]$
clustering	31.8	40.5	22.2	13.4	6.07	1.27
mean location	33.0	44.0	25.5	17.0	8.5	0.0
deviation	σ_1	σ_2	σ_3	σ_4	σ_5	σ_6
clustering	4.63	3.65	4.64	3.58	2.08	0.81
mean location	5.0	5.0	5.0	5.0	5.0	5.0

Table 2. Evaluation of estimation results

	J	$E_{v\ell}$
clustering	3.8201×10^{-5}	0.2598
mean location	1.08×10^{-4}	0.2947

sum of squared errors between experimentally measured pressure value $P_\ell(i)$ at each static $P-V$ points and the estimated pressure value $\hat{P}_\ell(i)$ corresponding to the points: $E_{v\ell} = \frac{1}{N_v}\sum_{i=1}^{N_v}(P_\ell(i) - \hat{P}_\ell(i))^2$, where N_v is the number of static $P-V$ points used in validation. In Fig.1, we can see that both the estimated elastance curve by clustering and estimated elastance curve by mean location fit the practically measured static P/V curve quite well. However, the values of J and $E_{v\ell}$ by clustering are smaller than the values by mean location.

6 Conclusion

For setting appropriate respiratory conditions to fit each patient who is receiving artificial respiration treatment, pulmonary elastance and airway resistance provide decisive information. Therefore, it is very important to identify respiratory system model which include pulmonary elastance and airway resistance terms expressed by RBF network. In this paper, we applied fuzzy clustering method to optimize RBF network which is include pulmonary elastance and airway resistance term. Using fuzzy clustering GK algorithm, nonlinear data can be partitioned into some cluster, and center of each cluster and partition matrix which determines what degree of the individual data samples belong to the corresponding cluster are obtained.

The proposed method is that each RBF are located in the center of each cluster, and deviation of RBF apply means of Euclid distance of between center of each cluster and data set which is obtained by applying an α-cut. The proposed fuzzy clustering method is validated by clinical examples.

References

1. Maeda, K, Kanae, S., Yang, Z.J., Wada, K.: Estimation of Respiratory Parameters Based on RBF Network. The 23rd SICE Kyushu Branch Annual Conference. Kitakyushu Science and Research Park, December 4–5 (2004) 133–136
2. Gustafson, DE., Kessel WC.: Fuzzy Clustering with a Fuzzy Covariance Matrix. Proceedings of the IEEE CDC. San Diego, CA, USA (1979) 761-766
3. Babuska, R., Alic, L., Lourens, M.S.: Estimation of Respiratory Parameters via Fuzzy Clustering. Artificial Intelligence in Medicine, **21** (2001) 91-105
4. Sagara, S., Zhao, Z.: Numerical Integration Approach to On-Line Identification of Continuous-Time Systems. Automatica **26** (1990) 63–74
5. Ljung, L.: System Identification Theory for the User. Prentice-Hall (1987)

Recognition of Fatty Liver Using Hybrid Neural Network

Jiangli Lin[1], XianHua Shen[1], Tianfu Wang[1], Deyu Li[1], Yan Luo[2], and Ling Wang[1]

[1] Department of Biomedical Engineering, Sichuan UniversityChengdu, China
Linjiangli@sicnu.edu.cn, Tuohua@163.com,
{Tfwang, Lideyu}@scu.edu.cn, Wangling_2005@tom.com
[2] Ultrasound Departments, the First Huaxi Hospital,
Sichuan University, Chengdu, China
Luoyan77@vip.sina.com

Abstract. A hybrid neural network based on self-organizing map (SOM) and multilayer perception(MLP) artificial neural network(ANN) is proposed for recognition of fatty liver from B-scan ultrasonic images. Firstly, four texture features including angular second moment, contrast, entropy and inverse differential moment were extracted from gray-level co-occurrence matrices of B-scan ultrasound liver images. They were mapped by a SOM for feature reduction, and then combined with other two features, named approximate entropy and mean intensity ratio. All features were imposed to a MLP for recognition. In the experiment, 130 B-scan liver images were divided into two groups: 104 in training group and 26 in validation group. Both the normal and fatty livers were recognized correctly. This study showed that the hybrid neural network could be used for fatty liver recognition with good performances.

1 Introduction

Fatty liver is caused by accumulation of triglycerides and other fats inside liver cells. There are mainly two types of fatter livers, alcoholic liver disease (ALD) caused by abnormal consumption of alcohol and non-alcoholic fatty liver (the first stage of NAFLD) caused mainly by obesity. Actually, fatty liver is not harmful itself. But if left untreated, it will lead to inflammation of liver and hepatocellular necrosis, ultimately resulting in liver fibrosis and then irreversible cirrhosis. In clinic, the golden standard for diagnosis of fatty liver is biopsy. However, this procedure is invasive and makes patients suffer a lot. So doctors often make a presumptive diagnosis based on the following criteria first, and then biopsy is applied on suspicious individual for establishment:

- Clinical and/or biochemical signs of insulin resistance
- Chronically (long duration) elevation of SGOT and SGPT
- Signs of fatty liver on ultrasound

Among these criteria, ultrasound is the most convenient and non-invasive way, so it is widely used in the clinic diagnosis of fatty liver. However, for there is no

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 754 – 759, 2006.

objective and quantitative standards in ultrasonic diagnosis, this method is greatly depended on the experience of the diagnostician.

To solve this problem, many computer-aided diagnosis(CAD) methods have been proposed to increase the recognizing rate of fatty liver from ultrasonic images. For the normal histology of liver will be destroyed by diffuse and non-diffuse liver diseases, there will be some differences in the texture of ultrasonic images between the normal and fatty liver. Hence, most of the computer-aided diagnosis methods are based on the texture analysis of liver ultrasonic images, mainly including structural texture analysis, statistical texture analysis, model-based texture analysis and transform-based texture analysis [1]. Besides, there are still some methods based on gray level feature analysis, such as liver to kidney gray level ratio, adjoining point kernel and histogram analysis, etc [2]. Based on our former work, in this paper, the statistical texture, gray level and complexity features are analyzed.

Though many studies have been conducted, the input parameters and topological structure of the neural networks need to be studied further for a useful computer-aided diagnosis means with acceptable recognition performance. Kadah et al used neural networks to recognize diffusive liver disease through B-scanned ultrasonic image and achieved high discriminating rate. However, this algorithm is based on the extraction of a number of features and a large amount of multiplications in spatial domain among the features [3], so the complexity of the algorithm is high. Gletsos et al proposed a hybrid neural network composed of multiple BP neural networks to classify liver diseases step by step [4]. This algorithm achieved rather high performance, but it is also accompanied with calculation of a lot of input features and large-scale network. Furthermore, the convergence is still significant problem, which will probably lead to a local optimal result.

In this paper, a hybrid neural network is proposed for B-scanned ultrasonic image recognition of fatty liver. This hybrid neural network consists of two consecutive neural networks, a self-organizing map (SOM) neural network and a multilayer perception(MLP) with back propagation (BP) technique(called BP network in the following). SOM is used for feature mapping, and BP for recognition.

2 Method

The purpose of the research on the topological structure of the neural network is to improve the discriminating rate of fatty liver and reduce the computational cost. For a CAD system in clinic, false negative(FN) must be zero (none of the fatter liver is recognized as normal liver). The hybrid network is built according to this principle.

2.1 Partial Feature Map Based on SOM

SOM is proposed by T. Kohonen. By learning the input pattern iteratively, SOM network may make the spatial density of connective weight vectors, which represent the spatial distribution of connective weight value, trend towards the probability of the input pattern. Traditionally, dimensional reduction process means reducing the dimensions of the whole input features, and then the output is used as the input of the

next network [5,6]. As three kind of features based on gray level, complexity and texture were used in this study, if all features with different physical meanings are equally mapped, the resultant features will be confused with no corresponding explanations. So a new partial feature mapping is proposed. This new feature mapping will not process all input features but only those features with same properties. Through mapping, fewer features, even single feature will obtained. This procedure will reduce the computational cost and weaken the feature interference in a great deal.

In our experiments, only four second-order texture features named Angular second moment(ASM), contrast(CON), entropy(ENT) and inverse differential moment (IDM)[7]) based on gray-level co-occurrence matrices are used for mapping. Though SOM network mapping, the four features will be classified into two categories, that is, the two feature vectors: map1 and map2. The weight value of these two feature vectors, w_1 and w_2, are corresponding to two clustering centers of input feature vector, and map1 and map2 are corresponding to the distance between map1 and w_1, map2 and w_2 respectively.

2.2 BP Network

BP network is a multilayer feed forward neural network, which makes the mapping from input to output by the minimizing the cost function. This minimizing procedure is composed of two stages. In the first stage, the output of each unit is obtained by using existing connective weights for forward propagation of the inputs. In the second stage, the generalized errors of the output layer will back propagate layer by layer towards the input layer to obtain the reference error of each node used for the adjustment of the connective weights. The transfer function for the hidden layer and output layer is generally sigmoid function:

$$f(x) = \frac{1}{1+e^{-x}} \tag{1}$$

BP network is very sensitive to the input sequence of training features. The explanation may be that the interference of some redundant feature will affect the whole network through propagation. Therefore, the input feature of the BP network should be strictly selected to insure that every input feature can correctly reflect the classified characteristics of the training patterns.

2.3 Hybrid Neural Network

It is proved that single neural network always has its own limitations. For instance, the SOM network is an unsupervised system, learning ability can't be controlled in training. On the other hand, BP network has the ability of supervised leaning, but it correspondingly requires complex training with multiple feature inputs, which will enlarge the net scale and increase the computational cost. It also suffers from convergent rate and local optimum.

A hybrid neural network is traditionally a combination of two or more different neural networks [6]. In this paper, an unsupervised SOM is combined with the supervised BP network, which emphasizes both the self-clustering property and the learning ability. The architecture of hybrid neural network is shown in Fig.1. The SOM network has four nodes in the input layer and will produce two clustering distances map1, map2 as outputs. The central position of the whole patterns used as the initial value of the weight parameter in SOM network. The Euclidean distance is used in competition, and the distances between input vector and nodal weights (the two clustering centers) d_1, d_2 are calculated. The corresponding output node of the minimum of d_1, d_2 will be regarded as enhanced node. In experiment, training periods *epochs*=1000; the ordering phase learning rate *OPL*=0.9; the tuning phase learning rate *TPL*=0.02. From experimental results, map_2 is chosen as an input node for the latter BP network.

A 3×7×1 topological structure of BP network is developed, three nodes (APE,MIR,map2) in the first layer, seven nodes in the hidden layer and one node in the output layer. The maximum iterative times is 10000, the objective error is 10^{-5} and the minimum value for gradient is $10^{-10.}$

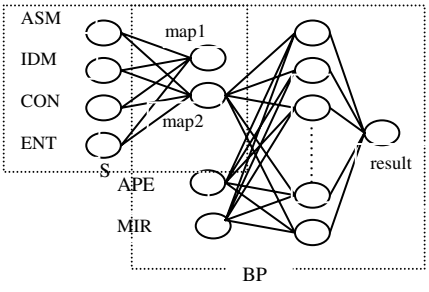
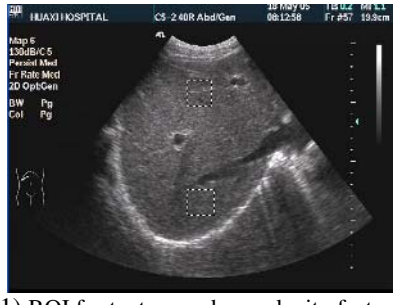


Fig. 1. The architecture of hybrid network

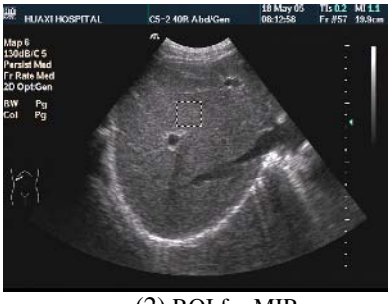
3 Experimental Result

All the ultrasonic images of liver used in the experiments have confirmed diagnosis results by expert team in Huaxi Hospital of Sichuan University. The total number of experimental images is 130, including 55 B-scanned ultrasonic images of normal liver and 75 images of homogeneous fatty liver. In the experiments, the images are separated into training group and validation group. There are 104 images (60 fatty and 44 normal livers) in the training group and 26 images (15 fatty and 11 normal livers) in the validation group.

Six features are extracted, including ASM, Con, Ent and IDM based on texture analysis, the APE based on complexity[8] and the MIR. The size of all the region of interest(ROI) used for feature extraction are 50×50 pixels. The position of the ROI for texture features and Ape is show in Fig.2 (1), while the position of ROI for MIR is shown in Fig.2 (2). The vasa and ribs have been avoided in all collected images.



(1) ROI for texture and complexity feature



(2) ROI for MIR

Fig. 2. ROI

For both training and validation group, 75 normal livers and 55 fatty livers are recognized correctly by the proposed hybrid neural network. Furthermore, the CAD system is robust. However, if all the six features were used for recognition by BP network, the recognizing rates of fatty liver and normal liver are 100% and 90.9% respectively, and the network is extremely unstable.

4 Conclusion

In this paper, a hybrid neural network composed of SOM and BP neural network is proposed in the application of B-scan ultrasonic image recognition of fatty liver. Experimental results implied that this model can be used in CAD system with high performance. This hybrid neural network has the property of both unsupervised and supervised learning. This kind of hybrid neural network is quite suitable for the reduction of feature come from quite different physical meanings.

Acknowledgements

This study was supported by the National Science Foundation of China (No 60272060) and the Sichuan Provincial Young Science Foundation of China (No 05ZQ026-019).

References

1. Huang, Y.L., Li, F.H., Zhao, Z.: Experimental Study of Texture Feature Extraction Methods of Liver Ultrasonic Imaging. Chin. J. Med. Imaging Technol. 20 (2004) 1937-1939
2. Wang, L.J., Wang, S.L.: The Quantitative Assessment at Fatty Infiltration of The Liver by The Square Chart or B-mode Ultrasonography. Chin. J. Med. Imaging Techn. 13.(1997) 388-389
3. Kadah, Y.M., Farag, A.A., Zurada, J.M., Badawi, A.M., Youssef, A.B.M.: Classification Algorithms for Quantitative Tissue Characterization of Diffuse Liver Disease from Ultrasound Images. IEEE Transactions on Medical Imaging 15. (1996) 466-478

4. Gletsos, M., Mougiakakou, S.G., Matsopoulos, G.K., Nikita, K.S., Nikita, A.S., Kelekis, D.: A Computer-aided Diagnostic System to Characterize CT Focal Liver Lesions: Design and Optimization of A Neural Network Classifier. IEEE Transactions on Information Technology in Biomedicine 7 (2003) 153-62
5. Zheng, Y., Greenleaf, J.F., Gisvold, J.J.: Reduction of Breast Biopsies With a Modified Self-organizing Map. IEEE Transactions on Neural Networks 8 (1997) 1386-1396
6. Miyanaga, Y., Jin, H.L., Islam, R., Tochinai, K.: A Self-organized Network with A Supervised Training. IEEE Proceedings of International Symposium on Circuits and Systems 1 (1995) 482-485
7. Haralick, R.: Statistical and structural approaches to Texture. Proceedings of IEEE 67 (1979) 786-804
8. Pincus, S.M.: Approximate Entropy as A Measure of System Complexity. Proceedings of the National Academy of Sciences 88 (1991) 2297-2299

A Novel Fast Fuzzy Neural Network Backpropagation Algorithm for Colon Cancer Cell Image Discrimination

Ephram Nwoye, Li C. Khor, Satnam S. Dlay, and Wai L. Woo

School of Electrical, Electronic and Computer Engineering,
University of Newcastle upon Tyne, United Kingdom
{ephraim.nwoye, l.c.khor, s.s.dlay, w.l.woo}@ncl.ac.uk

Abstract. In this paper a novel fast fuzzy backpropagation algorithm for classification of colon cell images is proposed. The experimental results show that the accuracy of the method is very high. The algorithm is evaluated using 116 cancer suspects and 88 normal colon cells images and results in a classification rate of 96.4%. The method automatically detects differences in biopsy images of the colorectal polyps, extracts the required image texture features and then classifies the cells into normal and cancer respectively. The net function computation is significantly faster. Convergence is quicker. It has an added advantage of being independent of the feature extraction procedure adopted, with knowledge and learning to overcome the sharpness of class characteristics associated with other classifiers algorithms. It can also be used to resolve a situation of in-between classes.

1 Introduction

Colon Cancer is the third most common cancer in the UK after lung and breast cancer and also third leading cause of cancer death in the US after the lung/bronchial cancer and breast/prostate cancer. It also represents the second largest cause of death from cancer in the western world. Colon cancer is the second killer disease next to heart attack. Over 1000 people die every week in the USA alone according to the latest information from the American medical Journal. In UK over 30,000 people develop colorectal cancer each year. The American Cancer society reported almost 106,000 people will have colorectal cancer in 2003 and 2004 with over 57000 leading to death in America [1 & 2]. This disease is uncommon in Asia, Africa and South America. The pathologists, daily, screen large numbers of slides containing These cancerous cells manually. These cells are similar in shape, size and cells structure. This procedure becomes arduous, difficult and can effect the Pathologist's decision.

The aim of this research therefore is to use fast fuzzy neural backpropagation network approach together with statistical feature extraction method to classify colon cell images into normal or cancer. This will result in an automated tool for accurate diagnosis.

2 Formulation of Fast Fuzzy Backpropagation Algorithm [3-10]

The Fast Fuzzy Neural Backpropagation network is a three layered feed forward network with back propagation learning algorithm. The network has two phases, the

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 760 – 769, 2006.

learning and decision making phases. The network is made up of fuzzy neurons that are characterized by fuzzy input signals and fuzzy weights which perform the mapping function.

Let $\rho(x)$ be a sigmoid function and Γ be a compact set in $\Re^n$. Given any image function $f:\Re^n \to \Re$ on Γ, and for an arbitrary $\varepsilon > 0$, there exists an integer N and real constants $c_i, \theta_i, w_{ij}, i = 1,2,...,N; j = 1,2,...,M$ such that equation (1) satisfies the condition $Max_{x\in\Gamma} \parallel f(x_1, x_2, ..., x_n) - \bar{f}(x_1, x_2, ..., x_n) \parallel^2 < \varepsilon$

$$\bar{f}(x_1, x_2, ..., x_n) = \sum_{i=1}^{N} c_i \rho(\sum_{j=1}^{M} (w_{ij} x_j - \theta_i) \tag{1}$$

Based on the above lemma we formulate a three layer Fuzzy Neural Backpropagation Network where the hidden layer transfer function employs the sigmoid function $\rho(x)$. All vectors in this paper are column vectors shown by equations (2). The network input, hidden layer output, output layer output and the desired or target output are represented respectively by the following matrix notations:

$$\begin{aligned} X &= [x_1, x_2, ..., x_p]^T \in \Re^{L.P}; Y = [y_1, y_2, ..., y_p]^T \in \Re^{H.P} \\ O &= [o_1, o_2, ..., o_p]^T \in \Re^{K.P}; D = [d_1, d_2, ..., d_p] \in \Re^{K.P} \end{aligned} \tag{2}$$

L, K, P denote the number of input layer neurons, output layer neurons and the number of image features while W and V represent the hidden layer weight matrix. The subscript t denotes the index of iterative learning epochs while subscripts such as d_p^k refers to a specific output vector component and $X = [x_1, x_2, ..., x_p]^T \in \Re^{L.P}$ is texture features extracted from colon cell images. From equation 11 the error function is derived and the objective of network training is to minimize an error function J given by:

$$J = \frac{1}{2PK} \sum_{p=1}^{P} \sum_{k=1}^{K} (o_p^k - d_p^k)^2 \tag{3}$$

In the gradient descent steepest algorithm [9], the updates of weights are computed by:

$$W_{t+1} = W_t - \beta(\frac{\partial J}{\partial W_t}); V_{t+1} = V_t - \beta(\frac{\partial J}{\partial V_t}) \tag{4}$$

Consider a three layer feed forward neural net network shown in the fig.1 below, the error matrix is defined as the error difference between the actual outputs and the desire(target) outputs; That is

$$E_t = O_t - D = W_t Y_t - D = W_t V_t X - D \tag{5}$$

Substituting equation (4) into (3) we have

$$J_t = \frac{1}{2PK} T_r(E_t E_t^T) \tag{6}$$

where T_r is the matrix trace operator. So W_{t+1} and V_{t+1} are updated by the following equations:

$$W_{t+1} = W_t - \beta_t (\frac{\partial J}{\partial W_t}) = W_t - \beta_t \frac{1}{PK} E_t Y_t^T \tag{7}$$

$$V_{t+1} = V_t - \beta_t (\frac{\partial J}{\partial V_t}) = V_t - \beta_t \frac{1}{PK} W_t^T E_t X^T \tag{8}$$

From equation (3) to equation (8) we derived the computation of network training error as follows:

$$E_{t+1} E_{t+1}^T = (W_{t+1} Y_{t+1} - D)(W_{t+1} Y_{t+1} - D) \tag{9}$$

Substituting equation (5) and equation (6) into equation (7) will yields

$$\begin{aligned} E_{t+1} E_{t+1}^T =& [(W_t - \frac{\beta_t}{PK} E_t Y_t^T)(V_t - \frac{\beta_t}{PK} W_t^T E_t X^T) X - D)] \\ &*[(W_t - \frac{\beta_t}{PK} E_t Y_t^T)(V_t - \frac{\beta_t}{PK} W_t^T E_t X^T) X - D)]^T \end{aligned} \tag{10}$$

Expanding equation (10) and collecting like terms we get:

$$\begin{aligned} E_{t+1} E_{t+1}^T =& E_t E_t^T - \frac{\beta_t}{PK} [E_t (W_t W_t^T E_t X^T X)^T + E_t (E_t Y_t^T V_t X)^T + W_t W_t^T E_t X^T X E_t^T + E_t Y_t^T V_t X E_t^T] \\ &+ \frac{\beta_t^2}{(PK)^2} [E_t (E_t Y_t^T W_t^T E_t X^T X)^T + E_t Y_t^T W_t^T E_t X^T X E_t^T + E_t Y_t^T V_t X (W_t W_t^T E_t X^T X)^T \\ &+ W_t W_t^T E_t X^T X * (E_t Y_t^T V_t X)^T + E_t Y_t^T V_t X (E_t Y_t^T V_t X)^T + W_t W_t^T E_t X^T X (W_t W_t^T E_t X^T X)^T] \\ &- \frac{\beta_t^3}{(PK)^3} [W_t W_t^T E_t X^T X (E_t Y_t^T W_t^T E_t X^T X)^T + E_t Y_t^T V_t X (E_t Y_t^T W_t^T E_t X^T X)^T \\ &+ E_t Y_t^T W_T^T E_t X^T X (W_t W_t^T E_t X^T X)^T + E_t Y_t^T W_t^T E_t X^T X (E_t Y_t^T V_t X)^T] \\ &+ \frac{\beta^4}{(PK)^4} E_t Y_t^T W_t^T E_t X^T X (E_t Y_t^T W_t^T E_t X^T X)^T \end{aligned} \tag{11}$$

If we set;

$$g=\frac{1}{PK}Tr[E_t(W_tW_t^TE_tX^TX)^T+E_t(E_tY_t^TV_tX)^T+W_tW_t^TE_tX^TXE_t^T+E_tY_t^TV_tXE_t^T]$$

$$c=\frac{1}{(PK)^2}Tr[E_t(E_tY_t^TW_t^TE_tX^TX)^T+E_tY_t^TW_t^TE_tX^TXE_t^T+E_tY_t^TV_tX(W_tW_t^TE_tX^TX)^T$$

$$+W_tW_t^TE_tX^TX*(E_tY_t^TV_tX)^T+E_tY_t^TV_tX(E_tY_t^TV_tX)^T+W_tW_t^TE_tX^TX(W_tW_t^TE_tX^TX)^T]$$

$$b=\frac{1}{(PK)^3}Tr[W_tW_t^TE_tX^TX(E_tY_t^TW_t^TE_tX^TX)^T+E_tY_t^TV_tX(E_tY_t^TW_t^TE_tX^TX)^T$$

$$+E_tY_t^TW_T^TE_tX^TX(W_TW_t^TE_tX^TX)^T+E_tY_t^TW_t^TE_tX^TX(E_tY_t^TV_tX)^T]$$

$$a=\frac{1}{(PK)^4}Tr[E_tY_t^TW_t^TE_tX^TX(E_tY_t^TW_t^TE_tX^TX)^T].$$

We derived that the error function is updated by

$$J_{t+1}-J_t=\frac{1}{2PK}[a\beta_t^4+b\beta_t^3+c\beta_t^2+g\beta_t]\cong\frac{1}{2PK}f(\beta_t) \tag{12}$$

$A>0$ is always certain. According to algebraic theorem, $f(\beta_t)$ has two or four real roots including one constant zero root, there exist one or two constant intervals in β_t coordinates satisfying $f(\beta_t)<0$. We need to find the minimum value of $f(\beta_t)$ that result in the largest reduction of J at each epoch by solving the derivative of $f(\beta_t)$. So we differentiate $f(\beta_t)$ with respect to β_t get that

$$\frac{\partial f(\beta_t)}{\partial(\beta_t)}=4a\beta_t^3+3b\beta_t^2+2c\beta_t+g \tag{13}$$

Dividing through by $4a$, we have

$$\frac{\partial f(\beta_t)}{\partial(\beta_t)}=\beta_t^3+\frac{3b}{4a}\beta_t^2+\frac{2c}{4a}\beta_t+\frac{1}{4a}g \tag{14}$$

The above equation is a standard cubic equation, the solution [10] of which will produce the optimal learning rate. Computing the three root of the above polynomial and selecting the minimum value will yield

$$\beta_{t(optimal)}=\{\beta_i \mid f(\beta_i)=\min(f(\beta_1),f(\beta_2),f(\beta_3));\ \forall i\in\{1,2,3\}\} \tag{15}$$

where β_i are the three real roots of $\frac{\partial f(\beta_t)}{\partial(\beta_t)}$ and $\beta_{t(optimal)}$ is then the optimal learning rate.

The fuzzy backpropagation algorithm is described thus:

1) Set the input values of the training features. The input values of the training data are determined arbitrarily. They transformed in the unit interval of [0, 1].
2) Set target values of the input data. The target training data are determined in the unit interval [0, 1] according to our convergence conditions or semi-automatically by experts that define the direction and degree of target value changes.
3) Determine the neural network initial weights. The neural network initial weights are set also arbitrarily without requiring being different from each other within the unit interval [0, 1].
4) Compute the error function E_t. If $E_t > \varepsilon$ (desired accuracy), then the training stops, else go to the next step.
5) Compute $a, b, c, g, and \beta_{1,2,3}$; and also the DISCRIMINANT of the cubic equation. If the discriminant is less than zero, then $\beta_{t(optimal)}$ is the real root, but if it is less than zero we compute the $\beta_{t(optimal)}$ by equation (15).
6) Update the weights by equation (6 & 7). The net is therefore computed and described layer by layer as in equations 16, 17, and 18 below.

$$\text{Input layer: } \mathrm{X}_i : X_i^{/} = x_i = \left(x_1, x_2, ...x_n\right)^{\mathrm{T}} \in [0,1] \tag{16}$$

$$\text{Hidden layer: } net_i^{(1)} = \max[\sum(\min X_i^{/}).w_{ij}]; a_h^{(1)} = f(net_i^{(1)}) \tag{17}$$

$$\text{Output layer: } net^{(2)} = \max[\sum(\min a_h^{(1)}).\min w_{ij}] \tag{18}$$

The convergence condition for the fast fuzzy backpropagation is formulated and proved for a multiple training features with the equal values target vectors as;

Given $(X_p, T_p), p = 1,..,P; T_p = (t_{p1}, t_{p2}, ..., t_{pn_2})$, the fast fuzzy backpropagation will converge to a target values $t_{pk}, p = 1,...,P; k = 1,...,n_2$ if there exist a number $\delta \in Z$ such that the following equality holds:

$$t_{pk} = a_k^{(2)}(\delta + p - 1) = a_k^{(2)}(\delta + P + p - 1) = a_k^{(2)}(\delta + 2P + p - 1) = ...$$
$$w_{ij}^{(l)}(\delta) = w_{ij}^{(l)}(\delta + 1) = w_{ij}^{(l)}(\delta + 2) = ...; \forall l = 1.2$$

If $x_{i^{''}}; x_{i^{'}}$ are max and min values of the input element respectively such that $x_{i^{''}} \le t_{pk}; x_{i^{'}} \ge t_{pk}$; Then fast fuzzy backpropagation is convergent to the target t_{pk} $\forall p, k(t_{pk} = x_{i^{''}})$;

2.1 Colon Image Analysis and Feature Extraction

The concepts of image analysis and feature extraction are worldwide research topics and have applications which are wide ranging [11-13]. It includes methods that vary from wavelet analysis to matrices showing variations of grey scale levels between pixels of images. The common purpose for all of them is to structure these variations in the examined images and derive some measurements that can be used to classify a particular image [11]. In our classification, the individual elements of the gray level cooccurrence matrix were not used rather features were derived from the matrices. A large number of textural features have been proposed starting with the original fourteen features described by Haralick et al [14], however only some are in wide use. In this research we only presented five which showed good discrimination among the cell images. The following features were calculated using the cooccurrence matrices of the colon images: Contrast (CON.), Angular Second Moment (ASM), Entropy (ENT.), Inverse Difference Moment (IDM) and Correlation (COR). The co-occurrence method of texture description we implemented was rotation invariant and calculated with symmetry needing only four angles (θ=0, 45, 90, 135 degrees) as effective choice at a given distances (d=1,). Tables1 gives the summary of the formulae used to compute the above features.

3 Implementation

After tissue samples were collected through biopsy and prepared in the pathology and histology laboratories, slides were digitally captured and assessed using image analysis system consisting of a photomicroscope (Nikon) equipped with high resolution video camera (Robert Bosch JCCD). All images were then divided into small images of 256x256 formats. The images were then normalized. Fig 5 and Fig 6 show the samples histogram of colon cancer cell images before and after normalization respectively. Then texture features were extracted as described above and fed into the fast fuzzy neural backpropagation classifier.

3.1 Results and Discussion

The images were divided into three sets one for training, another for testing and the last set for validation. The first 44 normal images and 58 cancer images were used for training. The second 44 normal images and 58 cancer images were used for testing. For validation a third set, 44 normal images and 58 cancers were selected from the other two previous sets. Fig.3 show target met during training after 16 epochs while Fig.9 shows the error performance for training, testing and validation phases. Table2 give the best results returned from training testing and validation after performing many experiments consisting of 116 cancer cell images and 88 normal cell images. The overall classification rate achieved by combining all the five feature elements is 96.4%. There was significant increase in the classification rate as the statistical co-occurrence feature values were combined in a forward manner as seen in Table 3. To find how the errors are distributed across the classes the confusion matrix of Table 4

Table 1. Formulae for calculating image features

Features	Calculation
Entropy (ENT)	$\sum P(i,j)\log P(i,j)$
Contrast (CON)	$\sum (i,j)^2 P(i,j)$
Angular Second Moment (ASM)	$\sum P(i,j)^2$
Inverse Difference Moment (IDM)	$\sum P(i,j)/(i-j)^2$
Correlation(COR)	$\sum \left((i,j)P(i,j) - (\mu_x * \mu_y)\right)/(\sigma_x * \sigma_y)$ where $\mu_x = \sum_i i \sum_j P(i,j)$; $\mu_y = \sum_j j \sum_i P(i,j)$; $\sigma_x = \sum_i (i-\mu_x)^2 \sum_j P(i,j)$; $\sigma_y = \sum_j (j-\mu_y)^2 \sum_i P(i,j)$

Table 2. Result Returned from Testing and Validation

Selected features	Normal	Cancer
ENT	0.0121±0.2522	0.2632±0.0086
CON	0.3245±0.1993	0.6543±0.0096
ASM	0.0429±0.0073	0.0518±0.0102
COR	0.192±0.0532	0.3682±0.0421
IDM	0.0083±0.0035	0.1306±0.0034

Table 3. Forward combination of texture feature elements

Feature combinations in the calculations	Classification Rate
CON	0.8625
CON & ASM	0.8797
CON, ASM & ENT	0.9276
CON,COR, ASM ENT & IDM	0.9644

Table 4. Confusion matrix

Classified as / Correct classes	Cancer	Normal	Classification Rate (%)	Error Rate
Cancer	114	2	98.27	0.017
Normal	2	82	97.72	0.027

was constructed where some misclassifications are shown resulting in an error rate of 3.6% which could have resulted from the deficiency of statistical co-occurrence as feature extraction technique, incomplete elimination of noise in the images, medical specimen preparation and other pre-processing anomalies. This means that 3 out of 116 cancer images were recognized as normal while 4 out of 88 normal images were recognized as cancer images.

2.3 Figures and Photographs

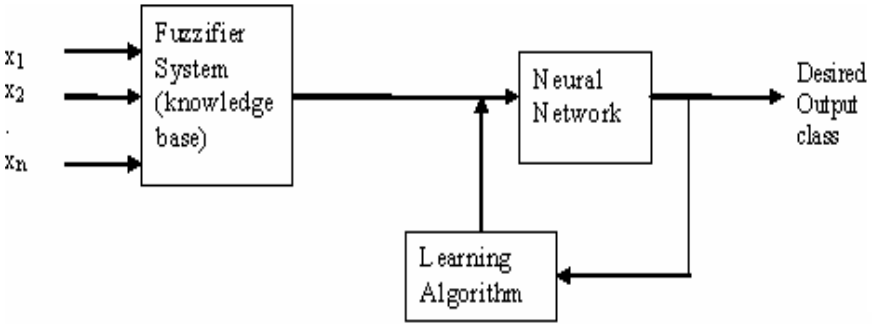


Fig. 1. Block diagram of fast fuzzy backpropagation system

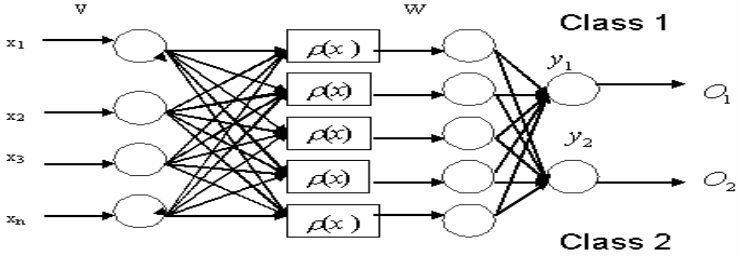


Fig. 2. Network diagram of fast fuzzy backpropagation system

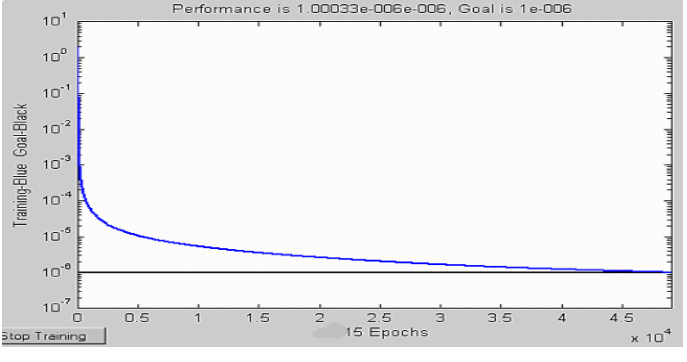


Fig. 3. Fast Fuzzy Training with set Target

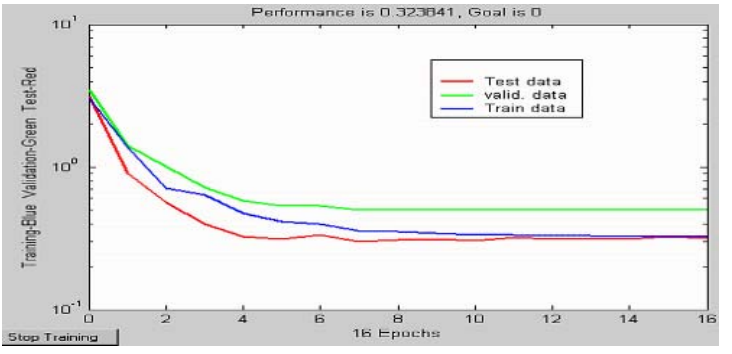


Fig. 4. The fast Fuzzy BP Error graphs

4 Conclusion

In this work, the superiority, high performance and quick convergence of fast fuzzy backpropagation algorithm in classification of colon cancer cells have been demonstrated. The work presents the classification of colon cells using fast fuzzy backpropagation neural network. It is shown that the error was minimal, net function computation significantly faster and convergence quicker only after 16 epochs in a two class application. It demonstrates that textures features extracted from colon cell images using statistical tools combined in a fuzzy neural classifier can result in a highly significant discrimination and classification of malignant and normal cells. The accuracy of our method on training set, testing set and validation with quick convergence is significantly higher, even though that the number of adjustable parameters remains the same. The experiment shows that the method is accurate, fast, and exhibits very useful feature such that it can be used to resolve a situation where a cell is neither cancer nor normal. However, we caution that this high accuracy reported does not represent a complete solution to the problem of classifying colon cancer cells. A complete solution may need a combination of methods.

References

1. Souhami, R., Tobias, J.: Cancer And Its Management. Blackwell Pub. Co. Inc., USA (2003)
2. Souhami, R., Tobias, J.: www.Cancer.Org.: Cancer Statistics 2004. Downloaded on 4th April (2004)
3. Donna, L.H., Cohen, M.E.: Neural Network And Artificial Intelligent For Biomedical Engineering. IEEE Press Series in Biomedical Eng., IEEE Press Inc. New York (2000)
4. Mitra, S., Hayashi, Y.: Neuro-Fuzzy Rule Generation: Survey in Soft Computing. IEEE Trans. Neural Networks 11 (2000), 748–768
5. Zadeh, L.A.: Fuzzy Sets, Information And Control 8 (1978),338–353
6. Wang, J., Lee, C.S.G.: Self Adaptive Neuro-Fuzzy Inference Systems for Classification Applications. IEEE Trans. Fuzzy Systems 10 (6)(2002) 790–802

7. Nwoye, E., Dlay, S.S., Woo, W.L.: Fuzzy Neural Machine with Image Feature Extraction For Colorectal Cancer Diagnosis. IASTED Proceedings on Visualization, Imaging & Image Processing, 2004, 304–308
8. Nwoye, E., Dlay, S.S., Woo, W.L.: Texture Features for Cancerous Cell Image Classification Using Fuzzy Neural Technique. CSNDSP Proceedings (2004) 364–370
9. Scherk, J.: Algebra: A Computational Introduction. Boca Raton, Fl Chapman, & Hall, (2000)
10. Tan, K. C., Tang, H. J.: New Dynamic Optimal Learning for Linear Multilayer FNN. IEEE Trans. on Neural Network, 15 (6), (2004), 1562–1568
11. Nixon, M.S., Aguado, A. S.: Feature Extraction & Image Processing. Newnes, Pub. Co. Oxford, UK (2002) 247–308
12. Tobias, O.J., Seara, R.: Image Segmentation by Histogram Thresholding Using Fuzzy Sets. IEEE Trans. on Image Processing 12 (1) (2003) 1450–1456
13. Gonzalez, R.C., Woods, R.E.: Digital Image Processing.Upper Saddle River, Prentice Hall (2002)
14. Haralick, R.M., Sternberg, S.R., Zhuang, X.: Image Analysis Using Mathematical Morphology. IEEE Transactions on Pattern Anal. & Machine Intellig. 9(4) (1987) 532–550

Poultry Skin Tumor Detection in Hyperspectral Images Using Radial Basis Probabilistic Neural Network

Intaek Kim[1], Chengzhe Xu[1], and Moon S. Kim[2]

[1] Department of Communication Engineering, Myongji University,
38-2 Namdong, Yongin, Kyonggido, South Korea
kit@mju.ac.kr, xcz75@sina.com
[2] USDA ARS, BA, ANRI, ISL, Bldg. 303, BARC-East,
10300 Baltimore Avenue, Beltsville, MD 20705-2350, U.S.A.
kimm@ba.ars.usda.gov

Abstract. This paper presents a method for detecting poultry skin tumors using hyperspectral fluorescence image. New feature space is generated by the ratio of intensities of two bands, the combination of images such that their intensity ratios yield the least false detection rate is selected by minimizing overlap area of normal and tumor's PDFs. Four feature images are chosen and presented as an input to a classifier based on the radial basis probability neural network. The classifier categorizes the input with three classes, where one is for tumor and two for normal skin pixels. The classification result based on this method shows the improved performance in that the number of false classification is reduced.

1 Introduction

A hyperspectral image is a three-dimensional (3D) volume of data containing two-dimensional (2D) spatial information measured at a sequence of individual wavelength across sufficiently broad spectral bands (spectrum). The spectral information provided by a certain pixel is valuable in the discrimination, detection, and classification of elements and structures within the image. It shows a great potential for detection of abnormality caused by the change of its composition. A tumor found in poultry, especially chicken, is a round, ulcerous lesion surrounded by a rim of thickened skin and dermis [1]. Tumors appear as small, scattered, localized shape deformation, with only slight discolorization. It is therefore that conventional vision systems operating in the visible region are often too limited in sensitivity for adequate detection of skin tumors.

Studies have shown that the presence of defects is more easily detected by using two or more spectral band images which is usually called multispectral image [2]. Detection of chicken skin tumor using multispectral reflectance imaging has also been reported to have the ability to differentiae wholesome and unwholesome agricultural products [3]. More recently, different two approaches have been proposed to address the tumor detection problem with the same set of data acquired in the USDA Instrumentation and Sensing Laboratory (ISL). Fuzzy inference scheme using a small number of fuzzy rules and Gaussian membership functions was applied to detect skin

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 770 – 776, 2006.

tumors on poultry carcasses [4]. The other proposal on the data employing a two-step procedure used both spectral and spatial information to differentiate the tumor and normal regions in the image [5]. In this study, 65 spectral bands are obtained from wavelength λ_1 = 425.4 nm to λ_{65} = 710.7 nm with the same interval. Fig. 1 shows spectral band images at the wavelength of λ_5, λ_{25}, λ_{45}, and λ_{65}.

This paper presents a method for poultry skin tumor detection in hyperspectral fluorescence images. It is the result of successive work on the data provided by USDA ISL. To detect the tumor, it is necessary not only to extract features for tumor but also to reduce the number of features. Features are selected so as to minimize the error probability. Radial basis probabilistic neural network (RBPNN) is used as a classifier. Experimental result shows there is an improvement in detecting tumor in that it lessens the false alarm rate.

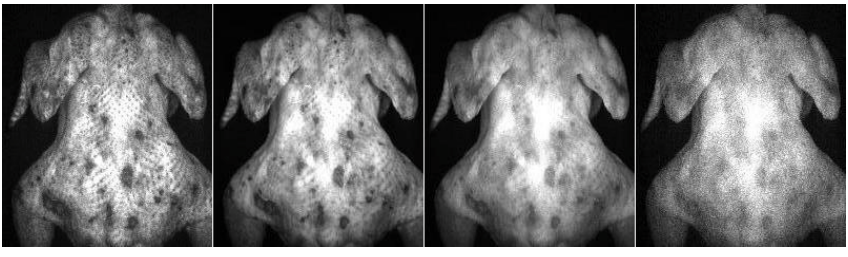


Fig. 1. Hyperspectral fluorescence images at the wavelength of λ_5, λ_{25}, λ_{45}, and λ_{65} (from left to right)

2 Tumor Detection Algorithm for Hyperspectral Images

A classifier for detecting tumors in a hyperspectral image requires reduction of input data because the entire hyperspectral data is so huge that it cannot be used as an input. The input to a classifier should be designed such that it represents a set of excellent clues for finding decision boundaries. From that viewpoint, feature extraction is very important issue in analyzing the hyperspectral image because the number of features could be very small compared with the original image data.

2.1 Method for Feature Extraction

Fig. 1 shows that the normal skin and tumor can be considerably differentiated by the intensity of pixels. It is however observed that there is an ambiguity in the intensity to distinguish between them. It indicates the intensity itself is an insufficient feature for the discrimination. On the other hand, using all of hyperspectral image data is also impractical in terms of computational load and convergence of decision boundary. Therefore, a method for obtaining efficient feature extraction should be addressed.

To this end, transforming feature space should be done to make the within-class distance smaller and the between-class distance larger. At the same time, it is also to achieve the reduction of dimension to avoid the curse of dimensionality and further

efficiency could be gain. Transforming feature space is a process to represent the features so that they can be well distinguished.

The training samples are selected from the normal skins and tumors in Fig. 1. The PDFs (Probability Density Function) of normal skin and tumor in the 21st image are depicted in Fig. 2. It can be observed that the PDF of tumor is overlapped with that of normal skin. The probability of error (false detection) is proportional to the overlapped area which is shaded in Figure and it could be very high if only an intensity of a pixel is utilized as a feature.

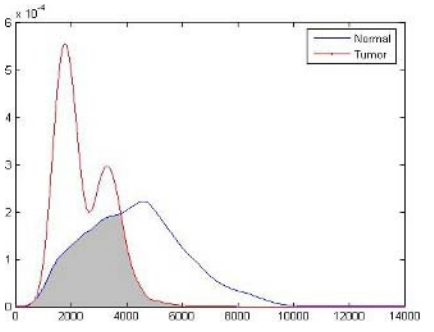


Fig. 2. PDFs of normal skin and tumor in the 21st band image

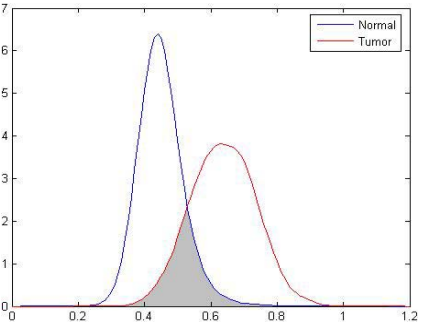


Fig. 3. PDFs of normal skin and tumor in I_{57}/I_{17}

The PDFs of normal skin and tumor are shown in Fig. 3 where the data is obtained from the ratio of the 57th image (I_{57}) to the 17th image (I_{17}). The overlapped area representing the probability of error shrinks. It means that the feature space generated by the ratio of intensities of two bands is preferred to the one by the intensity itself. Now it is necessary to find the combination of images such that their intensity ratios yield the least false detection rate.

Let denote the PDFs of normal skin and tumor as $f_N(n)$ and $f_T(n)$, and the distance between samples as δ_n. Then the overlapped region of two PDFs to be minimized is determined by the following Eq. (1).

$$P_S = 1 - 0.5 \sum_{i=1}^{K} \left| f_N(i) - f_T(i) \right| \delta_n . \tag{1}$$

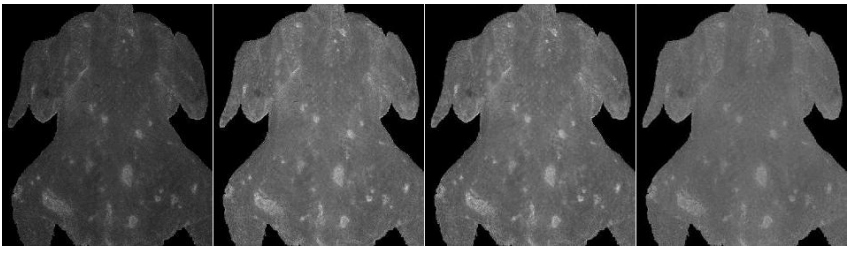


Fig. 4. Feature images (I_{61}/I_{18}, I_{58}/I_{19}, I_{52}/I_{21}, and I_{40}/I_{21}, from left to right)

In order to find the optimal pairs of band images, it is necessary to estimate the PDFs of normal skin and tumor by using training samples. Kernel density estimation (KDE) algorithm is utilized to find the PDFs in Eq. (1). The optimal pairs of bands selected by the least value of the matrix are 61/18, 58/19, 52/21, and 40/21. Fig. 4 depicts the feature image obtained by the above combinations of band images. The background is removed by applying mask. It can be observed that they are better than ones shown in Fig. 1 in discriminating between normal skin and tumor.

2.2 Radial Basis Probabilistic Neural Network

Radial basis probabilistic neural network (RBPNN) can be used for classification problems [6]. The RBPNN model, as shown in Fig. 5, comes from the radial basis function neural network (RBFNN) and the probabilistic neural network (PNN). Hence, it has the advantages of the above two networks while lowering their demerits. As shown in Fig. 5, this network consists of four layers: one input layer, two hidden layers, and one output layer. The first hidden layer is a nonlinear processing layer, generally consisting of selected hidden centers determined by input training set. The second hidden layer corresponds with the first hidden layer, which has generally the same size as the output layer for a labeled pattern classification problem. The second hidden layer makes the selection on the first hidden neurons and sums the selected first hidden outputs for its own output. In general, the corresponding weight values of the second hidden layer are 1s. The last layer is just the output layer.

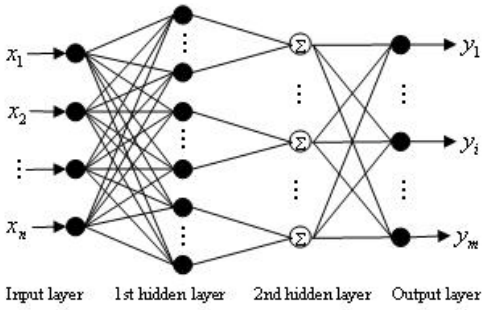


Fig. 5. The structure of radial basis probabilistic neural network model

In mathematics, the *i*-th output of RBPNN can be written as

$$y_i = \sum_{k=1}^{m} w_{ik} h_k(x) \quad i = 1,2,\cdots m. \tag{2}$$

$$h_k(x) = \sum_{i=1}^{n_k} \phi_i(x, c_{ki}) = \sum_{i=1}^{n_k} \phi_i(\|x - c_{ki}\|) \quad k = 1,2,\cdots m. \tag{3}$$

where x is an n-dimensional input vector, $h_k(x)$ is the k-th output value of the first hidden layer; w_{ik} is the weight between the k-th hidden neuron of the second hidden

layer and the *i*-th output neuron; c_{ki} is the *i*-th hidden center vector corresponding to the *k*-th hidden neuron of the second hidden layer; $\|\cdot\|$ denotes the Euclidean norm and $\phi_i(\cdot)$ is the nonlinear kernel function of the first hidden layer. Here, $\phi_i(\|x - c_{ki}\|)$ is a Gaussian kernel function with the mean c_{ki} and variance σ_i. Generally, the shape parameters (σ_i's) for are chosen the same values. In the case of hidden centers being fixed, different shape parameters for kernel functions can lead to different classification performance and generalization capability. The shape parameter is usually set according to the following heuristic relationship [7]. That is $\sigma = d_{\max} / \sqrt{N}$, where $d_{\max}$ is the maximum Euclidean distance among all the training samples, and *N* is the total number of the training samples.

In pattern classification, if a input vector is close to the most hidden center members of some pattern class, the corresponding output value of the pattern class is obviously bigger than the ones of any other pattern classes numerically. The second hidden layer weights are set to the matrix of target vectors, each vector has a 1 only in the row associated with that particular class of input, and 0's elsewhere. Finally, the second hidden layer transfer function, compete, produces a 1 corresponding to the largest element, and 0's elsewhere. Thus, the network has classified the input vector into a specific one of *m* classes because that class had the maximum probability of being correct.

3 Experimental Results

A total of 13 chicken carcasses with skin tumors are collected for experiment. Four feature images are extracted by applying the proposed method described in 2.1 Method for Feature Extraction. They are I_{61}/I_{18}, I_{58}/I_{19}, I_{52}/I_{21}, and I_{41}/I_{21}. The ratios of bands are chosen from the training set to satisfy minimization of the false detection rate.

The input of RBPNN consists of four neurons and the output layer has three neurons. In the training stage, 150 pixels from normal skin (100 pixels) and tumor (50 pixels) are used and three classes are assigned. One class is for tumor and two for normal skin pixels. It is found in the experiment that modifying the two classes into three classes provides the better result. The RBPNN is implemented by the above training sample set and its variance of Gaussian kernel function is obtained by $\sigma = d_{\max} / \sqrt{N}$.

The performance of detecting tumor is mostly corrupted by false detection rate, which is defined as the probability to label normal skin as tumor. Fig. 6 (a) is the results of classification by the proposed method, while Fig. 6 (b) is the result by the method in [5]. In Fig. 6, the black spot represents tumor and the white one does normal skin. The tumor spots are located in Fig. 6 (c) and they are identified by the FSIS (Food Safety and Inspection Service) veterinarian at the plant. Examination of Fig. 6 indicates that the proposed method yields less amount of false alarms and the same result is observed in other chicken carcasses which are not shown in this paper. The false detection rate is defined as the pixels of false classification over the total pixels belong to the area of chicken carcasses and they are 1.3% and 3.6% for Fig. 6 (a) and

(b), respectively. To compare with the result given by [8] where the same data set are used, the proposed method in this paper shows the classification rate of 98.2% while that of 96% is reported in [8].

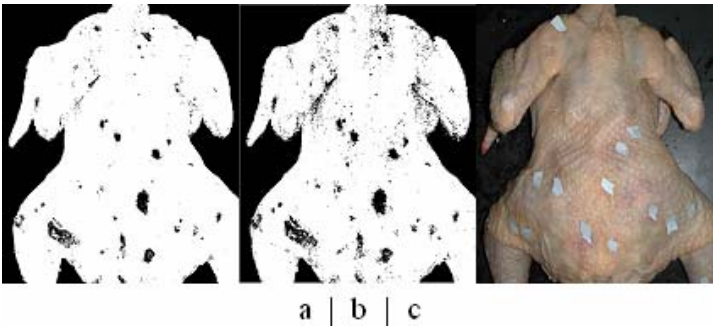


Fig. 6. (a) Classification result by the proposed method (b) Classification result from [5] (c) Tumor spots

4 Conclusion

A method for detecting skin tumors on chicken carcasses for hyperspectral images is presented. Hyperspectral images contain many data which can be characterized as complementary as well as redundant. The proposed detection method consists of two major steps: feature extraction and classification. Feature extraction process reduces the redundancy among the data and finds the complementary band images. The result of feature extraction is a set of the ratios of particular band images. It is presented as an input to a classifier based on the RBPNN. The classifier categorizes three classes rather than two because one class is assigned to tumor skin and two other classes to normal skin. The classification result shows the improved performance in that the number of false classification is reduced. Further research may include the implementation of the algorithm for the hyperspectral reflectance images. The reason for this is the reflectance image needs cheaper and simpler set-ups for the image acquisition for the same purpose.

Acknowledgement

This work is supported by grant No. (R05-2003-000-11268-0) from the Basic Research Program of Korea Science & Engineering Foundation.

References

1. Calnek, B.W., John Barnes, H., Beard, C.W., Reid, W.M., Yoder, H.W.: Diseases of Poultry. Ames, Iowa. Iowa State University Press (1991)
2. Chao, K., Chen, Y.R., Hruschka, W.R., Gwozdz, F.B.: On-line Inspection of Poultry Carcasses by a Dual-camera System. J. Food. Eng. 51(3) (2002) 185-192

3. Chen, Y.R., Park, B., Huffman, R.W., Nguyen, M.: Classification of On-line Poultry Carcasses with Backpropagation Neural Networks. J. Food. Eng. 21(1) (1998) 33-48
4. Kong, S.G., Chen, Y.R., Kim, I., Kim, M.S.: Analysis of Hyperspectral Fluorescence Images for Poultry Skin Tumor Inspection. Applied Optics 43(4) (2004) 824-833
5. Kim, I., Kim, M.S., Chen, Y.R., Kong, S.G.: Detection of Skin Tumors on Chicken Carcasses using Hyperspectral Fluorescence Imaging. Transactions of ASAE 47(5) (2004) 1785-1792
6. Huang, D.S.: Radial Basis Probabilistic Neural Networks: Model and Application. International Journal of Pattern Recognition and Artificial Intelligence 13(7) (1999) 1083-1101
7. Haykin, S.: Adaptive Filter Theory. 3rd edn. Upper Saddle River, NJ, Prentice-Hall (1996)
8. Fletcher, J.T., Kong, S.G.: Principal Component analysis for Poultry Tumor Inspection using Hyperspectral Fluorescence Imaging. Proceedings of the International Joint Conference on Neural Networks 2003. Vol. 1. Doubletree Hotel-Jantzen Beach, Portland, Oregon (2003) 149-153

Combination of Network Construction and Cluster Analysis and Its Application to Traditional Chinese Medicine

Mingfeng Wang[1], Zhi Geng[1], Miqu Wang[2], Feng Chen[1], Weijun Ding[2], and Ming Liu[2]

[1] School of Mathematical Sciences, Peking University, Beijing 100871, China
[2] Chengdu University of Traditional Chinese Medicine, Chengdu 610075, China

Abstract. Bayesian networks and cluster analysis are widely applied to network construction, data mining and causal discovery in bioinformation and medical researches. A Bayesian network is used to describe associations among a large number of variables, such as a gene network and a network describing relationships among symptoms. Cluster analysis is used to cluster associated variables, For example, genes with similar expressions or associated symptoms are grouped into a cluster. In this paper, we combine these approaches of network construction and cluster analysis together. On the one hand, we use Bayesian networks to explain relationships among variables in each cluster; on the other hand we use hierarchical cluster approach to assist network construction, and we propose a structure learning approach. In the stepwise approach, a subnetwork over a larger cluster is constructed by combining several subnetworks over small clusters whenever these small clusters are grouped together. The proposed approach is applied to a traditional Chinese medical study on a kidney disease.

Keywords: Bayesian Network; Cluster analysis; Traditional Chinese medicine; Biostatistics.

1 Introduction

Bayesian networks, also called directed acyclic graphs (DAG), have been widely applied to many fields, such as data mining, pattern recognition, artificial intelligence and causal discovery [1, 2, 3, 5, 6]. Bayesian networks are often used to describe associations among a large number of variables in bioinformatics. For example, a gene network is constructed from microarray data. Hierarchical cluster approach is also widely applied in bioinformatics and medical researches to group a large number of variables into a small number of clusters. For example, genes with similar expressions are grouped into a cluster.

In this paper, we combine approaches of network construction and cluster analysis together. On the one hand, we use Bayesian networks to explain rela-

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 777–785, 2006.

tionships among variables in each cluster. In a cluster analysis, the most correlated variables are initially grouped together, then the correlated clusters are grouped into a larger cluster, and so on. The cluster analysis, however, does not explain why and how these variables are grouped into a cluster at each step. We construct a subgraph for each cluster to explain the association relationships among variables in the cluster. On the other hand we use hierarchical cluster approach to assist network construction. To Construct Bayesian networks from observed data, Verma and Pearl [7] presented the inductive causation (IC) algorithm which searches for a separator S from all possible variable subsets such that two variables u and v are independent conditional on S. The PC algorithm limits possible separators to vertices that are adjacent to u and v [5, 6]. When there are a large number of variables, searching for a separator and statistical test are not efficient. In this paper, we propose a stepwise approach of network structural learning. In the stepwise approach, a larger subnetwork is constructed by combining several small subnetworks whenever small clusters are grouped into a larger cluster. In such a way, a difficult problem of constructing a large network is split into easy problems of constructing small networks. The stepwise approach of combining network construction and cluster analysis is applied to a traditional Chinese medical study on a kidney disease. We obtained a sample of 734 patients. For each patient, 170 variables including symptoms and backgrounds are recorded and 35 symptoms are used to construct a Bayesian network and a hierarchical cluster tree.

In Section 2, we give some notation and definitions of Bayesian networks and mutual information. In Section 3, we propose the stepwise approach of combining cluster analysis and network structural learning. Finally we apply the stepwise approach to a traditional Chinese medical study on a kidney disease.

2 Notation and Definitions

A Bayesian network, also called a directed acyclic graph (DAG), is denoted by a pair $G = (V, E)$ where V is a finite set of nodes and $E \subseteq V \times V$ is a set of ordered node pairs. $< i, j > \in E$ denotes a directed edge of i pointing to j. A directed edge is also called a arrows. If $< i, j > \in E$ and $< j, i > \in E$, an edge between nodes i and j is undirected, denoted by (i, j) and depicted by a line in the network. A network is directed (undirected) if it contains only directed (undirected) edges. In an undirected network, the neighbor set of a node i is defined as a set of nodes that have one edge connecting i in G, noted by $ne(i)$. The skeleton G^s of a Bayesian network is an undirected network that is obtained by removing all directions of arrows in G. If there is a subnetwork $a \rightarrow c \leftarrow b$ for three nodes a, b and c in V (that is, $< a, c > \in E$, $< b, c > \in E$, $< a, b > \notin E$ and $< b, a > \notin E$), we call (a, c, b) a v-structure or immorality. The moral graph G^m of a Bayesian network G is formed by (a) adding an undirected edge between two nonadjacent nodes in each v-structure, and (b) removing directions of all arrows. The added edges between immoralities are called moral edges.

Let $X = (X_1, \ldots, X_p)$ be a p-dimensional vector of random variables. Each variable X_i in X is depicted by a node i in G. If a joint distribution or density of variables $X_1, \ldots, X_p$ satisfies

$$P(x_1, \cdots, x_p) = \prod_{i=1}^{p} P(x_i|pa_i),$$

where $P(x_i|pa_i)$ is the conditional probability or density of X_i given $pa(X_i) = pa_i$, then the Bayesian network $G = (V, E)$ and the distribution P are said to be compatible [5] and P obeys the global directed Markov property of G [4]. In this paper, we assume that all the distributions are compatible with G. We also assume that all independencies of a probability distribution of variables in V can be checked by d-separation of G, called the faithful assumption [6]. The faithful assumption means that all independencies and conditional independencies among variables can be represented by G.

An undirected network $\bar{G}$ represents a family of probability distributions which has the Markov property over the undirected network $\bar{G}$ [4], that is, variables X_i and X_j are conditionally independent given all other variables (denoted by $X_i \perp X_j | X_{V \setminus \{i,j\}}$) if $(i, j) \notin E$.

3 A Stepwise Approach for Constructing a Network

In this section we propose the stepwise approach which combines cluster analysis and network structural learning. In the stepwise approach, we first construct a skeleton which is an undirected network, then we add moral edges to obtain a moral graph and finally we orient the directions of arrows to construct a Bayesian network. A skeleton is constructed along with construction of a hierarchical cluster tree. The cluster tree has each variable as a leaf at the bottom of the tree and an intermediate node denotes a cluster which groups several sub-clusters of its sun nodes. For constructing a skeleton, the stepwise approach initially takes each variable as an undirected subnetwork with only a single node, and then a larger undirected subnetwork is constructed by combining several small undirected subnetworks whenever some sub-clusters are grouped into a larger cluster. The process of constructing a skeleton consists of two main steps: clustering and combining undirected subnetworks. In the following two subsections, we separately describe the two steps. Finally we theoretically prove the validity of the stepwise approach.

3.1 Clustering

We use a hierarchical clustering method for clustering variables rather than objects. Since variables may be discrete, nominal (disordered multi-values) or continuous, we cannot use correlation coefficients, and thus we use the mutual information of two variables X_i and X_j. The mutual information between X_i and X_j is defined as

$$I(X, Y) = \int f(x, y) log \frac{f(x, y)}{f(x)f(y)} dxdy,$$

where $f(x_i)$, $f(x_j)$ and $f(x_i, x_j)$ are the densities of X_i, X_j and (X_i, X_j), respectively. We have that $I(X_i, X_j) \geq 0$, a larger value means a closer distance, and $I(X_i, X_j)$ equals 0 if and only if X_i and X_j are independent. Below we transform the mutual information into a measure of closeness or similarity between X_i and X_j

$$C(i,j) = \begin{cases} \frac{I(i,j)}{\sqrt{I(i,i)I(j,j)}}, & \text{if } I(i,i) \neq 0 \ \& \ I(j,j) \neq 0; \\ 0, & \text{otherwise;} \end{cases}$$

such that $C(i,i) = 1$ for all i. From the definition, we have that $0 \leq C(i,j) \leq 1$. We cluster variables based on the closeness matrix C with elements $C(i,j)$.

3.2 Combining Subnetworks into a Larger Network

We initially take each leaf node in the cluster tree as an undirected subnetwork with only a single variable. Whenever several sub-clusters are grouped into a bigger cluster, we combine the undirected subnetworks of these sub-clusters into a larger undirected subnetwork whose nodes are all variables in these sub-clusters. In the combination, we add some new edges crossing different sub-clusters and delete some edges within each sub-cluster.

Suppose that K sub-clusters are grouped into a bigger cluster at an intermediate node of the cluster tree, and that the K sub-clusters have K undirected subnetworks $\bar{G}_1 = (V_1, E_1), \ldots, \bar{G}_K = (V_K, E_K)$. We first combine two subnetworks $\bar{G}_1$ and $\bar{G}_2$ to construct a network $\bar{G}_{1,2}$, and then sequentially combine other subnetworks to obtain the network $\bar{G}_{1,\ldots,K}$.

Without loss of generality, we consider combination of two undirected subgroups $\bar{G}_1$ and $\bar{G}_2$. First we add some new undirected edges between nodes in two different sub-clusters. Define $S_{ab} = ne_{V_1}(a) \cup ne_{V_2}(b)$ where $a \in V_1$, $b \in V_2$, $ne_{V_1}(a)$ is the neighbor set of a in $\bar{G}_1$ and $ne_{V_2}(b)$ is the neighbor set of b in $\bar{G}_2$. If there is a subset $S \subseteq S_{ab}$ such that $a \perp b|S$, then the undirected edge between a and b is not added, otherwise it is added. After checking all pairs of a node from $\bar{G}_1$ and the other from $\bar{G}_2$, we obtain a larger undirected network $\bar{G}'_{1,2} = (V_1 \cup V_2, E'_{1,2})$.

Next we remove some edges within each subnetwork. An edge (a', a'') of the original subnetwork $\bar{G}_1$ is removed from $\bar{G}_{1,2}$ if a' and a'' may become to be independent given some subset S of all other variables $(V_1 \cup V_2) \setminus \{a', a''\}$ in $\bar{G}_{1,2}$. The above process of removing edges within subnetworks can be postponed until the whole undirected network has been obtained at the end of cluster analysis.

We show below that $\bar{G}_{1,\ldots,K}$ obtained by combining all subnetworks in the above process does not lose any edge among nodes in $V_1 \cup \ldots \cup V_K$ of the skeleton G^s.

Theorem 1. *The undirected network $\bar{G}_{1,\ldots,K}$ obtained by combining subnetworks $\bar{G}_1, \ldots, \bar{G}_K$ in the above method contains all edges in the skeleton G^s of the true directed network G.*

Proof. If there is no edge between nodes a and b, then we find a set $S \subseteq (V_1 \cup \ldots V_K)$ in the above algorithm such that $a \perp b|S$. Thus the edge between a and b should not appear in the skeleton G^s. □

After the cluster analysis finishes, we obtain a cluster tree and an undirected subnetwork at each node of the tree. For an intermediate node of the cluster tree, we have a subnetwork $\bar{G}_{1,\ldots,K}$. No edge between a and b means that a and b are not directly associated, and an edge between a and b means that a and b may associate directly or there may be other variables outside the cluster which associate with both a and b. Thus we can see from the subnetwork the relationships among variables in the cluster, and thus we have more information on association among variables from the network than that from only a cluster tree. Although $\bar{G}_{1,\ldots,K}$ may contains redundant edges, we do a process to remove these redundant edges after combining all clusters together. In the following subsection, we discuss the process for constructing a moral network G^m of G in which redundant edges are removed from the skeleton.

3.3 Constructing a Moral Graph

A moral network G^m is an undirected network from which we can easily justify conditional independencies among all variables. The undirected network finally obtained in the process does not contain moral edges and may contain some redundant edges since, in the undirected network, a neighbor set of a node a may not contain these nodes that should be connected by moral edges. Thus we must first add moral edges to the undirected network and then check again whether each edge can be removed by using the new neighbor sets.

First consider add moral edges. For any pair of nonadjacent nodes (a, b) that have a nonempty set $S = ne(a) \cap ne(b)$ of common neighbor nodes, we add a moral edge between a and b if $a \not\perp b|S$, which implies that there is a node c in S such that a, b and c form a v-structure. By repeating the above process for every pair of nonadjacent nodes, we obtain an undirected network which contains all moral edges, see the following theorem.

Theorem 2. *All moral edges can be identified by the above process.*

Proof. From Theorem 1, the undirected graph $\bar{G}$ contains all edges in the true network G. If $\bar{G}$ does not contain the moral edge between a and b of some v-structure $a \rightarrow c \leftarrow b$, then we have $a \not\perp b|S$ for any S which contains c, and thus the moral edge between a and b can be identified. □

Next remove redundant edges. After adding all moral edges, we have an undirected network which contains all edges and all moral edges of the true network G. But the undirected network may contain redundant edges and we must check every edge again. For each undirected edge (a, b), we remove it if $a \perp b|S$ where $S = ne(a) \cup ne(b)$.

After the above two processes of adding moral edges and removing redundant edges, we obtain the moral network G^m of the true network G. To construct

a Bayesian network, we can apply the IC algorithm [7] or the decomposition algorithm [2] to the moral graph G^m.

3.4 Advantages of the Stepwise Approach

There are several obvious advantages of the stepwise approach for structural learning proposed in this paper. On one hand, we apply Bayesian networks to the cluster analysis to explain the relationships among variables in each cluster. Thus our approach provides more information on relationships among variables than the cluster analysis. On the other hand, we use the hierarchical cluster approach to assist construction of Bayesian networks so that independence tests are performed conditionally on smaller sets rather than on the full set of all other variables. Thus when there are a large number of variables, our approach has higher power for statistical tests.

4 Simulation

In this section, we show a simulation to illustrate the stepwise approach. Suppose that the true network is given in Fig. 1 (a) and that the true conditional probabilities are:

$P(X_1 = 1) = 0.7$, $P(X_2 = 1|X_1 = 0) = 0.4$, $P(X_2 = 1|X_1 = 1) = 0.8$,
$P(X_3 = 1|X_1 = 0) = 0.65$, $P(X_3 = 1|X_1 = 1) = 0.3$,
$P(X_4 = 1|X_2 = 0, X_3 = 0) = 0.1$, $P(X_4 = 1|X_2 = 1, X_3 = 0) = 0.5$,
$P(X_4 = 1|X_2 = 0, X_3 = 1) = 0.5$, $P(X_4 = 1|X_2 = 1, X_3 = 1) = 0.15$,
$P(X_5 = 1|X_4 = 0) = 0.56$, $P(X_5 = 1|X_4 = 1) = 0.25$,
$P(X_6 = 1|X_5 = 0) = 0.9$, $P(X_6 = 1|X_5 = 1) = 0.4$.

We draw 500 observed data from the above probabilities. We first obtain a cluster tree in Fig. 2. Firstly in the cluster tree, two subclusters are $\{X_1, X_2, X_3\}$ and $\{X_4, X_5, X_6\}$. For the subclusters, we separately construct two subnetworks, as shown in Fig. 1 (b). From the subnetworks, it can be seen that X_2 and X_3 are conditionally independent given X_1, and X_4 and X_6 are conditionally independent given X_5. Although X_2 and X_3 do not directly associate each other, X_2 and X_3 are grouped into a cluster since X_1 associates both X_2 and X_3. Secondly these two subclusters are grouped into a larger cluster, and then we combine two subnetworks into a larger network in Fig. 1 (c). Thirdly, we add a moral edge between X_2 and X_3 and obtain a moral network in Fig. 1 (d). Finally, we find an immorality (X_2, X_4, X_3), and then we can orient the edges: X_4 pointing at X_5 and X_5 pointing at X_6 to avoid new immorality, as shown in Fig. 1 (e). The directions of edges between X_1 and X_2 cannot be determined uniquely, although they must be oriented as one of $X_2 \leftarrow X_1 \rightarrow X_3$, $X_2 \leftarrow X_1 \leftarrow X_3$ or $X_2 \rightarrow X_1 \rightarrow X_3$ such that no immorality appears. We obtained the same cluster tree and networks for 50 simulations. This illustrates the good performance obtained by the proposed approach, and further we show a valuable application of the approach in the next section.

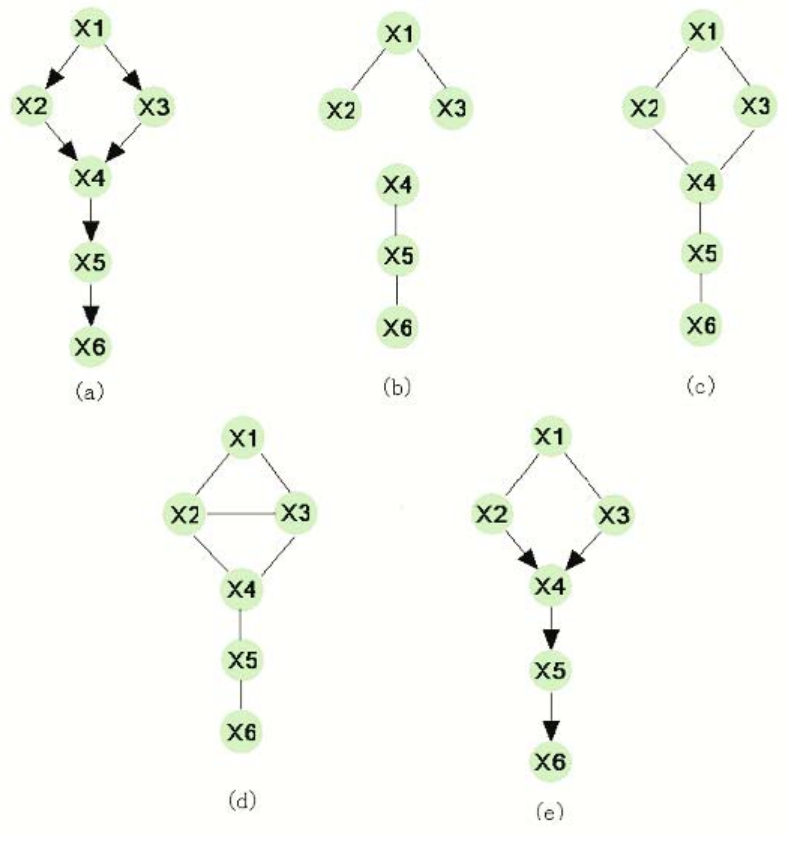


Fig. 1. Networks

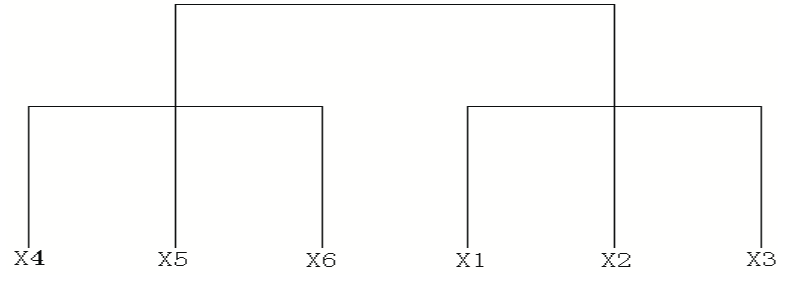


Fig. 2. A hierarchical cluster tree

5 Application to a Traditional Chinese Medical Study on Kidney Disease

In this section, we show an application of our approach to a traditional Chinese medical study on kidney disease. There are 734 patients in the study. For each

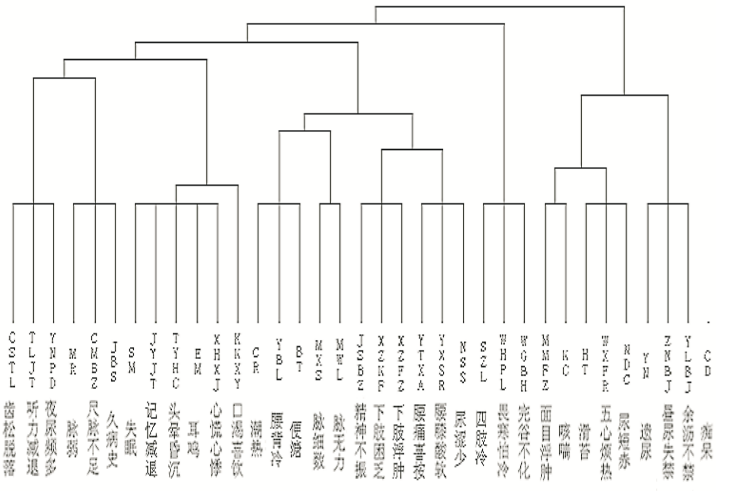


Fig. 3. Clusters of symptoms

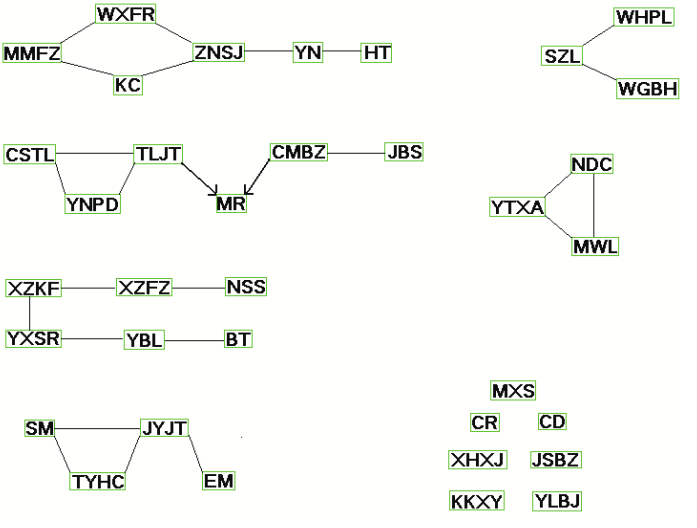


Fig. 4. The network constructed with the stepwise approach

patient, 170 variables are recorded including patients' backgrounds and symptoms. We select 35 main symptoms to study the relationships and associations among them. The result of cluster analysis is shown in Fig. 3, and the constructed network is given in Fig. 4.

Acknowledgement

This research was supported by NSFC and NBRP 2003CB715900.

References

[1] Cowell, R.G., David, A.P., Lauritzen, S.L., Spiegelhalter, D.J.: Probabilistic Networks and Expert Systems, Springer Publications, New York (1999)
[2] Geng, Z., Wang, C., Zhao, Q.: Decompsition of Search for v-Structures in DAGs. J. Multivar. Analy. **96** (2005) 282-294
[3] Heckerman, D.: A Tutorial on Learning with Bayesian Networks. In:Jordan, M. I. (ed.): Learning in Graphical Models. Kluwer Academic Pub., Netherlands (1998) 301–354
[4] Lauritzen, S.L.: Graphical Models. Oxford University Press, Oxford (1996)
[5] Pearl, J.: Causality. Cambridge University Press, Cambridge (2000)
[6] Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction and Search. 2nd edn. MIT Press, Cambridge (2000)
[7] Verma, T., Pearl, J.: Equivalence and Synthesis of Causal Models. In: Bonissone, P., Henrion, M., Kanal, L.N., Lemmer, J.F. (eds.): Uncertainty in Artificial Intelligence, Vol. **6**. Elsevier, Amsterdam (1990) 255–268
[8] Akaike, H.: On Entropy Maximization Principle. In: Krishnaiah, P.R. (ed.): Application of Statistics, Amsterdam, North-Holland (1976) 27–41

Differentiation of Syndromes with SVM

Zhanquan Sun, Guangcheng Xi, and Jianqiang Yi

Key Laboratory of Complex Systems and Intelligence Science,
Institute of Automation, Academy of Sciences, 100080, Beijing, China
{zhanquan.sun, guangcheng.xi, jianqiang.yi}@mail.ia.ac.cn

Abstract. Differentiation of syndromes is the kernel theory of Traditional Chinese Medicine (TCM). How to diagnose syndromes correctly with scientific means according to symptoms is the first problem in TCM. Several modern approaches have been applied, but no satisfied results have been obtained because of the complexity of diagnosis procedure. Support Vector Machine (SVM) is a new classification technique and has drawn much attention on this topic in recent years. In this paper, we combine non-linear Principle Component Analysis (PCA) neural network with multi-class SVM to realize differentiation of syndromes. Non-linear PCA is used to preprocess clinical data to save computational cost and reduce noise. The multi-class SVM takes the non-linear principle components as its inputs and determines a corresponding syndrome. Analyzing of a TCM example shows its effectiveness.

1 Introduction

Traditional Chinese Medicine (TCM) is the treasure of China with about 3000 years' history. It is the only existing systematic classic medicine all over the world now. But most people prefer western medicine and take TCM as experience science. It is unassailable that TCM has better curative effect on many stubborn and chronic diseases than western medicine. So it attracts more and more researchers to study TCM with modern techniques. Differentiation of syndromes is a method of understanding and diagnosing diseases by the theories of TCM. A syndrome is a TCM diagnosis conclusion, which comes from an assessment of all information gathered from four diagnostic techniques: inquiry, inspection, smelling and palpation. In this diagnosis process, factors of age, gender, constitution, season, weather, geographical area, as well as emotion, stress, diet, lifestyle and all aspects of symptoms and signs (tongue picture and pulse) of the body are taken into consideration to form a syndrome for a patient. The treatment strategy is given according to the syndrome. So correct diagnosis of a syndrome is very important in TCM. In the past, differentiation of syndromes is diagnosed according to the doctor's experience. Different doctor maybe diagnose different syndromes to the same patient because of complicated relation between symptoms and a syndrome. How to find the just syndrome with scientific means is a crucial problem in TCM field. Many techniques have been developed, such as cluster, fuzzy cluster, BP network, RBF network and so on[1-2]. But the classification results of all these methods can't meet with practice requirement.

Support Vector Machine (SVM) developed by Vapnik[3] is a new classification technique and has gained popularity due to many attractive features and promising

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 786 – 791, 2006.

empirical performance. SVM is a powerful supervised learning algorithm belonging to the statistical learning theory. It minimizes the structural risk performance in various classification problems. In many applications, SVM has been shown to provide higher performance than traditional learning machines and has been introduced as powerful tools for solving classification problems [4]. Furthermore, SVM is suitable to analyze small sample problems. In SVM, it will cost more computational time if there are too much feature variables. In view of practical requirement, the source data should be preprocessed. Principle Component Analysis (PCA) is a multivariate statistical method. PCA extracts the modes in a set of variables. Its main purpose is to reduce the dimensionality of the dataset by retaining only the first few modes. But PCA is a linear method and implies a potential oversimplification of the datasets being analyzed. The advent of non-linear PCA can overcome the shortcomings. Nonlinear PCA has been applied in many kinds of field, which is usually approximated by a neural network [5-6].

2 Nonlinear PCA Neural Network

Non-linear PCA can be realized with neural network. Let $\mathbf{x}$ denote an L-dimensional random vector. A non-linear neuron model is configured with weight vector $\mathbf{w}$, input sample $\mathbf{x}$ and non-linear output neuron $z = f(\mathbf{w}^T\mathbf{x})$. A simple reconstruction of input vector $\mathbf{x}$ is represented as $\hat{\mathbf{x}} = z\mathbf{w}$. The largest non-linear principal component can be achieved by optimizing the reconstruction mean-square error, i.e., the following objective:

$$\min \quad J = E\{\|\mathbf{x} - \hat{\mathbf{x}}\|^2\} = E\{\|\mathbf{x} - z\mathbf{w}\|^2\} \tag{1}$$

The optimum solution $z = f(\mathbf{w}^T\mathbf{x})$ is the largest nonlinear principle component, f is a non-linear mapping function. Here we choose f as a sigmoidal function bounded between 0 and 1.

A stochastic approximation approach will lead to the following learning rule

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \mu_t(e_t^T\mathbf{w}_t\mathbf{x}g + z\mathbf{e}_t) \tag{2}$$

where $e_t = \mathbf{x} - z\mathbf{w}_t$ is the reconstruction error, g is the derivative of z, and $g = f'(\mathbf{w}_t^T\mathbf{x})$. The learning rule Eq. (2) can be further extended to a network with M neurons. In this case, the network can be treated as autoencoder, which has equal number of L input and output nodes and M hidden nodes, $M < L$. During learning, the inputs are duplicated at the output nodes to perform identity mapping. $L \times M$ matrix W, $M \times L$ matrix $\hat{\mathrm{W}}$ denote the connection weight from input layer to hidden layer and from hidden layer to output layer respectively. In our work, we only consider the symmetrical case, i.e., $\hat{\mathrm{w}} = \mathrm{w}^T$. The structure of non-linear PCA neural network is shown as in Fig. 1. In this structure, the nonlinear principle components are a vector $\boldsymbol{z} = (z_1, z_2, \cdots, z_M)$ with M elements. The larger is the number M, the less will be the reconstruction error, and meanwhile the more computational time will be cost. So it should be determined according to practical requirement.

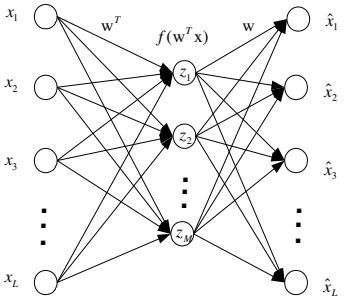


Fig. 1. Structure of nonlinear PCA neural network

3 Support Vector Machine

SVM first maps the input points into a high-dimensional feature space and finds a separating hyperplane that maximizes the margin between two classes in this space. Maximizing the margin is a quadratic programming problem and can be solved from its dual problem by introducing Lagrangian multipliers. Without any knowledge of the mapping, the SVM finds the optimal hyperplane by using the dot product functions in feature space that are called kernels. Many modern optimization methods have been developed and can be used now [7]. The solution of the optimal hyperplane can be written as a combination of a few input points that are called support vectors.

Given a training dataset (x_i, y_i), for $i = 1,2,\cdots,n$ and $x_i \in R^d$ and $y_i \in \{1,-1\}$ and a non-linear transformation to a higher dimensional space (the feature space) $(\Phi(\cdot), R^d \overset{\Phi(\cdot)}{\rightarrow} R^H)$, the SVM solves:

$$\begin{aligned} &\min_{w,\xi_i,b} \quad \left\{ \frac{1}{2}\|\mathrm{w}\|^2 + C\sum_i \xi_i \right\} \\ &\text{s.t.} \quad y^i(\Phi^T(\mathrm{x}_i)\mathrm{w} + b) \geq 1 - \xi_i, \qquad \xi_i \geq 0 \quad \forall i = 1,\cdots,n \end{aligned} \tag{3}$$

where **w** and b define a linear classifier in the feature space, and $\xi_i, i = 1,2,\cdots,n$ are relaxation variables and $\Phi(\cdot)$ is mapping function. The optimization of the problem ensures that the solution is maximum margin. By introducing Lagrangian multipliers, the optimization problem can be transformed into its dual problem.

$$\begin{aligned} &\min_{\alpha} \quad \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_i K(x_i, x_j) - \sum_{j=1}^{n} \alpha_j \\ &\text{s.t.} \quad \sum_{i=1}^{l} y_i \alpha_i = 0 \qquad 0 \leq \alpha_i \leq C, i = 1,2,\cdots,n \end{aligned} \tag{4}$$

where $\alpha_i, i = 1,2,\cdots,n$ are Lagrangian coefficients, $K(x_i, x_j) = (\Phi(x_i) \cdot \Phi(x_j))$ is kernel function and C is penalty coefficient prescribed according to practical requirement. The solution of the dual problem is much easier than the original problem.

A classification task usually involves training and testing data that consist of some data instances. Each instance in the training set contains one "target value" (class label) and several "attributes" (features). The goal of SVM is to produce a model which predicts target values of data instances in the test set, for which only the attributes are given. After obtaining optimum solution α^*, b^*, the following decision function is used to determine which class the sample belongs to.

$$f(x) = \mathrm{sgn}\left(\sum_{i=1}^{l} y_i \alpha_i^* K(x_i, x) + b^* \right) \tag{5}$$

It is important to choose an appropriate kernel function to SVM. Kernel function must satisfy the Mercer theorem. Many kinds of kernel functions have been developed. In this paper, we choose the most commonly used radial basis function

$$K(x_i, x_j) = \gamma \exp(\left| x_i - x_j \right|)^2 \tag{6}$$

where γ, r are kernel parameters.

SVM has been originally designed for two-class classification problems. But many practical problems are multi-class. Several methods have been proposed for multi-class SVM in recent years. In this paper, we adopt a commonly used one. The approach to solve *k*-class classification problem is to construct *k* binary SVM classifiers. Each of them is for one class. The *i*th SVM classifier will be trained with all the examples in the *i*th class with positive labels and all other examples with negative labels. The *i*th SVM classifier constructs a separating hyperplane between class i and the other classes. The multi-class SVMs trained in this way are called 1-*v*-*r*. The conclusion is given with the class that corresponds to the SVM with the highest output value

$$g^j(x) = \sum_{i=1}^{l} y_i \alpha_i^j K(x_i, x) + b^j, \; j = 1,2,\cdots,k \tag{7}$$

In this paper, we combine non-linear PCA neural network and multi-class SVM mentioned above together. The source data are preprocessed by the non-linear PCA neural network at first. Both the dimensionality and noise of source data are reduced. The non-linear principle components are used as the inputs of multi-class SVM for classification. Through the combination, it can not only save classification time but also improve the correct rate of classification.

4 TCM Example

We have 1022 patients' clinic data collected by Chinese Academy of Traditional Chinese Medicine. In these clinic data, 71 symptom variables are recorded. All these symptoms have been quantified. Some symptoms have two values denoted by 0 and 1, and some have four values denoted by 0, 1, 2 and 3. For consistence, some real number variables, such as age, height, are divided into several sects and denoted by 0, 1, 2 and 3. All symptoms are listed in Table 1. There are five different syndromes among these clinic data. They are Syndrome of blood stasis due to deficiency of *qi*, stagnation of *qi* and blood stasis, blood stasis due to deficiency of *yang*, mutual

obstruction of phlegm and stasis, and obstruction of collaterals by stagnant blood. The syndrome is denoted by the class variable whose five different values correspond to the five different syndromes respectively.

Table 1. Symptom variables

Serial number	Symptoms	Serial number	Symptoms	Serial number	Symptoms
1	Gender	25	Tinnitus	49	Walk unsteadiness
2	Age	26	Loss of memory	50	Hemianesthesia
3	Nation	27	Insomnia	51	Acroanesthesia
4	Height	28	Drowsiness	52	Oral lip Anaesth
5	Weight	29	Painful spasm of nape	53	Thirst
6	Occupation	30	Facial distortion	54	Thin
7	Life style	31	Aphasia	55	Five feverish centres
8	Appetite hot	32	Angina pectoris	56	Polyorexia
9	Smoking	33	Chest complaint	57	Anorexia
10	Drinking	34	Chest distress	58	Diuresis
11	Psyche status	35	Cardiopalmus	59	Incontinence of urine
12	Irritable tantrum	36	Hypochondrium distending pain	60	Swell lymph
13	Fatigued and weak	37	Abdominal tenderness	61	Joint pain
14	Chilly	38	Vena epigastrica	62	Vein abnormity
15	Developmental condition	39	Rebound tenderness	63	Cacomelia
16	Operation	40	Abdominal mass	64	Swelling of limb
17	External injury	41	Nausea and vomitting	65	Red tongue
18	Palatine mucosa syndrome	42	Hemiplegia	66	Tongue ecchymosis
19	Dark gum	43	Animal force	67	Varicose lingual vessels
20	Dark labia oris	44	Muscular tension	68	Clavus
21	Scaly skin	45	Dark circles under the eyes	69	Pulses unsmooth
22	Headache	46	Dark face	70	Wiry
23	Dizziness	47	Skin ecchymosis	71	Acrotism
24	Vertigo	48	Patho-sign		

In this example, 855 clinic data are chosen to train a SVM, and the remaining are used to test. The 71 symptom variables are used as input of nonlinear PCA neural network. The kernel function is chosen as sigmoid function. The neural unit number of output is 25, which can be determined according to practical requirement. The 25 nonlinear principle components are used to diagnose a syndrome with the SVM. The setting of the SVM is: penalty factor $C = 20$, kernel function is chosen as RBF function and the width is set as $\sigma^2 = 0.1$.

After training with the method developed in this paper, the test result is: 127 clinic data out of 167 test samples can be classified correctly. The correct rate is 76.05%, which is much higher than RBF neural network used in [8]. We find that it is suitable

to analyze TCM problems with SVM classifier. It can provide an objective reference for differentiation of syndromes in clinic.

5 Conclusion

In this paper, we combine non-linear PCA neural network with multi-class SVM for differentiation of syndromes. Non-linear PCA can reduce dimension and get rid of some noise of source data. Through the application on TCM example, we find the results obtained in this paper are better than other methods. Particularly, SVM is superior to other methods when sample size is small. It is suitable to solve the differentiation of syndromes with multi-class SVM. In TCM, many other non-linear classification problems can also be solved with SVM, which are our further study.

Acknowledgement

The research was supported by National Basic Research Program of China (973 program) under grant No. (2003CB517106), and MOST Projects (No. 2004DFB02100).

References

1. Huang, H.Z., Wu, W.D., Liu, C.S.: A Coordination Method for Fuzzy Multi-objective Optimization of System Reliability. Journal of Intelligent and Fuzzy Systems 16 (2005) 213-220
2. Huang, H.Z., Li, Y.H., Xue, L.H.: A Comprehensive Evaluation Model for Assessments of Grinding Machining Quality. Key Engineering Materials 291-292 (2005) 157-162
3. Cortes, C., Vapnik, V.: Support Vector Networks. Machine Learning, 20 (1995) 273–297
4. Burges, J.C.: A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery 2 (1998) 121–167
5. Xu, L.: Least Mean Square Error Reconstruction Principle for Self-organization. Neural Networks 6 (1993) 627-648
6. Karhunen, J., Joutsensalo J.: Representation and Separation of Signals Using Nonlinear PCA type learning. Neural Networks 7 (1994) 113-127
7. Huang, H. Z., Tian, Z. G., Zuo, M. J.: Intelligent Interactive Multiobjective Optimization Method and Its application to Reliability Optimization. IIE Transactions 37 (2005) 983–993
8. Bian, Q., He, Y. M., Shi, X. C., Wang, H. W.: Neural Network Model of TCM Syndromes Based on MFBP Learning Algorithm. Chinese Journal of Basic Medicine in Traditional Chinese Medicine 7 (2001) 66-69

Neural Network Based Posture Control of a Human Arm Model in the Sagittal Plane

Shan Liu, Yongji Wang, and Jian Huang

Department of Control Science and Engineering,
Huazhong University of Science and Technology,
Wuhan, Hubei, 430074, China
shanshanliu222@163.com

Abstract. In this paper posture control of a human arm in the sagittal plane is investigated by means of model simulations. The arm is modeled by a nonlinear neuromusculoskeletal model with two degrees of freedom and six muscles. A multilayer perceptron network is used in this paper, and effectively adapted by Levenberg-Marquardt training algorithm. The duration of next movement is regulated according as current feedback states. Simulation Results indicate that this method can maintain two joints at different location in allowable bound. The control scheme provides novel insight into neural prosthesis control and robotic control.

1 Introduction

There is a widespread research trying to model and understand the biological motor control system. Hierarchical model [7][10], neural network [9][7], optimal control [1], adaptive control theory [3][8], iterative learning and combination of these methods [11] are applied in motor control of the human arm. However, so far few research on the human mechanical system which, working in a gravity field, is reported.

Neural networks are more and more applied in the study of motor control of musculoskeletal systems [7][8][9]. The goal of the study is to carry out posture control of a human arm in the sagittal plane via a neural network controller.

2 Problem Description

The structure of the neuromusculoskeletal control system is shown in Fig. 1. The input variables are the desired posture $\mathbf{q}_d(t) = [\theta_{d1}(t) \quad \theta_{d2}(t)]^T \in R^2$ and the external force $\mathbf{F}_e(t) \in R^2$. The output variables are the current posture $\mathbf{q}(t) = [\theta_1(t) \quad \theta_2(t)]^T \in R^2$. The state variables are neural control signals $\mathbf{u}(t) \in R^6$ and muscle forces $\mathbf{F}_m(t) \in R^6$.

The skeletal system consisting of two links and two revolute joints is moved by four lumped monoarticular and two biarticular muscles. A schematic picture of the musculoskeletal model is shown in Fig. 2.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 792–798, 2006.

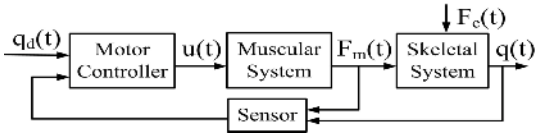


Fig. 1. Representation of a neuromusculoskeletal control system for posture control

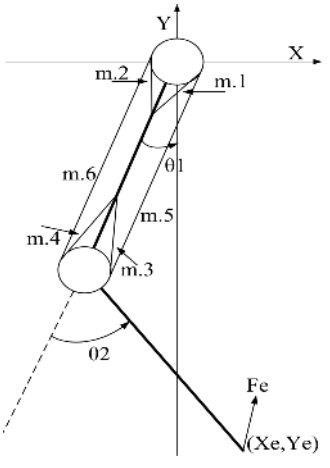


Fig. 2. Schematic picture of the arm model

The equations (1) - (3) describe the musculoskeletal system. Passive elasticities are assumed to restrict the movement range of the shoulder from $-80°$ to $90°$ and of the elbow from $0°$ to $150°$.

$$\mathbf{M}_q(\mathbf{q};t)\ddot{\mathbf{q}}(t) + \mathbf{C}_q(\mathbf{q},\dot{\mathbf{q}};t) + \mathbf{G}_q(\mathbf{q};t) = \mathbf{T}_m(t) + \mathbf{T}_e(t) + \mathbf{T}_p(\mathbf{q};t) \qquad (1)$$

The left side of Eq. (1) describes the dynamics of the skeletal system, in which $\mathbf{M}_q(t) \in R^{2\times 2}$ is the joint inertia matrix, $\mathbf{C}_q(t) \in R^2$ is the vector of centrifugal and Coriolis torques, and $\mathbf{G}_q(t) \in R^2$ is the vector of gravity torques. The right side of Eq. (1) represents the input of the skeletal system, in which $\mathbf{T}_m(t)$ is the net torque generated by the active muscle forces, $\mathbf{T}_e(t)$ is the external torque, and $\mathbf{T}_p(t)$ is the passive torque. In Eq. (2), the 2 by 6 matrix of muscle moment arms $\mathbf{R}_m(t)$ is used to compute the net torques. The jacobian matrix $\mathbf{R}_e(t)$ describes joint torques in equivalent endpoint forces.

$$\begin{aligned} \mathbf{T}_m(t) &= \mathbf{R}_m(\mathbf{q};t) \cdot \mathbf{F}_m(\mathbf{a}, \mathbf{l}(\mathbf{q}), \dot{\mathbf{l}}(\mathbf{q},\dot{\mathbf{q}});t); \\ \mathbf{T}_e(t) &= \mathbf{R}_e^T(\mathbf{q};t) \cdot \mathbf{F}_e(t) \end{aligned} \qquad (2)$$

The applied muscle model is a simplified version of the model proposed by [2]. The active muscle force $F_{mi}(a,l,\dot{l};t)$ $(i = 1,...6)$ is produced by contractile element. For simplicity, $R_i = 1$ and $Y_i = 1$ for $i = 1,...6$ are assumed.

$$F_{mi}(a_i, l_i, \dot{l}_i; t) = F_{0i} R_i A f_i(a_i, l_i(\mathbf{q}), Y_i) F_{li}(l) F_{vi}(l_i(\mathbf{q}), \dot{l}_i(\mathbf{q}, \dot{\mathbf{q}})) \quad (3)$$

Muscle activation a is reasonably well modeled with a first order nonlinear filter of the form $\dot{a} = (u - a)/t(u, a)$, where $t = t_{deact} + u(t_{act} - t_{deact})$ when $u > a$, and $t = t_{deact}$ otherwise [10].

3 Neural Network Control

3.1 Structure

In view of the successful applications of neural networks on the human arm movement in the horizontal plane [7][9], we adopt a multilayer perceptron (MLP) network. The feedforward control component represents the inverse dynamics of the musculoskeletal system. Error due to imperfect feedforward control and external disturbances can be compensated by feedback control.

In this study, the controller is represented by a MLP network with the 10 by 12 by 6 structure, where $\mathbf{s}(t) = [\mathbf{r}(t)^T \quad 1]^T$, $\mathbf{r}(t)$ is the network input, $\mathbf{u}(t)$ is the network output, $\mathbf{W}_1$ and $\mathbf{W}_2$ are weight matrices containing the bias vector, and Γ_1 and Γ_2 are arrays of sigmoid functions $\gamma(x) = 1/[\exp(-x) + 1]$.

$$\mathbf{u}(t) = \Gamma_2\{\mathbf{W}_2 \Gamma_1[\mathbf{W}_1 \mathbf{s}(t)]\} \quad (4)$$

The output vector $\mathbf{u}(t)$ specifies the neural input signals of six muscles. The input vector $\mathbf{r}(t)$ ($\mathbf{r}(t)^T = [\mathbf{q}_d(t) \quad \mathbf{q}_d(t-\tau) - \mathbf{q}(t-\tau) \quad F_m(t-\tau)]$) specifies the desired posture $\mathbf{q}_d(t)$ and contains feedback signals from the musculoskeletal system. It is assumed that the distributed sensors provide information after a loop delay τ.

3.2 Algorithm

The weights of the controller are adapted by Levenberg-Marquardt algorithm. The goal of the learning procedure is to minimize a cost function J (Eq. (5)), with T the evaluation time, V_i the volume of muscle i, V_{av} the average volume of all six muscles, and α a weighting parameter determining the relation between the angle deviations and the muscular activation term. In the case of posture control, this cost function implies the search for a combined solution for both the inverse dynamics and the optimal distribution of force between the various muscles [9].

$$J = \frac{1}{2} \int_0^\tau \left\{ \sum_{j=1}^{2} [\theta_j(t) - \theta_{dj}(t)]^2 + \alpha \sum_{i=1}^{6} \frac{V_i}{V_{av}} a_i(t)^2 \right\} dt, \tau \in [0, T] \quad (5)$$

The learning process commences by choosing small random initial weights and situating the arm in the rest posture. The initial activation of one muscle

is almost zero. Following this initialization, the system is either desired to reach a series of desired postures, or to stay still while an external force acts on the endpoint. After each time step passed, the cost J (Eq. (5)) and the cost gradient with respect to the neural weight vector $\nabla_{\mathbf{w}} J$ are calculated. The weight renewal equation is represented in Eq. (6), where η is a learning constant. Hessian matrix $H^{(k)}$ is approximated by $H^{(k)} = \left(\nabla_{\mathbf{w}^{(k)}} J\right)^T \left(\nabla_{\mathbf{w}^{(k)}} J\right)$. $\lambda^{(k)}$ a scalar is large at the beginning of learning, decreased after each successful step and increased when a tentative step would increase the cost function.

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \eta(H^{(k)} + \lambda^{(k)} I)^{-1} \nabla_{\mathbf{w}^{(k)}} J \tag{6}$$

The evaluation time T is adjusted for each control process according to the approach proposed by [5].

$$T^{(k)} = \beta \sqrt{\sum_{t=0}^{T^{(k-1)}} \mathbf{q}(t)^T \mathbf{q}(t)} + T_{\min}. \tag{7}$$

4 Results

The neuromusculoskeletal model is constructed in Simulink by Virtual Muscle [4], and all control procedures are implemented in Matlab. Two classes of learning situations are considered in this paper.

1. The system is required to pursue minimum variance movements without external forces acting on the endpoint.
2. The system is demanded to hold a constant reference posture with endpoint coordinates while an constant external force acts on the endpoint.

4.1 Simulation Results

Here we illustrate the optimal performance found by the network controller. The optimal trajectories of the joint angles θ_1 and θ_2 are shown in Fig. 3. We find that the shoulder angle and elbow angle arrive to the desired posture $\theta_1 = 10°, \theta_2 = 20°$ from the initial posture $\theta_1 = -4°, \theta_2 = 14°$, respectively. The process is shown in Fig. 4. Fig. 5 illustrates the fluctuation of the rest arm posture under the action of $9.8N$ paralleling with the ordinate, solid lines figure responses under our controller, and dashed lines are without controller.

4.2 Compared with Existing Algorithm

Existing algorithms for neural networks can be classified in two groups, based respectively on heuristic techniques and standard numerical optimization techniques [6]. We chose resilient backpropagation algorithm and Levenberg-Marquardt algorithm to train our controller. Judging from the runtime (shown in Table 1), Levenberg-Marquardt algorithm is better.

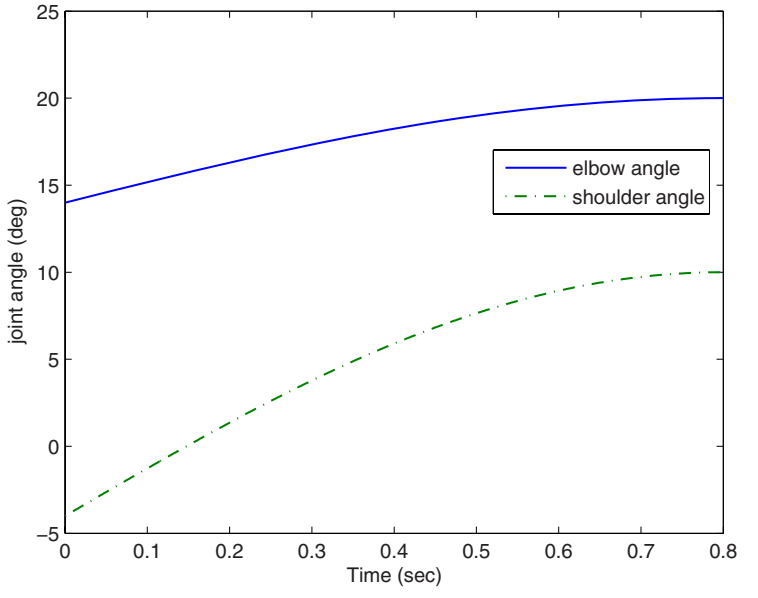


Fig. 3. Optimal trajectories of joint movement

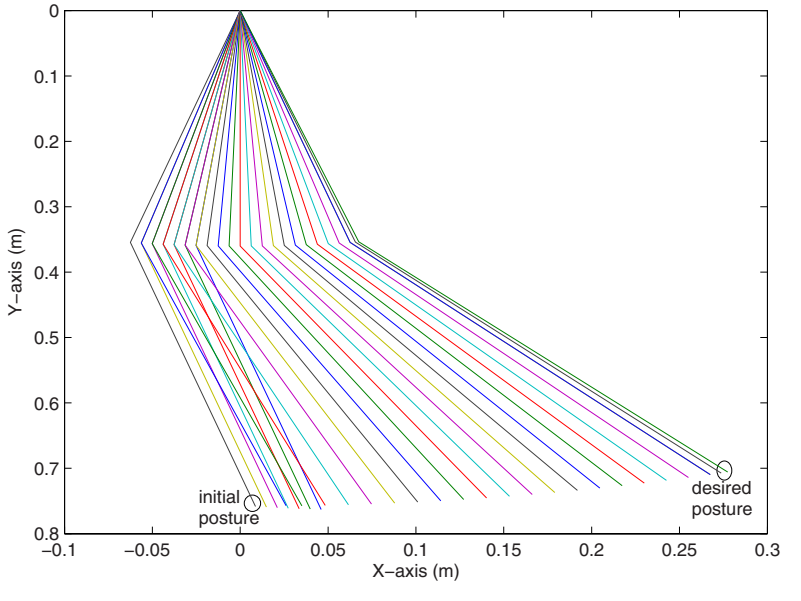


Fig. 4. Stick diagram of the movement without external forces in 0.8sec

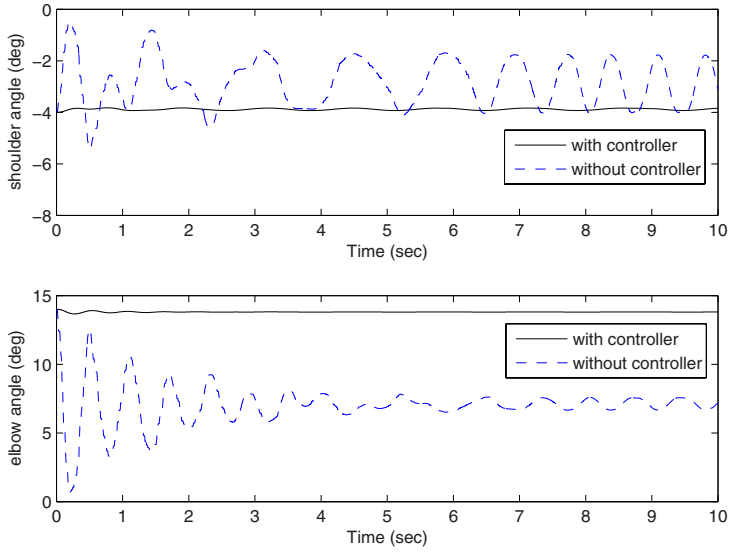


Fig. 5. Response of model under the action of $9.8N$

Table 1. Comparison of Two Algorithms

Algorithm	Runtime(sec)	Iteration
Resilient Backpropagation	603.40	3562
Levenberg-Marquardt	19.54	53

5 Conclusion

Here we applied a MLP network to posture control of the human arm in the sagittal plane. The controller is trained by Levenberg-Marquardt algorithm, and the duration of next movement is regulated according as current feedback states. Simulation Results indicate that the motor control system can maintain two joints at different location within allowable bound. The control scheme can be used in neural prosthesis control and control of a robotic assisted upper extremity.

6 Further Works

In this paper, we only studied the arm motor control in the sagittal plane. Therefore there are several possible extensions in our future works:

1. We may adopt global methods, such as simulated annealing algorithm, and genetic algorithm, to resolve the local minimum problem.
2. Neural networks combining with iterative learning may get hold of better control performance.
3. We should make use of the experience of previous works to further study the impedance characteristics of the human arm in the sagittal plane.

Acknowledgement

This work is supported by grants from the National Nature Science Foundation of China, No 60274020 and No 60340420431, and the Ph.D. Programs Foundation of Ministry of Education of China, No 20050487013. The authors would like to thank Prof. Jiping He for the guidance regarding muscular control and providing the arm model in Simulink.

References

1. Schouten, A.C., de Vlugt, E., van der Helm, F.C.T., Brouwn, G.G.: Optimal Posture Control of a Musculo-Skeletal Arm Model. Biol. Cybern. **84** (2001) 143-152
2. Brown, I. E., Cheng, E. J., and Leob, G.: Measured and Modeled Properties of Mammalian Skeletal Muscle. II. the Effects of Stimulus Frequency on Force-Length and Force-Velocity Relationships. J. of Muscle Research and Cell Motility **20** (1999) 627-643
3. Franklin, D.W., Milner, T.E.: Adaptive Control of Stiffness to Stabilize Hand Position with Large Loads. Exp. Brain Res. **152** (2003) 211-220
4. Cheng, E.J., Brown, I.E., Loeb, G.E.: Virtual Muscle: a Computational Approach to Understanding the Effects of Muscle Properties on Motor Control. Journal of Neuroscience Methods **101** (2000) 117-130
5. Katayama, M., Inoue, S., Kawato, M.: A Strategy of Motor Learning Using Adjustable Parameters for Arm Movement. In: Chang, H.K., Zhang, Y.T. (eds.): Proceeding of 20th IEEE EMBS. IEEE Engineering in Medicine and Biology Society, Vol. 20. IEEE Press, New Jersey (1998) 2370-2373
6. Hagan, M. T., Demuth, H. B., and Beale, M. H.: Neural Network Design. Boston, MA: PWS Publishing (1996)
7. Lan, N.: Analysis of an Optimal Control Model of Multi-Joint Arm Movements. Biol. Cybern. **76** (1997) 207-117
8. Sanner, R.M., Kosha, M.: A Mathematical Model of the Adaptive Control of Human Arm Motions. Biol. Cybern. **80** (1999) 369-382
9. Stroeve, S.H.: Impedance Characteristics of a Neuromusculoskeletal Model of the Human Arm. I. Posture Control. Biol. Cybern. **81** (1999) 475-494
10. Li, W., Todorov, E., Pan, X.: Hierarchical Optimal Control of Redundant Biomechanical Systems. In: Hudson, D., Liang, Z.P. (eds.): Proceeding of 26th IEEE EMBS. IEEE Engineering in Medicine and Biology Society, Vol. 26. IEEE Press, New Jersey (2004) 4618-4621
11. Todorov, E., Li, W.: A Generalized Iterative LQG Method for Locally-optiml Feedback Control of Constrained Nonlinear Stochastic Systems. In: Suhada, J., Balakrishnan, S.N. (eds.): Proceeding of 2005 ACC. American Control Conference, Vol. 1. IEEE Press, New Jersey (2005) 300-306

Neural Network Models for Transforming Consumer Perception into Product Form Design

Chung-Hsing Yeh[1] and Yang-Cheng Lin[2]

[1] Clayton School of Information Technology, Monash University,
Victoria, Australia
Graduate Institute of Finance and Banking, National Cheng Kung University, Taiwan
ChungHsing.Yeh@infotech.monash.edu.au
[2] Department of Fine Arts Education, National Hualien University of Education,
Hualien, 970, Taiwan
lyc0914@cm1.hinet.net

Abstract. This paper presents a number of neural network (NN) models for examining how a given product form affects product images perceived by customers. An experimental study on mobile phones is conducted. The concept of consumer oriented design is used to extract the experimental samples as a design database for the numerical analysis. The result of the experiment demonstrates the advantages of using NN models for the product form design. NN models can help product designers understand consumers' perception and translate consumers' feeling of a product into design elements.

1 Introduction

Intelligent systems are developed to possess humanlike expertise within a specific domain, adapt themselves and learn to do better in changing environments, and explain how they make decisions [5]. Neural network (NN) models are widely used to examine the complex relationship between input variables and output variables. With its effective learning ability, NN models have been applied successfully in such fields as marketing economics, product design, and consumer perception. As consumers' perception of a product image is often a black box and cannot be precisely described, NN models are suitable for exploring the relationship between the feeling (perception of the product image) of the consumers and the design elements of the product [4]. In addition, it can be used to develop a form design database for simulating and supporting the product design process.

2 Experimental Samples and Analysis

To illustrate how NN models can be used as a useful design support tool, we conduct an experimental study on mobile phones for their popularity as a consumer product and their wide variety of product form elements.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 799 – 804, 2006.

2.1 Experimental Samples of Mobile Phones

In the experimental study, we collected numerical data for building NN models by investigating and categorizing various mobile phones in the market. 33 mobile phones (as shown in Fig. 1) were used as representative experimental samples for training and test the NN models. A group of 15 subjects were involved in the experiment to evaluate the degree to which the 33 mobile phone samples match a set of given product images. Based on our prior study [3], we used three image word pairs for describing the images of mobile phones, including Simple-Complex (S-C), Handsome-Rustic (H-R), and Young-Ripe (Y-R). For example, the 15 subjects were asked to assess the form elements of mobile phones on a simplicity-complexity scale of 1 to 7, where 1 is most simple and 7 is most complex. The result is given in Table 1, which provides a numerical data source for applying the neural network techniques. The detailed processes and methods used are reported in our prior studies [1-3].

Fig. 1. The 33 representative mobile phone samples

2.2 Experimental Analysis of Neural Network Models

NN models are widely used to examine the complex relationship between input variables and output variables [6]. In our experimental study, we use the multilayered feedforward NN models trained with the backpropagation learning algorithm, as it is an effective and popular supervised learning algorithm. We develop four NN models for the training and test process [1-3].

A combined NN model is developed by combining all three images as three output neurons of the model, using the average values of three image word pairs. The other three models are developed using each of three image word pairs (i.e. the S-C value, the H-R value, and the Y-R value) as the output neuron respectively. In our prior study, we apply the four most widely used rules to determine the number of hidden neurons in the

single hidden layer for each model [6]. Each model is associated with the rule used, such as -HN1, -HN2, -HN3, or -HN4.

(The numbers of input neurons + the numbers of output neurons) / 2 (1)

(The numbers of input neurons * the numbers of output neurons) ^ 0.5 (2)

(The numbers of input neurons + the numbers of output neurons) (3)

(The numbers of input neurons + the numbers of output neurons) * 2 (4)

The 28 samples in the training set (given in Table 1) are used to train the four NN models. Each model trained 5,000 epochs at each run. To evaluate the performance of the four NN models in terms of their prediction ability in determining the best design combination of mobile phone form elements, the 5 samples in the test set are used.

Table 1. Consumer perception for 33 selected mobile phones

Phone	X	X	X	X	X	X	X	X	X	S-C	H-R	Y-R
1	2	2	1	3	2	2	1	2	1	4.46	5.03	4.87
2	2	3	1	2	3	2	1	2	3	3.13	3.37	3.31
3	1	3	2	1	2	1	2	3	1	2.86	5.85	5.23
4	3	3	1	2	2	1	3	1	1	5.53	5.75	4.61
5	4	1	1	2	3	2	3	1	2	5.53	4.87	3.94
6*	1	2	1	2	2	1	1	3	1	6.40	3.63	5.08
7	2	2	1	2	2	1	2	2	3	6.86	2.90	3.57
8	4	1	2	3	2	2	1	2	3	6.80	3.26	4.82
9	2	2	1	1	1	1	2	1	2	4.60	4.66	4.40
10	2	3	1	2	2	1	1	1	1	6.20	6.74	6.63
11*	3	2	3	3	2	2	1	1	2	6.60	2.54	2.07
12	2	1	1	2	2	1	2	3	1	4.53	6.68	6.74
13	1	2	1	2	1	1	3	3	1	3.93	5.65	5.85
14	3	2	2	3	1	1	2	3	3	6.93	5.65	5.80
15	2	2	1	3	1	1	1	3	1	5.60	5.23	4.82
16*	1	1	1	3	1	1	1	1	1	2.73	3.68	2.28
17	1	1	1	2	3	1	2	2	3	4.33	5.08	5.03
18	2	2	1	2	1	1	2	3	3	2.53	4.20	3.05
19*	2	3	2	2	1	1	2	1	2	4.53	3.16	1.81
20	2	2	2	2	1	1	1	2	1	3.20	1.86	2.17
21	2	2	2	2	2	1	1	2	1	5.53	2.95	3.83
22	2	2	2	2	3	1	1	1	1	6.86	3.63	3.42
23	2	2	1	2	2	2	1	2	3	4.46	2.95	3.99
24	2	2	1	2	2	1	2	2	1	3.66	4.30	3.57
25	3	3	1	2	1	1	2	2	3	3.06	2.95	1.76
26	3	3	3	3	2	1	2	3	1	4.53	4.20	2.07
27	3	2	1	3	2	2	1	2	1	3.60	3.88	3.57
28*	1	2	2	2	1	2	1	2	3	4.60	1.08	1.71
29	2	2	2	3	1	1	1	1	1	4.93	2.17	1.76
30	1	3	1	3	1	1	1	1	1	5.33	1.71	2.43
31	2	3	3	1	1	1	1	3	3	3.73	3.52	1.55
32	2	3	3	2	1	2	2	3	2	3.53	4.51	3.21
33	2	1	3	2	2	1	2	2	2	3.73	3.42	2.43

Rows 2-4 of Table 2 show the average values of the three image word pairs on the 5 test samples evaluated by the 15 subjects, which are used as a comparison base for the performance evaluation. With the 5 test samples as the input, Table 2 shows the corresponding three image values predicted by using the four NN models.

Table 2. Predicted image values and RMS errors of the four models for the test set

Phone no.			6	11	16	19	28	RMS errors	
Evaluated Image value	S-C value		6.400	6.600	2.733	4.533	4.600		
	H-R value		3.630	2.541	3.681	3.163	1.089		
	Y-R value		5.081	2.074	2.281	1.815	1.711		
The combined NN model	-HN1	S-C	6.793	7.286	3.583	5.537	2.492		*0.1325*
		H-R	4.404	2.345	2.452	3.336	1.153	0.1325	
		Y-R	6.525	2.491	2.637	4.008	2.810		
	-HN2	S-C	6.026	7.329	3.254	5.358	2.140		
		H-R	5.397	3.337	3.505	5.076	1.121	0.1911	
		Y-R	6.996	3.844	3.353	5.638	2.898		
	-HN3	S-C	6.805	7.339	4.131	5.877	2.897		
		H-R	5.553	2.688	3.366	4.807	1.524	0.1871	
		Y-R	7.136	3.090	3.639	5.380	3.586		
	-HN4	S-C	6.930	7.315	3.748	5.592	3.324		
		H-R	5.622	2.199	2.829	3.810	1.905	0.1585	
		Y-R	7.132	2.410	2.950	4.360	4.075		
The S-C NN model	-HN1		6.529	7.158	4.227	5.364	2.184	0.1839	*0.1647*
	-HN2		5.960	6.969	4.107	5.275	1.967	0.1899	
	-HN3		6.437	7.299	4.310	5.262	2.655	0.1647	
	-HN4		6.562	7.313	4.392	5.303	2.592	0.1715	
The H-R NN model	-HN1		5.156	2.424	4.104	4.011	0.909	0.0859	*0.0614*
	-HN2		4.764	2.233	3.922	3.645	1.100	0.0614	
	-HN3		5.194	2.630	3.969	4.042	0.910	0.0868	
	-HN4		5.388	2.628	3.778	4.185	1.270	0.0971	
The Y-R NN model	-HN1		6.684	2.274	3.952	4.696	2.307	0.1940	*0.1765*
	-HN2		6.032	2.624	4.624	4.007	2.081	0.1765	
	-HN3		6.694	2.865	4.032	4.351	3.070	0.1975	
	-HN4		7.237	2.512	3.323	5.183	3.999	0.2455	

As shown in Table 2, the H-R NN model has the lowest RMS error (0.0614). This suggests that the H-R NN model has the highest prediction ability in determining the best design combination of form elements. The results obtained from the NN models can be used to help product designers work out the best combination of the form elements for a particular design concept represented by a set of product images.

3 The Scatter Space of Product Form Design Database to Consumer Perception

The combined NN model can be used for quantitative prediction analysis. We can use the model to input the values of product form elements, and then output the prediction values of the image word pairs. Each possible combination of 9 form elements can be used to build a form design database consisting of 17,496 different combinations of

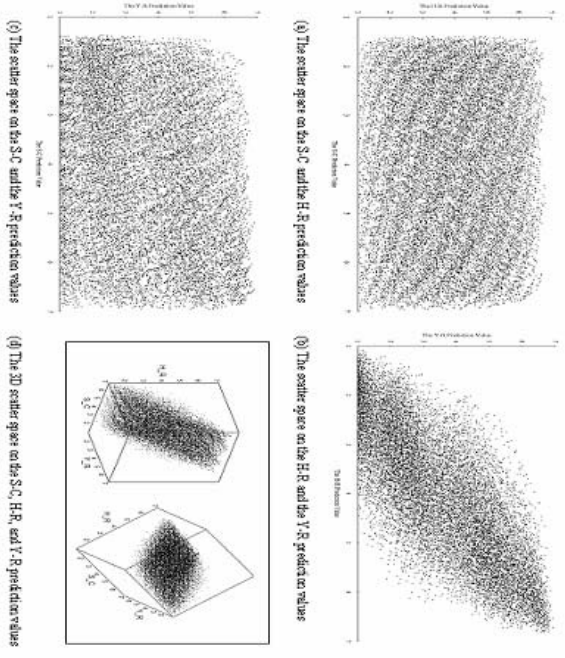
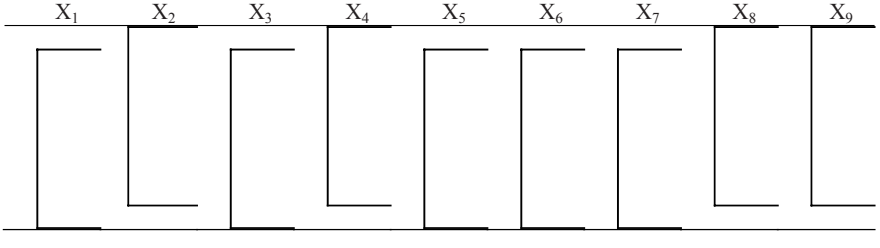


Fig. 2. The scatter space on the S-C, the H-R, and the Y-R prediction values

mobile phone form elements together with their associated image values. The form design database can then be used to determine which form combination(s) will best match the given image values. Fig. 2(a) shows the scatter space of 17,496 different combinations of form elements on the S-C and the H-R prediction values. The prediction values range between 1 and 7. In addition, Fig. 2(a) shows that there is no consistent tendency on the S-C and the H-R prediction values. Fig. 2(b) shows the scatter space on the H-R and the Y-R prediction values. It is noteworthy that the prediction value of the Y-R image increases, while the H-R image value increases. In other words, if a combination of mobile phone form elements trends towards the Handsome image, it also has the Young image tendency. Fig. 2(d) shows the 3D scatter

Table 3. The best combination of product form elements



space on the S-C, the H-R, and the Y-R prediction values. As shown in Fig. 2(d), there is a significant display tendency on the 3D space, like an ellipsoid from some angles. It indicates that the consumer's perception has a certain direction on the three product images. The result can help product designers understand the consumer's perception of form elements for the corresponding product image. As an illustration, Table 3 shows the best form combinations for a set of product images represented by the S-C, H-R, and Y-R values of 4. The combination of form elements given in Table 3 has its S-C value being 4.086, H-R value being 4.066, and Y-R value being 3.943, which is the closest among all 17,496 form combinations.

4 Conclusion

In this paper, we have presented a NN-based consumer oriented approach for transforming product forms into a set of product images perceived by consumers. To illustrate how NN models can be built to help determine the best design combination of form element for matching given product images, we have conducted an experimental study on mobile phones. We have examined four NN models to find out the best model for determining the ideal design of product form. The experimental result has showed that the H-R NN model has the highest prediction ability, but the smallest RMS error of the combined NN model is smaller than the S-C NN model and the Y-R NN model. This result has suggested that the combined NN model can be used to predict the values of the product images of "Simple-Complex", "Handsome-Rustic", and "Young-Ripe" for a given set of form elements. The NN models built can help product designers determine what combination of form elements can best match the desired product images.

References

1. Lai, H.H., Lin, Y.C., Yeh, C.H.: Form Design of Product Image Using Grey Relational Analysis and Neural Network Models. Computers and Operations Research 32 (2005) 2689-2711
2. Lin, Y.C., Lai, H.H., Yeh, C.H.: Consumer Oriented Design of Product Forms. Lecture Notes in Computer Science 3174 (2004) 898-903
3. Lin, Y.C., Lai, H.H., Yeh, C.H.: Neural Network Models for Product Image Design. Lecture Notes in Computer Science 3215 (2004) 618-624
4. Nagamachi, M.: Kansei Engineering: A New Ergonomics Consumer-Oriented Technology for Product Development. International Journal of Industrial Ergonomics 15 (1995) 3-10
5. Negnevitsky, M.: Artificial intelligence. Addison-Wesley, New York (2002)
6. Nelson, M.: A Practical Guide to Neural Nets. Addison-Wesley, New York (1991)

2D Pattern Design of Upper Outer from 3D Specification Based on Neural Networks

Dongyun Wang and Yulin Xu

School of Electrical and Information Engineering,
Zhongyuan University of Technology,
Zhengzhou, Henan, 450007, China
{wdy, xuyulin}@zzti.edu.cn

Abstract. In this article, the 2D pattern design of the upper outer garment from 3D specification by the artificial neural network was proposed. An Elman neural network and back propagation (BP) algorithm were employed. Experiment shows that it is an efficient and appropriate method to solve the problem of 2D pattern from 3D specification. The method presented here can be introduced into integrated garment design and manufacturing system.

1 Introduction

With the increasing of technology development and international competition, today's industrial environment requires further improvement of design quality and efficiency, in the content of the overall production development life cycle. The introduction and adoption of the integrated garment design and manufacturing principle has led to many archetype industrial successes, due to its capability of producing a balanced design and reducing the number of re-designs. In this way design, it can break the barrier between design and manufacturing modules, and incorporate manufacturing considerations into the design phase, thereby generating designs which need fewer re-design and have lower production costs.

Computer integrated approach proposed [1] here is presented in Fig. 1. This block diagram depicts the core elements required for an integrated approach. The key elements which have prevented such an integrated approach from being adopted to date are:

1. an effective design interface for designers to create 3D specifications of garments;
2. a robust 2D pattern flattening module i.e. The creation of 2D patterns through 3D specifications of garments.

For an integrated manufacturing approach to be successful, all two of these difficult elements must be adequately accomplished.

In addition, there is a high degree of automation available in subsequent manufacturing processes which nest 2D patterns and generate cutter head path information for large cutting tables. The lack of development in these areas has resulted in the inability of companies to truly adopt a computer integrated

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 805–814, 2006.

approach to design and manufacture. Our paper aims to describe an approach to obtain the 2D patterns from a given 3D specifications of garments. The procedure is also called auto pattern making. The function of auto pattern making is: when the finished specifications of certain apparel have been selected, let's take trousers as an example, input the sizes and specifications of the trousers, such as length, waist, hip, stand crotch, trouser legs, etc. then, the pattern generation system will make the pattern of the trousers automatically.After select an upper outer garment, input the finished specifications and sizes of the controlled parts, such as breast (B), length (L), shoulder (S), sleeve length (SL), etc.,auto pattern making system can also produce the pattern of an upper outer garment. This can be called making the garment according to your own sizes. Indeed this procedure is from the garment style (3D) to the apparel pattern (2D).

The 2D pattern of an upper.

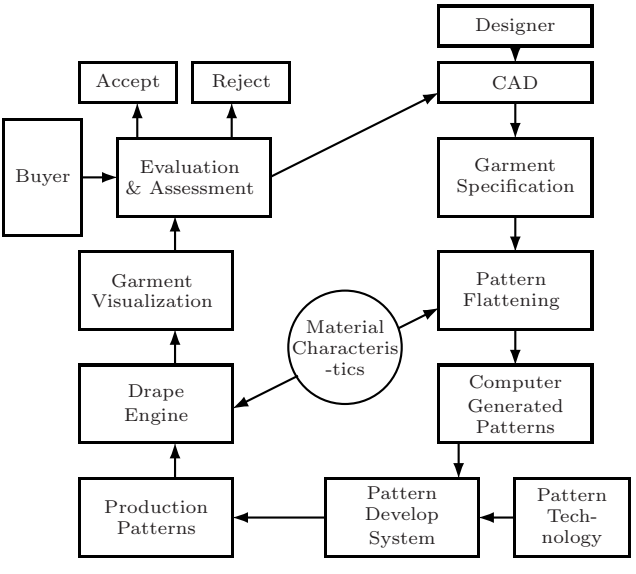


Fig. 1. Proposed garment design cycle

2 Relative Works

The scientists [2] supposed that the intelligent pattern making system could be set up based on the expert intelligence. Using this kind of system, designer can input the specifications and sizes of different key parts (controlled parts), such as breast (B), length (L), shoulder (S), sleeve length (SL), etc., into the computer. After reference mechanism, system can output the different detailed sizes of the front breast width (BWF), the back width in the back (BWB), the front neck width (NWF), the back neck width (NWB), etc. this will offer comprehensive reference data for the pattern design and realize the function combination of the

style design system and the pat-tern design system. Through this way , a certain apparel style has changed into 2D pattern automatically.

When the designer design the pattern, he will first design the finished sizes of the product on the basis of the apparel's style. The finished sizes are the specifications and sizes of the controlled parts. those are key for the style and fashion of the apparel. Then the detailed sizes are designed based on the finished sizes of the product. The design should comply with apparel style, human body structure and his movement. The linear programming methods [3] were used this design. It was supposed that the relation between finished sizes and detailed sizes is linear. It can be expressed as: $y = f(x)$. Here, x is the known data vector (the finished sizes), y is the unknown data vector (the detailed sizes), f is mapping function. Actually, the results of such design is not ideal, and it is difficult for the new learners to master.

The neural network is an ideal nonlinear predicate model, the problems are handled and resolved by adapting environment's influence of various factors gradually and by the mean of its structure's variability, it needn't established the precise math model of the problem, and by discovering the internal causality among the input and output of system which the causality displays an imprecise input/output value, not a precise math analytical formula.

In this article, the pattern design method of the upper outer garment was studied by the technology of artificial neural network. An Elman neural network and back propagation (BP) algorithm were employed. Experiment shows that it is an efficient and appropriate method to solve the problem of 2D pattern from 3D specification. The method present here can be introduced into integrated garment design and manufacturing in garment industry.

3 2D Pattern Design of Upper Outer from 3D Specification Based on Neural Network

3.1 3D Specification and Experiment Data

The sample data that is measured in the test, is the used pattern of the upper outer garment in the garment enterprises. 30 sets of patterns of the upper outer garment that have different sizes are tested. The 3D specification of detailed structure is shown in Fig 2. The procedures of experiment are as following:

1. Take the specifications and sizes of the finished upper outer garment as the in-put data, take the specifications and sizes of the detailed parts as the output data, study their nonlinear mapping relation;
2. Test 30 specifications and sizes of the detailed parts and the finished product in the upper outer garment's pattern of the actual production;
3. Take the 30 specifications and sizes of the testing finished garment as the in-put data, take the specifications and sizes of detailed parts as the output data, train the neural network each pair of data, to obtain the weights of the neural network;

4. Experimental verification; Take the data (that is gained in the structure design of the pattern making of the garment) as the standard, compare it with the out-put data of the neural network, analyze the error.

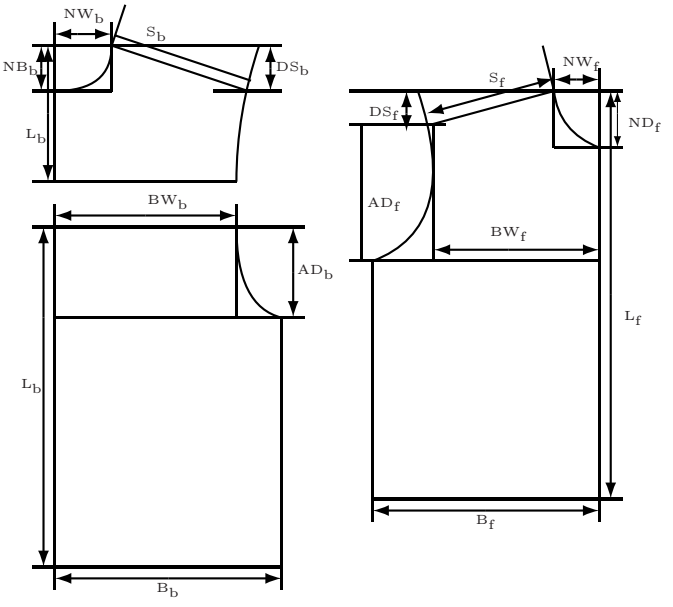


Fig. 2. The structure of the upper outer garment and the specification of detailed parts

3.2 The Structure of the Neural Network and the Error Back Propagation Algorithm for Weights of Network

The static state multi-layer forward network has disadvantage factors, such as the low learning speed and sensitive to the outer noises, etc. here, the Elman static state network is adapted for the complicated apparel model. The structure of the Elman network is as shown in Fig. 3. Beside the units of the input layer, the hidden layer, and the output layer, there is a special structure unit in Elman network. The hidden layer is outputted by postpone and storage the structure unit, then, connected it to the input of the hidden layer automatically. The way of auto connection makes the network has more sensitivity to the data of the past state. It is advantage to set up model in the system.

Suppose the input value of the network is $u(k-1)$, $x_c(k)$ is the state vector of the hidden neurons, the output value is $y(k)$, then, the nonlinear formula is tenable:

$$x(k) = f(W^1 x_c(k) + W^2 u(k-1)) \ .$$

$$x_c(k) = x(k-1) \ .$$

$$y(k) = g(W^3 x(k)) \ .$$

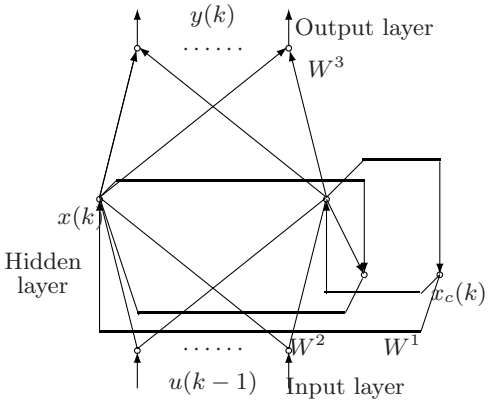


Fig. 3. The structure of Elman neural networks

Among them, W^1, W^2, W^3 are respectively the connecting weights matrix: from the structure unit to the hidden layer, from the input layer to the hidden layer, and from the hidden layer to the output layer. $f(\cdot)$ and $g(\cdot)$ is respectively the active function of the output unit and the unit of the hidden layer, here, the S-type active function and the linear function are adapted. We can know from above:

$$x_c(k) = x(k-1) = f(W^1_{k-1}x_c(k) + W^2_{k-1}u(k-1)) \ .$$

And because $x_c(k-1) = x(k-2)$, the above formula can be expanded again, that is, $x(k)$ is gained by continually iterating the weights W_1^{k-1}, W_2^{k-2}. The error back propagation algorithm for weights in the network training is as following, the supposed target error function is:

$$E = \sum_p E_p \ .$$

Where $E_p = (1/2)(y_d(k) - y(k))^{\mathrm{T}}(y_d(k) - y(k))$, $y_d(k)$ is sample output of the neural network.

For the connecting weights W^3 from the hidden layer to the output layer,

$$\frac{\partial E_p}{\partial w^3_{ij}} = -(y_{d,i}(k) - y_i(k)) \cdot \frac{\partial y_i(k)}{\partial w^3_{ij}} = -(y_{d,i}(k) - y_i(k))g_i^{'}(\cdot)x_j(k) \ .$$

Let $\delta^0_i = (y_{d,i}(k) - y_i(k))g_i^{'}(\cdot)$, then

$$\frac{\partial E_p}{\partial w^3_{ij}} = -\delta^0_i x_j(k), \ i = 1, 2, \cdots, m; j = 1, 2, \cdots, n \ .$$

For the connecting weights W^2 from the input layer to the hidden layer,

$$\frac{\partial E_p}{\partial w^2_{jq}} = \frac{\partial E_p}{\partial x(k)}\frac{\partial x(k)}{\partial w^2_{jq}} = \sum_{i=1}^{m}(-\delta^0_i w^3_{ij}) \cdot f_j^{'}(\cdot)u_q(k-1) \ .$$

Similarly, denote $\delta_j^h = \sum_{i=1}^{m}(-\delta_i^0 w_{ij}^3)\cdot f_j^{'}(\cdot)$, then

$$\frac{\partial E_p}{\partial w_{jq}^2} = -\delta_j^h u_q(k-1),\ j=1,2,\cdots,n; q=1,2,\cdots,r\ .$$

We can get W^1, the weights for structure unit to hidden layer,

$$\frac{\partial E_p}{\partial w_{jl}^1} = \sum_{i=1}^{m}(-\delta_i^0 w_{ij}^3)\cdot\frac{\partial x_j(k)}{\partial w_{jl}^1},\ j=1,2,\cdots,n; l=1,2,\cdots,n\ .$$

and

$$\begin{aligned}\frac{\partial x_j(k)}{\partial w_{jl}^1} &= \frac{\partial}{\partial w_{jl}^1} f_j\left(\sum_{i=1}^{n} w_{ji}^1 x_{c,i}(k) + \sum_{i=1}^{r} w_{ji}^2 u_i(k-1)\right)\\ &= f_j^{'}(\cdot)\left(x_{c,l}(k) + \sum_{i=1}^{n} w_{ji}^1 \frac{\partial x_{c,i}(k)}{\partial w_{jl}^1}\right)\\ &= f_j^{'}(\cdot)\left(x_l(k-1) + \sum_{i=1}^{n} w_{ji}^1 \frac{\partial x_i(k-1)}{\partial w_{jl}^1}\right)\ .\end{aligned}$$

Because $\Delta w_{ij} = -\eta(\partial E_p/\partial w_{ij})$, so the error back propagation algorithm for weights Elman network is as following

$$\begin{aligned}\Delta w_{ij}^3 &= \eta\delta_i^0 x_j^k,\ i=1,2,\cdots,m; j=1,2,\cdots,n\ .\\ \Delta w_{jq}^2 &= \eta\delta_j^h u_q^{k-1},\ j=1,2,\cdots,n; q=1,2,\cdots,r\ .\\ \Delta w_{jl}^1 &= \eta\sum_{i=1}^{m}\delta_i^0 w_{ij}^3\frac{\partial x_j(k)}{\partial w_{jl}^1},\ j=1,2,\cdots,n; l=1,2,\cdots,n\ .\\ \frac{\partial x_j(k)}{\partial w_{jl}^1} &= f_j^{'}(\cdot)\left(x_l(k-1) + \sum_{i=1}^{n} w_{ji}^1\frac{\partial x_i(k-1)}{\partial w_{jl}^1}\right)\ .\end{aligned}$$

Here:

$$\begin{aligned}\delta_i^0 &= (y_{d,i}(k) - y_i(k))g_i^{'}(\cdot)\ .\\ \delta_j^h &= \sum_{i=1}^{m}(-\delta_i^0 w_{ij}^3)\cdot f_j^{'}(\cdot)\ .\end{aligned}$$

4 Experiment and Result Analysis

Based on the algorithm mentioned above, we train the neural network by using nntool in the controlling toolbox of the neural network in the Matlab6.1. We select 30 random groups of effective data from the existing data as the learning sample to train the network. After 24449 steps of training, the network is converged. The error square summation E reaches 0.02, and get the network weights matrices W^1, W^2, W^3 shown in Appendix.

Table 1 and Table 2 (see Appendix) are 10 groups compare between the data of testing sample with the output value of the network. Sample data can be acquired from the actual one. It shows that the output value of the network is basically accord with the actual value. Furthermore, the result is more comply with the demands of the fashion design according to the experiences.

5 Conclusions

1. Study the design method of the apparel' structure by using the artificial nerve network, it has high sensitivity and has good application value.
2. The research can combine the function of the fashion design system with the pattern making system, to produce the 2D pattern of the fashion model automatically.
3. It is a new attempt to study the apparel science and the information dealing technology science together, it will bring new vitality to the research of the apparel production technology.

Acknowledgement

This work is supported by Natural Science Foundation of Henan Province, China.

References

1. Collier, B.J., Collier, J.: CAD/CAM in the Textile and Apparel Industry. Clothing Textile 8(3) (1990) 7-12
2. Dekker, L., Douros, I., Buxton, B., Treleaven, P.: Building Symbolic Information for 3D Human Body Modeling from Range Data. 3-D Digital Imaging and Modeling Proceedings, (1999) 388-397
3. Wu, J.: Suit-Dress Structure Design and Its Applications. Textile Industry Press, Beijing (2000)
4. Wang, C.C.L., Smith, S.S.F., Yuen, M.M.F.: Surface Flattening Based on Energy Model. Computer Aided Design 36(11) (2002) 823-833
5. Levin, A.: Combined Subdivision Schemes for the Design of Surfaces Satisfying Boundary Conditions. Computer Aided Geometry 16(5) (1999) 345-354
6. Hinds, B.K., McCartney, J., Woods, G.: Pattern Development for 3D Surfaces. Computer Aided Design 23(8) (1991) 583-592
7. McCartney, J., Hinds, B.K.: The Flattening of Triangulated Surfaces Incorporating Darts and Gussets. Computer Aided Design 31 (1999) 249-269

Appendix

$$W^1 = \begin{pmatrix} 0.153 & 0.3163 & 0.717 & -0.397 & -0.292 & 0.065 & 0.005 & 0.352 \\ -0.499 & -0.602 & -0.521 & -0.217 & 0.16 & 0.653 & 0.328 & -0.322 \\ 0.683 & -0.156 & 0.767 & -0.647 & -0.674 & -0.328 & -0.402 & -0.494 \\ 0.888 & -0.242 & 0.524 & -0.018 & 0.236 & -0.191 & -0.755 & 0.406 \\ 0.174 & -0.607 & -0.101 & -0.165 & -0.462 & -0.349 & -0.103 & 0.516 \\ -0.864 & 0.611 & 0.221 & -0.067 & 0.160 & -0.162 & -0.061 & 0.891 \\ 0.516 & 0.560 & 0.747 & 0.184 & 0.010 & -0.354 & -0.803 & -0.044 \\ 0.192 & -0.085 & 0.245 & -0.749 & -0.062 & -0.185 & 0.287 & 0.704 \end{pmatrix}$$

$$W^2 = \begin{pmatrix} -0.166 & 2.197 & 1.6507 & -0.145 & -1.363 \\ 0.508 & -1.618 & 0.279 & 0.792 & 0.327 \\ 2.605 & -2.249 & 2.002 & -1.114 & 0.372 \\ 0.269 & -0.710 & -0.413 & -0.681 & -0.758 \\ -2.908 & 3.369 & 1.717 & -1.336 & 4.564 \\ 0.559 & -2.250 & -0.156 & 1.156 & -0.018 \\ -0.349 & 0.302 & 1.408 & -0.151 & -0.565 \\ 0.328 & -0.628 & -0.367 & -0.287 & -0.780 \end{pmatrix}$$

$$W^3 = \begin{pmatrix} 1.002 & -0.609 & -0.232 & -0.159 & 0.437 & 1.161 & -0.021 & -0.460 \\ 0.488 & -0.992 & -0.011 & -0.143 & 0.518 & -0.816 & 0.625 & 0.761 \\ 1.004 & -0.452 & -0.148 & -0.195 & 0.613 & 0.307 & 0.272 & 0.149 \\ 0.206 & -0.875 & -0.024 & -0.451 & 0.061 & -0.160 & 0.004 & 0.029 \\ 0.388 & -0.619 & -0.892 & -0.042 & -0.211 & -0.355 & -0.485 & -0.871 \\ 0.178 & -0.262 & -2.264 & 1.504 & -0.278 & 1.076 & -0.108 & -1.766 \\ 0.175 & -1.006 & 0.186 & -0.349 & 0.005 & -0.454 & 0.578 & -0.001 \\ 0.642 & 0.131 & -0.049 & -0.288 & 0.192 & 0.125 & 0.164 & -0.357 \\ 0.653 & 0.325 & 0.084 & -0.193 & 0.105 & 0.051 & 0.487 & -0.312 \\ 0.321 & -0.505 & 0.212 & -0.133 & -0.061 & -0.356 & 0.600 & -0.166 \\ 0.173 & -0.615 & -0.179 & -0.526 & -0.200 & -0.086 & 0.081 & -0.867 \\ -0.459 & 0.253 & 0.334 & -0.447 & -0.454 & 1.139 & -0.394 & -0.965 \\ -0.621 & -0.183 & 0.397 & -0.299 & -0.479 & 1.115 & -0.795 & -1.005 \\ -0.664 & 0.305 & 0.460 & -0.324 & -0.505 & 0.646 & -0.769 & -0.904 \\ 0.065 & 0.146 & 0.238 & -0.444 & -0.120 & -0.711 & -0.055 & -0.272 \\ 0.086 & 0.169 & 0.073 & -0.017 & -0.082 & -1.148 & -0.815 & -0.314 \\ -0.272 & -0.573 & 0.406 & 1.157 & -0.198 & -0.970 & -2.052 & -0.299 \\ 0.317 & 0.115 & -0.015 & -0.067 & 0.0497 & -0.438 & 0.942 & 0.101 \\ -0.058 & 0.0064 & 0.166 & -0.397 & -0.12 & -0.400 & 0.622 & -0.223 \\ 0.148 & 0.059 & 0.262 & -0.196 & -0.178 & -0.193 & 0.771 & -0.333 \\ 0.213 & 0.327 & 0.323 & -0.286 & -0.078 & -0.172 & 1.069 & -0.013 \\ 0.124 & -0.684 & -0.207 & -0.598 & -0.193 & 0.0717 & 0.252 & -0.876 \\ 0.005 & -0.214 & 0.591 & 0.232 & -0.571 & -0.442 & -0.914 & -0.979 \\ -0.177 & 0.436 & 0.571 & 0.0513 & 0.0197 & -1.451 & -0.154 & 0.305 \\ 0.065 & -0.035 & 0.348 & -0.120 & -0.197 & -0.794 & 0.067 & -0.319 \\ 0.008 & -0.677 & 0.482 & -0.234 & -0.225 & -0.605 & 0.146 & -0.396 \end{pmatrix}$$

Table 1. Compare result

unit:cm

	1		2		3		4		5	
	sample	result	sample	result	sample	result	sample	result	sample	result
NW_f	6.5	6.3973	6.4	6.5923	4.3	4.4018	6.1	6.0494	4.2	4.3484
ND_f	9.2	9.1677	9.8	9.1782	5.5	5.9976	9.6	9.5702	5.3	5.6682
NL_f	11.1	11.484	11.7	11.7	7.7	7.9717	11.6	11.561	7.3	7.7496
S_f	18.1	17.937	19.8	19.481	18.2	16.437	17.8	16.939	17	16.299
DS_f	5.5	5.5739	.5	5.5832	5.2	5.1244	4.4	4.5351	5	4.8782
SS_f	18	18.052	16.5	17.169	170	16.907	320	15.956	17	16.971
AD_f	22.6	22.594	23.6	23.337	22.4	22.476	22.7	22.559	21.8	21.734
BW_f	22.8	22.735	23.6	23.61	20.8	20.997	22.2	22.137	20.2	20.329
B_f	30.9	30.867	32.2	32.214	29.8	29.909	30.6	30.597	28.9	28.906
AA_f	26.6	26.625	28	27.918	27.3	27.457	26.6	26.731	26.7	26.4
L_f	66.9	67.181	70.7	70.461	68.4	68.139	61.9	61.856	65.2	65.038
NW_b	8.6	8.6206	8.8	8.8361	10.0	9.8392	8.6	8.593	10	9.8609
ND_b	6	6.004	6.1	6.1304	6.9	6.763	6	6.0029	6.8	6.7966
NL_b	12.5	12.415	12.6	12.69	14.3	14.17	12.5	12.446	14.3	14.207
S_b	17.9	17.82	19.1	19.035	18.3	18.28	17.5	17.434	7.4	17.411
DS_b	5.2	5.1812	5.4	5.4011	5.0	5.0309	10	5.0016	4.8	4.8512
SS_b	16.8	16.694	17	16.764	170	17.001	17	16.947	17	17.094
AD_b	14.3	14.427	15.2	14.886	13.7	14.017	14.3	14.349	13.5	13.449
BW_b	24.5	24.504	25.3	25.237	25.0	25.104	24.2	24.24	24.5	24.448
B_b	31.7	31.755	32.8	32.896	33.7	33.748	31.8	31.718	33	32.836
AA_b	17.3	17.155	17.4	17.984	18.2	18.498	17.4	17.636	18.1	17.84
L_b	53.4	53.935	57	56.986	55.2	55.021	48.9	49.145	52.5	52.22
ST	9.7	9.6477	9.9	9.945	10.9	10.801	9.7	9.6629	21.5	10.671
SL △	103	51.4	54	54	55	55	50.5	50.5	53.5	53.5
SC	39.6	39.889	40.5	39.98	38.0	38.352	40.8	40.441	37.1	37.497
BC	57.5	57.359	60.1	60.056	60.0	60.136	57.4	57.283	116.8	58.098
STA_f	30.3	30.217	31.7	31.657	32.0	32.152	30.5	30.393	31.3	31.164
L△	71.5	71.5	75	75	73.0	73	67	67	70	70
B△	125	125	130	130	128.0	128	125	125	124	124
S△	48.5	48.5	50.5	50.5	50.0	50	45.5	45.5	48.5	48.5
N△	47	47	47.5	47.5	44	44	47.5	47.5	44	44

Table 2. Compare result

unit:cm

	6		7		8		9		10	
	sample	result	sample	result	sample	result	sample	result	sample	result
NW_f	6.5	6.4474	4.1	4.149	6.3	6.3464	4.5	4.6153	8.1	8.1479
ND_f	9.4	9.6008	5.2	5.1214	9.7	9.6462	5.8	5.7586	6.9	6.9344
NL_f	11.8	11.857	7.2	7.267	11.7	11.81	7.9	8.0013	11.8	11.838
S_f	20.6	20.815	17	16.669	20.5	20.459	19.0	17.638	17.9	18.073
DS_f	5.7	5.55	4.9	4.7799	5.6	5.529	5.2	5.4229	5.1	5.0476
SS_f	16	16.127	17	17.096	160	15.884	170	16.931	170	16.558
AD_f	24.9	24.707	21.3	21.118	24.5	24.489	23.0	23.126	21.8	21.846
BW_f	24.7	24.493	19.8	19.69	24.3	24.299	21.7	21.819	24.0	24.148
B_f	34.4	33.958	28.1	27.915	33.5	33.178	30.8	31.145	32.3	32.514
AA_f	30.1	29.937	25.9	25.504	29.4	29.032	28.2	28.55	26.2	26.293
L_f	74.8	74.196	63.1	63.152	70.0	72.746	72.5	72.512	70.3	70.044
NW_b	8.4	8.973	9.8	9.953	8.8	8.8249	10.2	10.11	10.2	10.058
ND_b	6.1	6.1561	6.8	6.8727	6.2	6.1025	6.9	6.8926	6.9	6.8544
NL_b	12.8	12.871	14.3	14.327	12.8	12.786	14.6	14.48	14.3	14.189
S_b	20.4	20.449	16.8	16.837	20.3	20.276	19.3	19.285	18.3	18.422
DS_b	5.6	5.5943	4.7	4.7436	5.7	5.6984	5.2	5.1906	5.2	5.1108
SS_b	16.5	16.558	17	17.2	170	16.883	170	16.812	170	16.668
AD_b	15.8	15.839	13.1	12.957	15.0	15.452	14.4	14.437	13.4	13.196
BW_b	26.3	26.45	24	23.956	26.0	26.059	26.0	25.874	24.6	24.563
B_b	34.9	34.842	32.2	32.096	34.0	33.835	35.1	34.943	32.1	32.294
AA_b	19.7	19.627	17.4	17.203	18.6	18.622	19.1	19.286	17.1	17.304
L_b	60.5	60.402	50.6	50.532	56.9	58.735	59.0	59.078	57.0	56.913
ST	10.5	10.294	10.5	10.607	10.4	10.221	11.2	11.075	10.2	10.284
SL △	57	57	52.5	52.5	55.2	56	58.0	58	59.3	59
SC	41.7	41.982	36.5	36.539	43.2	42.221	39.4	38.614	37.4	37.169
BC	64.2	63.872	65.5	56.533	64.0	62.826	62.4	62.372	57.4	57.623
STA_f	67.6	33.693	30.5	30.396	33.8	33.051	33.4	33.34	30.9	30.961
L △	79	79	68	68	76.0	77	77.0	77	75.0	75
B △	138	138	120	120	134.0	134	132.0	132	128.0	128
S △	52.5	52.5	47.5	47.5	50.0	50.5	52.0	52	51.0	51
N △	46	46	44	44	49	49	45	45	52	52

Design of a Broadband Microwave Amplifier Using Neural Performance Data Sheets and Very Fast Simulated Reannealing

Yavuz Cengiz[1], Hüseyin Göksu[1], and Filiz Güneş[2]

[1] Süleyman Demirel University, Isparta Turkey
ycengiz@mmf.sdu.edu.tr, goksu@sdu.edu.tr
[2] Yıldız Technical University, Istanbul,Turkey
gunes@yildiz.edu.tr

Abstract. In this work, the neural performance data sheets of the transistor are employed to determine the feasible design target space in the optimization of a microwave amplifier. The basic amplifier with a single transistor between the input and output matching circuits (IMC and OMC) is also worked out. Very Fast Simulated Reannealing (VFSR) is utilized in the multi - objective optimization process for the global minimum of the objective function which is expressed as a function only gain of a matching circuit, in the negative exponential form to ensure the rapid convergence.

1 Introduction

Optimization is one of the fundamental processes frequently encountered in the engineering problems and is highly nonlinear in terms of the descriptive parameters of the system.

For optimization of a microwave amplifier, design variables are generally the matching circuit parameters whose lower and upper limits are very often determined by the technology to be utilized in the realization stage of the design. Nevertheless, design targets are still one of the main problems of the amplifier optimization. Generally, the optimization is focused on the transducer power gain (G_T) over a frequency band of operation without controlling the other performance criteria such as the noise (F), the input standing wave ratio (Vi). Certainly, within the optimization process one can easily embed the desired performance goals without knowing the physical limits and / or compromise relations among F, Vi and G_T appropriately. Unfortunately this process often fails to attain the desired goals. However, the Neural Performance Data Sheets (NPDS) of the transistor overcomes all the above - mentioned handicaps and embeds the compatible (F, V_i , G_T) triplets with their source (Z_S) and load (Z_L) terminations together over a predetermined frequency band of operation [1].

2 Neural Performance Data Sheets for a Microwave Transistor

Neural performance data sheets for a microwave transistor can be obtained from the neural block diagramme given in the Fig.1. According to this block diagramme, this work can be described in three main stages.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 815 – 820, 2006.

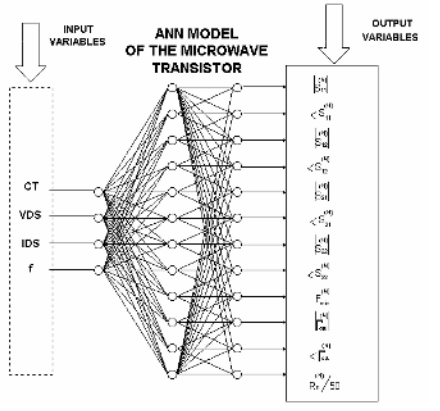


Fig. 1. Neural Network Model for a Microwave Transistor [6]

In the first stage the signal and noise behaviours of the small – signal transistor are modeled by a Multiple Bias and Configuration, Signal – Noise neural network [2,3]. The second stage consists of determining of the compatible performance (F_{req}, V_{ireq}, G_{Treq}) triplets and their associated source (Z_{sreq}) and load (Z_{Lreq}) terminations [4,5]. The third stage of the work is to obtain the neural performance data sheets using the interrelations among the performance measure components F, V_i, G_T and the operation parameters CT, V_{DS}, I_{DS}, f of the transistor.

Two approaches can be followed in the utilization of the (F, V_i, G_T) triplet and the Z_S, Z_L functions in the design of the microwave amplifiers circuits :

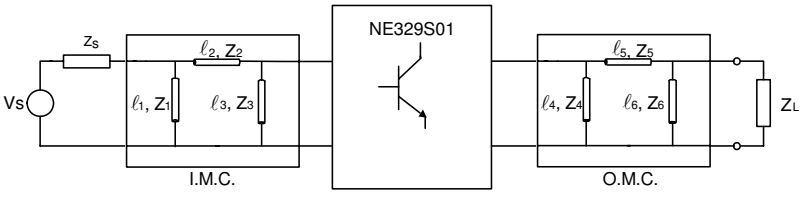


Fig. 2. The Single Transistor Amplifier with the Π -T Type Matching Circuits

i. Only the (F, V_i, G_T) triplet function can be employed to provide the design targets over the predetermined bandwidth to the optimization process of the parameters of the front- and back-end matching networks that was recently applied and completed by Aliyev [6].
ii. In the second approach only the Z_S, Z_L termination functions can be employed in the design of the front- and back-end matching networks, respectively, to obtain corresponding performance (F, V_i, G_T) triplet over the predetermined bandwidth.

Block Diagramme of the microwave amplifier is given in the Fig.2, where the input matching (IMC) and output matching circuits (OMC) are lossless and reciprocal, which are designed by the elements of distributed or lumped parameters in the various configurations.

3 Compatible Performance (F, V_i, G_t) Triplets and the (Z_s, Z_L) Terminations

Compatible Performance (F, V_i, G_T) Triplets and the (Z_S, Z_L) Terminations are the physically realizable simultaneous solutions of the following three nonlinear performance measure equations in the operation domain of the device:

$$F = \frac{(S/N)_i}{(S/N)_o} = F\{R_s, X_s\} = F_{\min} + \frac{R_N}{|Z_{opt}|^2} \frac{|Z_s - Z_{opt}|^2}{R_s} \tag{1}$$

$$VSWR = V_i\{R_s, X_s, R_L, X_L\} = \frac{1+|\rho_i|^2}{1-|\rho_i|^2}, \rho_i = \frac{Z_s - Z_i^*}{Z_s + Z_i} \tag{2}$$

$$G_T\{R_s, X_s, R_L, X_L\} = \frac{4R_s R_L |Z_{21}|^2}{|(z_{11}+Z_s)(z_{22}+Z_L) - z_{12}z_{21}|^2} \tag{3}$$

Where $[Z] = [Z_{11}\ Z_{12}\ Z_{21}\ Z_{22}]^t$, $[N] = [F_{\min}\ |Z_{opt}|\ \psi_{opt}\ R_n]^t$ are respectively signal and noise parameters of the transistor, two-part with the given operation parameters (Fig 2.); Z_S, Z_L are respectively the passive source and load terminations ensuring stable operation of the transistor $R_e\{Z_i\} > 0$ and $R_e\{Z_0\} > 0$ Where Z_i and Z_0 are respectively the input and output impedances of the transistor two-part.

4 Microwave Amplifiers and Its Optimization

In our work, two different types of elements are used in the matching circuits:

(i) Unit Element (UE) which is the transmission line segment with the electrical length $\theta = \beta\lambda$ and the characteristics impedance Z_O;

(ii) Reactive Element, which may be either capacitor or inductor. One type of matching circuits are employed by cascade connecting of these elements in –T- configurations.

In our circuitry, the optimization variables are the characteristic impedances Z_{oi} and the physical lengths l_i, i = 1…6 of the transmission line segments used in the matching circuits.

$$\vartheta_C = [l_1\ l_2\ l_3\ l_4\ l_5\ l_6\ Z_{o1}\ Z_{o2}\ Z_{o3}\ Z_{o4}\ Z_{o5}\ Z_{o6}]^t$$

i. Type A vector for The IMC Terminated by the Conjugated Source Termination;

$$\vartheta_A = [l_1\ l_2\ l_3\ Z_1\ Z_2\ Z_3\ ]^t \tag{4}$$

ii. Type B vector for The OMC Driven by the Conjugated Load Termination;

$$\vartheta_A = [l_1\ l_2\ l_3\ Z_1\ Z_2\ Z_3\ ]^t$$

The function to be optimized are the transducer gain G_T, the input VSWR V_i and noise figure F, all of which can be expressed in terms of Transfer Scattering

parameters of the two-parts in the Fig. 2. G_T function of the amplifier can be reexpressed as

$$G_T = \frac{\left(1-|\Gamma_S|^2\right)\cdot|s_{21}|^2\cdot\left(1-|\Gamma_L|^2\right)}{|1-\Gamma_{in}\Gamma_S|^2\cdot|1-\Gamma_{out}\Gamma_L|^2} \tag{5.1}$$

$$\Gamma_{in} = s_{11}+\frac{s_{21}\Gamma_L s_{12}}{1-s_{22}\Gamma_L} \qquad \Gamma_{out} = s_{22}+\frac{s_{21}\Gamma_s s_{12}}{1-s_{11}\Gamma_s} \tag{5.2}$$

Where Γ_{in} and Γ_{out} are the input and output reflection coefficients respectively. A, B, C, D belongs to the total two-part, Z_S^1 and Z_L^1 are the given real source and load impedances. Similarly Z_S and Z_L are respectively the output impedance (Z_{out1}) of IMC and the input impedance (Z_{in2}) of OMC. Z_{out1} and Z_{in2} can be given in terms of the related ABCD parameters as follows:

$$Z_{S1} = Z_{out1} = \frac{B_1+D_1Z_{S1}}{A_1+C_1Z_{S1}} \qquad Z_{L1} = Z_{in2} = \frac{A_{21}Z_{L2}+B_2}{C_2Z_{L2}+D_2} \tag{6}$$

These objective functions use the termination functions $Z_L(w)$, $Z_S(w)$ providing the compatible (F, V_i, G_T) over the feasible frequency band of operation:

$$E_{out} = \sum_{i=1}^{N}\left|Z_L^c(\omega_i)-Z_{Lref}^d(\omega_i)\right|^2 \qquad E_{in} = \sum_{i=1}^{N}\left|Z_s^c(\omega_i)-Z_{sref}^d(\omega_i)\right|^2 \tag{7}$$

Here objective functions numbered by (7) are respectively for the output and input matching circuits and superscripts “c” and “d” indicating the calculated and desired impedances, respectively.

5 Very Fast Simulted Reannealing

Very Fast Simulated Reannealing (VFSR) is one of the better improved versions of the Simulated Annealing (SA) algorithm which is originated from the statistical mechanical model of atoms of metals being first heated and then cooled down slowly for a globally stable crystal form. An analogy can be formed between the evolution of the energy of each atom and the error function of a general optimization problem simulating the formation of the globally stable metal crystal during the SA process.

Simulated Annealing, before reaching the VFSR form, has gone through the stages of Boltzmann Annealing (BA) which uses a Boltzmann distribution to guide the heuristics and Fast Simulated Annealing (FSA) which uses a Cauchy Annealing.

In Very Fast Simulated Reannealing, transition rules are guided by the parameter y_i where a random number u_i is drawn from a uniform distribution U[0,1] which is then mapped as: $y_i = \mathrm{sgn}(u_i-1/2)T_i\left[\left(1+1/T_i\right)^{|2u_i-1|}-1\right]$ VFSR runs on an annealing schedule decreasing exponentially in time k, $T = T_0e^{-ck^{1/D}}$ This annealing schedule

makes the algorithm faster than fast Cauchy annealing (FSA), where $T = T_0 / k$ and much faster than Boltzmann annealing, where $T = T_0 / Ink$.

The introduction of re-annealing also permits adaptation to sensitivities of the parameters [7]. Although early versions of SA such as BA lacked from the curse of dimensionality, VFSR is a very fast tool which was shown to be much efficient than Genetic Algorithms (GA) for six nonconvex functions [8]. It has also been useful in successful solution of complex engineering problems [9], [10].

6 Computed Results and Discussion

The aim of the optimization is to obtain a flat gain response subject to the constraint functions which are $V_{ireq}(w)$, $F_{req}(w)$ in our case, in the frequency band of interest. It should be pointed out that stability is not included to the optimization as a constraint since all gains values G_T between $G_{T\,max}$ and $G_{T\,min}$ take place in the "Unconditionally Stable Working Area" (USWA).

NE329S01 is biased at I_C = 5mA and V_{CE}=10V, for which the potential performance characteristics are given $F_{req}(w_i)$=0,46dB, $V_{ireq}(w_i)$=1,0, $G_{Treq}(w_i)$ =12dB, i= 3,…11 are supplied into the optimization process as the reference values over the operation bandwidth. These reference values give the physically realizable $Z_S(w_i)=R_S(w_i)+jX_S(w_i)$; $Z_L(w_i)=R_L(w_i)+jX_L(w_i)$, i = 3,…11 termination solutions to the simultaneous nonlinear equations of F (R_S, X_S) = F_{req}, V_i (R_S,X_S,R_L,X_L)=V_{ireq}, G_T= (R_S,X_S,R_L,X_L) = G_{Treq}. Since the optimization process also find out the approximately

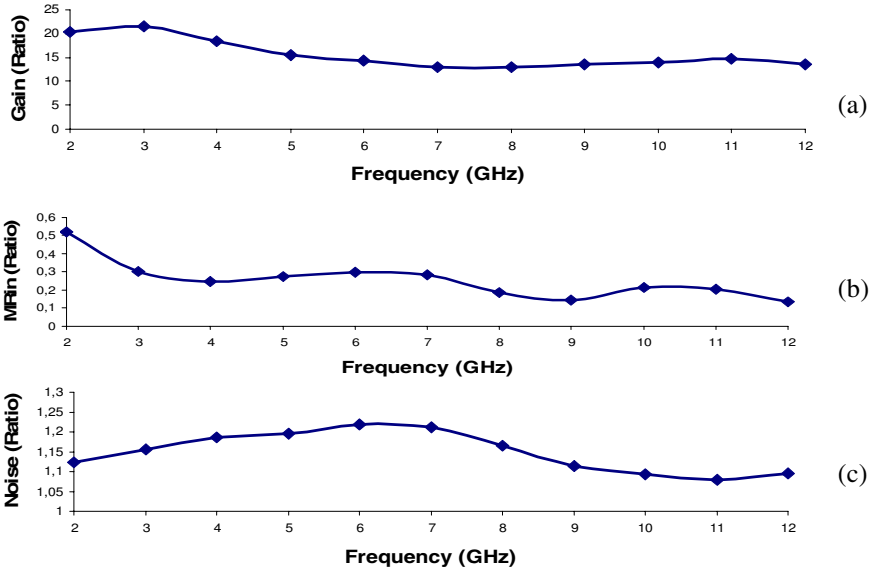


Fig. 3. (a) G_T (ratio) (b) Input Reflection Coefficient (c) Noise (ratio) Variation of the Microwave Amplifier G_{Treq}(ratio)=15.85 V_{ireq}(ratio)=1.0 F_{req}(ratio)=1.11

solutions to the same equations in terms of the predetermined variables, so the values will no longer be computed design equal to the reference values, but will be values nearly the reference values ruled by the objective function and its data processing method. So Figure 3a, 3b and 3c give the resulted F(ω_i), V_i (ω_i), G_T (ω_i) - frequency variations and network parameters for NE329S01 transistor. –T- type matching circuit of all kinds of optimization vector. Bandwidth of amplifier is between 3 GHz and 11 GHz. T-type matching circuit parameters for NE329S01 transistor are found as follows: ℓ_1=0.8122cm, ℓ_2=12.8437cm, ℓ_3= 0.5079cm, $\boldsymbol{Z_{o1}}$= 37.0384Ω, $\boldsymbol{Z_{o2}}$=46.3369Ω, $\boldsymbol{Z_{o3}}$=142.439Ω ; ℓ_4=1.1654cm, ℓ_5= 13.7641cm, ℓ_6= 0.34809cm, $\boldsymbol{Z_{o4}}$=144.8575Ω, $\boldsymbol{Z_{o5}}$=77.5388Ω, $\boldsymbol{Z_{o6}}$=88.6165Ω.

References

1. Güneş, F., Tepe, C.: Gain-Bandwidth Limitations for a Microwave transistor. RF and Microwave Computer Aided Engineering 10 (2002)
2. Güneş, F., Gürgen, F., Torpi H.: Signal-Noise Neural Triplet Network Model for Active Microwave Device, IEE Proc- Circuit Devices and Systems 143(1) (1996) 1-8
3. Güneş, F., Torpi, H., Gürgen F.:A Multidimensional Signal-Noise Neural Model for Microwave Transistors, IEE Proc- Circuit Devices and Systems 145(2) (1998) 11-117.
4. Güneş, F., Güneş, M., Fidan, M.: Performance Characterization of a Microwave Transistor, IEEE Proc. Circuits Devices System 141 (1994) 337-344
5. Güneş, F., Çetiner, B.A.: Smith Chart Formulation of Performance Characterization for a Microwave Transistor, IEEE Proc. Circuits Devices Syst. 145 (1998) 1-10
6. Aliyev, Ilgar : Design of the Microwave Amplifier using the Performance (F,Vi ,GT) Triplets, MSc Thesis submitted to the Science Research Institute of Yıldız Technical University, Istanbul, Turkey (2001)
7. Ingber, L.: Very Fast Simulated Reannealing. Mathl. Comput. Modeling **12** (1989) 967–993
8. Ingber, L., Rosen, B.: Genetic Algorithms and Simulated Reannealing. Mathl. Comput. Modeling **16** (1992) 87–100
9. Manzak, A., Goksu, H.: Application of Very Fast Simulated Reannealing to Low Power Design. Lecture Notes in Computer Science, vol. 3553 (2005) 308-313
10. Goksu, H., Kaya, M. A., Kokce, A.: Solution of the Inverse Electromagnetic Problem of Spontaneous Potential (SP) by Very Fast Simulated Reannealing (VFSR): Lecture Notes in Computer Science, vol. 3611 (2005) 618-621

An Intelligent System for the Heatsink Design

Yao-Wen Hsueh, Hsin-Chung Lien, and Ming-Hsien Hsueh

Department of Mechanical Engineering,
Northern Taiwan Institute of Science and Technology,
2, Xue Yuan Rd., Peitou, Taipei 112, Taiwan, ROC
{ywhsueh, hclien, mhhsueh }@ntist.edu.tw

Abstract. Via two-stage back-propagation neural network (BNN) learning algorithm, this paper establishes the relationship between different heatsink design parameters and performance evaluation, and induces 5 corresponding performance outputs from 6 different heatsink design and operating condition parameters (inlet airflow velocity, heatsink length or width, fin thickness, fin gap, fin height and heatsink base height) by using Computation Fluid Dynamics (CFD). After two stages well-trained, the BNN model with error compensator is able to accurate estimate the output values under different heatsink design and operation conditions.

1 Introduction

As the volume of semiconductive component is getting more miniaturized and the density of component in the unit area on circuit board is getting higher, the temperature rises rapidly during the operation of system, and the stability of system is further worsened. In order to dispel the concentrated heat energy effectively to the external environment for maintaining a smooth operation of system under a safe temperature, the cooling design and analysis have become a very important issue. Besides, electronic heatsink manufacturers are confronted with competition of rapid business. How to design heatsink rapidly and show its performance evaluation immediately so as to propose a heatsink that can meet the quoted operating temperature is a major key to the competitiveness of the industry.

The studies presented in recent years included Kraus and Bar-Cohen's [1] which resolved the cooling problem of electronic module by means of heat transfer. Bar-Cohen also carries out some studies on the cooling of chip packages [2-4]. Besides, Bar-Cohen and Krueger [5] introduce a compact model for chip packages. Over the cooling problem of electronic components, Nakayama [6] proposed the construction of electronic database and the control of heat transfer environment to strengthen the functions of electronic components. Most above themes are the heat transfer modeling studies related to electronic assembling or the cooling analysis of system by means of experiments. Nevertheless, the above studies have not focused on the various input design parameters of cooling system and the 5 corresponding performance outputs to make an integrated analysis of their mutual effects. Back-propagation neural network (BNN) has been taken to solve the problems with classification and prediction [7-9]. Thus, for a faster performance evaluation of different heatsink designs, this paper employs BNN with an error compensator to increase its accuracy.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 821 – 827, 2006.

2 Problem Description

The heatsink base is the part with the closest contact with the CPU die. Another part of the heatsink is the heatsink fin, which is the major part that achieves cooling effects. When the heatsink base absorbs the high temperature produced by the CPU die, the heat is guided upward to the heatsink fin. And the function of fin is to allow larger area to be blown by the electric fan so as to accelerate the cooling effect. As shown in Figure 1, the geometric parameters of heatsink include 6 items: heatsink width and length, heatsink base height, fin height, thickness and gap. Assume the heatsink material is the aluminum, the heating element (dummy die) material is copper, and the dimension of dummy die is 20 mm×20 mm, the heatsink is square. In the numerical wind tunnel, the heatsink is cooling under the different wind airflow velocity. Therefore, the six inputs of the BNN are heatsink length (h_l), fin thickness (f_t), fin height (f_h), fin gap (f_g), heatsink base height (b_h) and inlet airflow velocity (V_{in}).

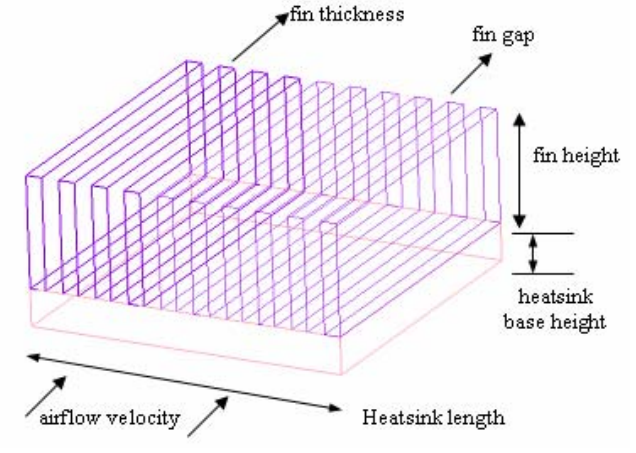


Fig. 1. The illustration of a heatsink

The important references of evaluating the cooling performance are the thermal resistance (θ_r ,℃/W), pressure drop (△P, N/m^2), airflow velocity drop (△V, m/s), average heatsink temperature rise (△T, ℃) and the loss coefficient caused by airflow passing through the fins of the heatsink (L_c, N-S^2/m-Kg). The θ_r is defined as the temperature gap between center point at the top surface of a chip (here is a dummy die) and the ambient divided by the power input of the dummy die. The △P and △V are defined as the average pressure and airflow velocity drop caused by airflow passing through the fins of the heatsink, respectively. The loss coefficient is defined as the pressure drop divided the airflow kinetic energy. Therefore, these five references of evaluating the cooling performance are fed into BNN as the desired outputs.

3 Basic Theories

3.1 Computation Fluid Dynamics (CFD)

The CFD numerical process can solve the heat and fluid related governing equations, and the fluid flow, heat transfer and mass transfer could be determined through this simulation. One of the CFD scheme, finite volume method (FVM) is widely used in the computational fluid dynamics field. In the FVM, the domain is discredited into a finite set of control volumes or cells. The general conservation (transport) equation for mass, momentum, energy, etc., are discredited into algebraic equations. The general conversation equation is shown in the Equation 1. The CFD simulation is done by the Icepak which is finite-volume method base CFD software.

$$\underbrace{\frac{\partial}{\partial t}\int_V \rho\phi dV}_{\text{Unsteady}} + \underbrace{\oint_A \rho\phi V \cdot dA}_{\text{Convection}} = \underbrace{\oint_A \Gamma\nabla\phi \cdot dA}_{\text{Diffusion}} + \underbrace{\int_V S_\phi dV}_{\text{Generation}} \tag{1}$$

Eqn.	ϕ
Continuity	1
x-mom.	u
y-mom.	v
energy	h

3.2 Back-Propagation Neural-Network (BNN)

Generally, there are three kinds of processing units in the neural network: input layer nodes, hidden layer nodes and output layer nodes. The input to a node can be expressed as

$$net_{pj} = \sum_i w_{ji} O_{pi} \tag{2}$$

where net_{pj} is summation function for all the input signals, w_{ji} is weight form the ith node to the jth node and O_{pi} is output of the ith node. The input-output behavior is described by a sigmoid function, that is

$$O_{pj} = \frac{1}{1+e^{-(net_{pj}+b_j)}} \tag{3}$$

where b_j is the bias of the sigmoid function. Then the learning process is applied to obtain the proper weights in the neural network. The actual output pattern is compared to the desired output pattern and the square error, *Ep*, can be expresses as

$$E_p = \frac{1}{2}\sum_j (T_{pj} - O_{pj})^2 \tag{4}$$

where T_{pj} is desired output of the jth node, then, the error is propagated from the output layer to input layer of the network. In order to minimize the error, the gradient of the error with respect to the weights of the nodes is used to modify the weights of the nodes. That is

$$dw_{ji} = -\eta \frac{\partial E_P}{\partial w_{ji}} \tag{5}$$

where dw_{ji} is the incremental change of weight from the i[th] node to the j[th] node and η is learning rate. As a result, all the weights of the nodes randomized with the initial values can be appropriately adjusted to minimize the error between the desired and actual outputs.

3.3 Estimate the Heatsink Performance with Error Compensator

Via two-stage BNN training algorithm, this paper establishes the relationship between different heatsink design parameters and performance evaluation, and induces 5 corresponding performance outputs from the abovementioned 6 different heatsink design and operating condition parameters by using CFD. In the first stage of training, the 6 different heatsink design and operating condition parameters and the 5 corresponding performance evaluations are considered as the input values and the desired output values of BNN respectively for completion of mathematical corresponding relationship, as shown in Fig. 2(a).

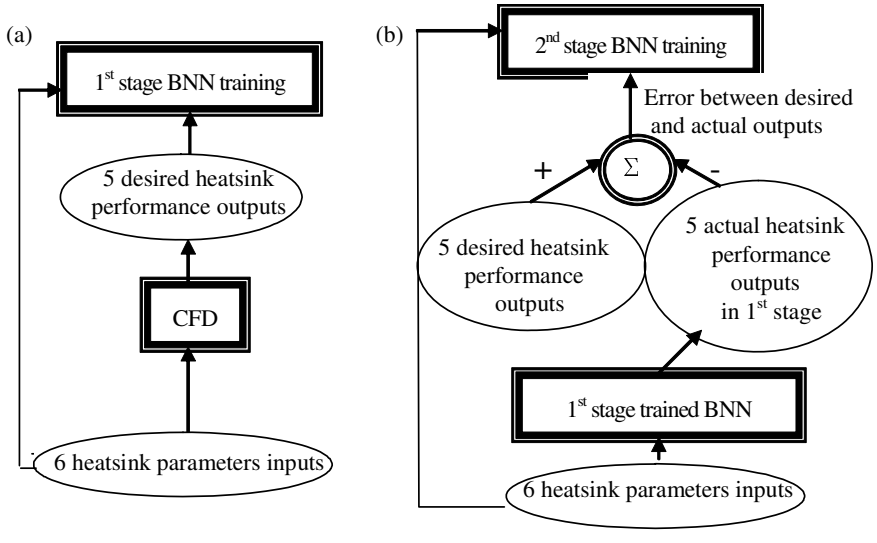


Fig. 2. Two-stage Back-Propagation Neural Network (BNN) learning structure

Then, these 6 parameters are entered to the well-trained first-stage BNN for acquisition of 5 corresponding actual performance evaluations, as shown in Fig. 2(b). In the second stage of training, the 6 different heatsink parameters and the error caused between the desired performance evaluation and the actual performance evaluation are considered as the second-stage BNN input values and the desired output values respectively for completion of mathematical corresponding relationship so as to acquire the error compensator values of the 5 performance evaluations, as shown in Fig. 2(b). Finally, The 6 parameters are entered to the well-trained first and second stage BNN, the error compensator values are added to the 5 corresponding actual performance evaluations in order to obtain the final 5 performance evaluations, as shown in Fig. 3.

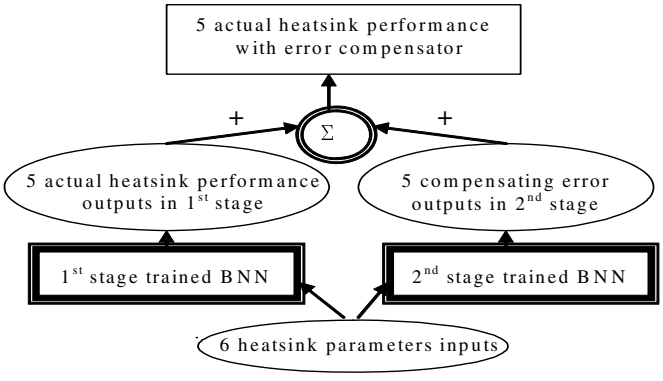


Fig. 3. BNN testing structure with error compensator

4 Examples and Results

The 6 design and operating condition parameters of the 80 different sets of heatsink are brought in the CFD, in which h_l is 38-50mm , f_t is 0.1-3mm, f_h is 5-45mm, f_g is 0.1-3.5mm, b_h =1-8mm and V_{in} is 0-5m/s. Among them, 60 sets of data include six heatsink parameters and five desired performances of the heatsink are taken as the training patterns of BNN. After 15000 epochs, the error values of the BNN training in the above two stages tend to be restrained. The remaining 20 sets of data presented in Table 1 are taken as the testing data and brought in the well-trained two-stage BNN. Then the two-stage heatsink performance evaluation can be obtained, and the results are presented in Fig. 4.

Each different set of data takes about 30 minutes for CFD simulation. But after well training, the BNN model with fixed node and weights can predict thousands sets data of heatsink performance in minutes. The RMS (Root-Mean-Square) error between the 5 desired performance evaluation acquired from the 6 design and operating condition parameters after being brought in the well-trained first-stage BNN, and the actual performance evaluation calculated by CFD is **8.44**%. After the second-stage calculation of error compensator, it can be found that the RMS error is reduced to **4.27**%. Therefore, with the second stage BNN training error compensator, the accuracy of estimating heatsink performance increases about 48%.

Table 1. The 6 design and operating condition parameters of heatsink

ID	V_{in}	h_l	f_t	f_g	f_h	b_h	ID.	V_{in}	h_l	f_t	f_g	f_h	b_h
1	1.25	38	1	1.8	32.5	6	11	4.37	39.5	2.3	2.2	25	2.5
2	1.25	50	0.5	1.1	25	4.9	12	4.37	48.5	1.9	1.8	19.4	1.6
3	2.5	38	1.5	3	17.5	4.9	13	1.09	47.4	2.1	2.5	33.4	5.0
4	2.5	50	1.0	2.4	10	3.8	14	1.8	42.3	1.5	2.1	33.4	2.1
5	3.75	41	2.5	1.8	10	4.9	15	1.8	45.7	2.1	1.1	20.8	3.5
6	3.75	47	1.0	3	25	1.5	16	3.2	40.6	1.8	1.4	33.5	3.5
7	1.56	39.5	1.1	1.8	30.6	5.4	17	3.2	47.4	1.5	1.1	29.2	2.8
8	1.56	48.5	0.8	1.3	25	4.5	18	1.45	42.7	1.3	1.5	21.8	3.0
9	3.44	39.5	1.9	1.3	36.3	3.5	19	1.45	45.3	1.7	2	28.2	4.0
10	3.44	48.5	1.5	0.8	30.6	2.5	20	3.03	41.5	1.7	1.5	31.3	3.5

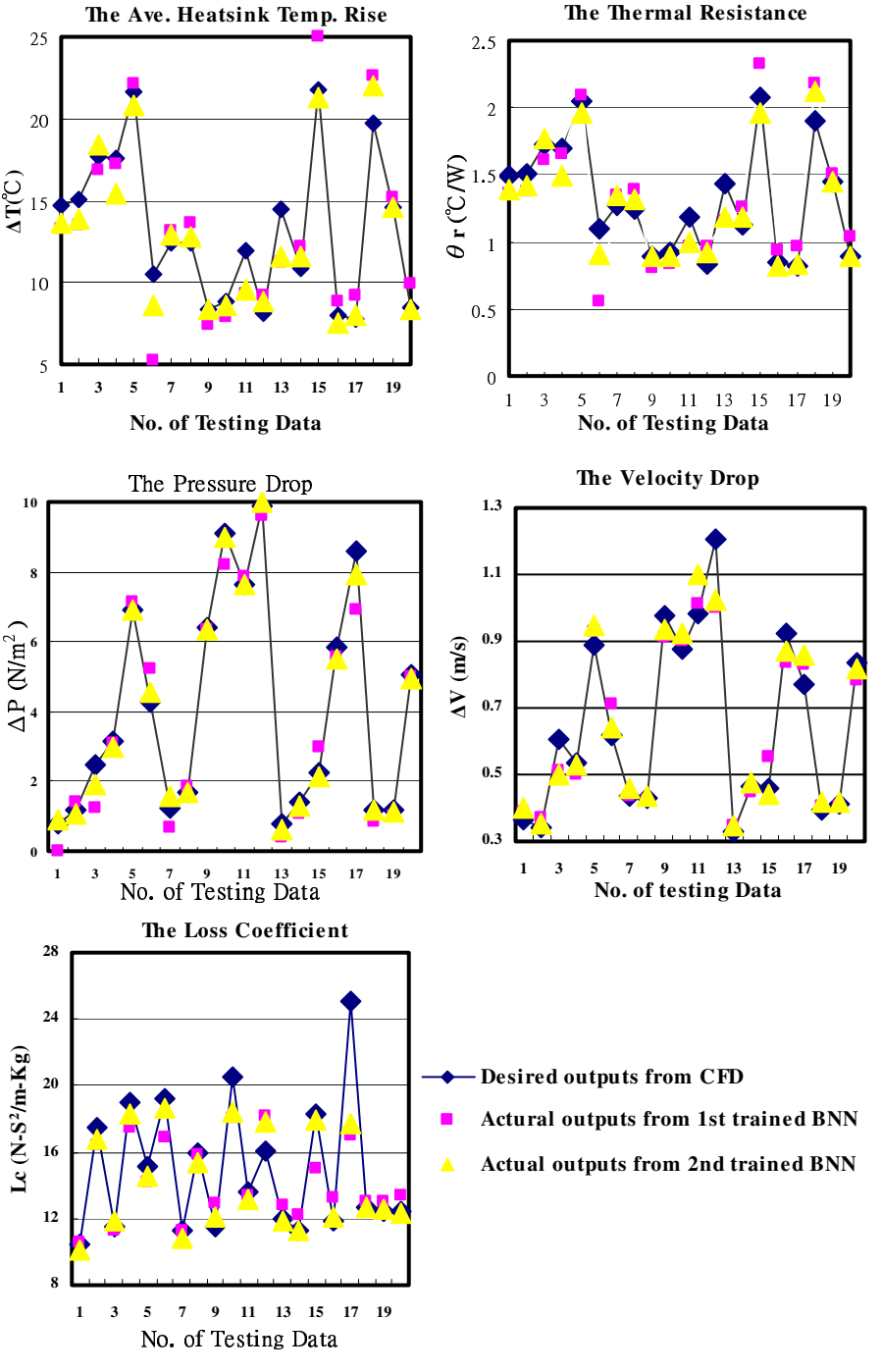


Fig. 4. The 5 heat sink performance outputs from two well-trained BNN structure

5 Conclusions

This paper proposes a new approach that uses two-stage BNN to obtain a faster and more accurate estimation of the heatsink performance under some specific conditions. Each different set of data takes about 30 minutes for CFD simulation. But after well training, the BNN model with fixed node and weights can predict thousands sets data of heatsink performance in minutes. Therefore, after well trained under the training and testing data taken from CFD, the two stage BNN model can provide a very fast estimation under acceptable accuracy. The two stage BNN model make not only the prediction of heatsink performance under some specific conditions in a short time, but also a better heatsink design based on the two stage BNN model.

References

1. Kraus, A., Bar-Cohen, A.: Thermal Analysis and Control of Electronic Equipment, Hemisphere Publishing Corp. (1983)
2. Bar-Cohen, A.: State of the Art and Trends in the Thermal Packaging of Electronic Equipment. ASME Journal of Electronic Packaging 114 (1992) 257-270
3. Bar-Cohen, A.: Thermal Management of Air- and Liquid- Cooled Multichip Modules. IEEE Transactions CPMT 10(2) (1988) 159-175
4. Bar-Cohen, A.: Thermal Packaging for the 21st Century. Invited Talk, 5th Therminic – International Workshop Thermal Investigations of ICs and System, Rome, Italy, Oct.3-6, (1999)
5. Bar-Cohen, A., Krueger, W.B.: Thermal Characterization of Chip Packages: Evolutionary Development of Compact Models. IEEE Transactions CPMT - Part A (1997)
6. Patanker, S.V.: Numerical Heat Transfer and Fluid Flow, Hemisphere Publishing Co. (1980)
7. Hsueh, Y.W.: Sensing Tool Breakage in Face Milling with a Neural Network. Int. J. Machine Tool Manufacture 34 (1994) 341-350
8. Lien, H.C., Lee, S.: Application of Pattern Recognition Principles for Grading Textile Yarns. Textile Res. J. 72 (2002) 320-326
9. Kuan, Y.D., Lien, H.C.: The Integration of the Neural Network and Computational Fluid Dynamics for the Heatsink Design. Lecture Notes in Computer Science (LNCS), Vol. 3498 (2005) 933-938

Learning Shape for Jet Engine Novelty Detection

David A. Clifton[1,2], Peter R. Bannister[1], and Lionel Tarassenko[1]

[1] Department of Engineering Science, Oxford University, UK
{davidc, prb, lionel}@robots.ox.ac.uk
[2] Oxford BioSignals Ltd., Magdalen Centre, Oxford Science Park, Oxford, OX4 4GA, UK

Abstract. Application of a neural network approach to data exploration and the generation of a model of system normality is described for use in novelty detection of vibration characteristics of a modern jet engine. The analysis of the *shape* of engine vibration signatures is shown to improve upon existing methods of engine vibration testing, in which engine vibrations are conventionally compared with a fixed vibration threshold. A refinement of the concept of "novelty scoring" in this approach is also presented.

1 Introduction

Novelty detection, defined to be the identification of departures from a model of "normal" behaviour of a system, is particularly suited to the monitoring of jet engines, in which examples of normal engine behaviour greatly outnumber examples of abnormal behaviour (in common with many high-integrity systems).

Current engine testing techniques depend on the comparison of engine vibration to a simple maximum vibration threshold. This paper presents a method for novelty detection of engine vibration characteristics developed for use with a modern civil jet engine to improve these existing methods, and provides a worked example of the use of neural networks in developing a model of normality for use within novelty detection.

Existing work in the analysis of the *shape* of engine vibration characteristics is extended for use with the new engine type, in which *novelty scores* are computed for each pattern and compared to a decision threshold. This paper presents a refinement of the score definition, such that novelty scores have an intuitive interpretation. Results of applying this technique are shown to emphasise the benefits over existing methods, and the exploration of data and resultant models of normality using neural network methods are presented.

2 Example Data

This investigation considers a modern civil jet engine, consisting of several rotating engine *shafts*, mounted concentrically. In normal operation, air enters the

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 828–835, 2006.

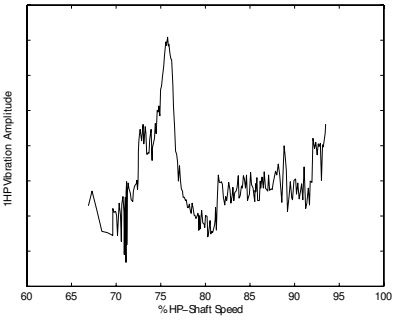


Fig. 1. Speed-based vibration signature $a(s)$ for an example engine (y-axis units anonymised)

low pressure ("LP") shaft for compression, and is passed to the *high pressure* ("HP") shaft, before ultimately being used within the fuel combustion chamber. A further *radial drive* shaft, mechanically connected to the HP shaft, is used to start the engine and provide power during take-off. Vibration energy at the frequency of rotation of each shaft is referred to as the *fundamental tracked order* of each shaft.

During product testing, engines of the class considered within this investigation perform a controlled two-minute acceleration from idle to maximum speed, followed by a two-minute deceleration back to idle. Vibration amplitude levels of the fundamental tracked orders corresponding to each shaft are recorded during these tests, from which the speed-based vibration signature $a(s)$ is constructed for rotational speeds s of each engine shaft.

Engines are tested against a contractual limit H which must not be exceeded for the engine to be deliverable; i.e., $a(s) < H$.

An example vibration signature from the data set used in this investigation is shown in Figure 1. The vibration signature for the fundamental tracked order of an engine shaft is constructed with respect to the percentage speed *for that shaft*; thus, amplitude of vibration of the HP shaft is plotted against the percentage of maximum speed for the HP shaft.

The data set used in the investigation described in this article consists of 71 engine examples, initially divided into four sub-sets $D_1 \ldots D_4$, according to their maximum vibration amplitude $\max\{a(s)\}$ compared with the contractual vibration limit H, as shown in Table 1. Note that sub-set D_4 is formed from recordings of engines with seeded fault conditions, in which small masses are deliberately applied to the engine fan blades.[1] This unbalance is noted by engine

[1] Weights are deliberately applied to fan (and sometimes turbine) blades in order to correct any imbalances during rotation of those blades about the shaft to which they are connected. In order to determine the effect upon engine vibration of adding masses to fan blades, a small number of weights are applied, forcing the blades into a state of unbalance. Examples from class 4 are recorded from an engine undergoing this procedure.

Table 1. Data classified according to maximum vibration amplitude

Sub-set	$\lvert D_n \rvert$	Class membership criteria
D_1	29	Examples for which $\max\{a(s)\} < 0.9H$
D_2	15	Examples for which $0.9H \leq \max\{a(s)\} \leq H$
D_3	17	Examples for which $H < \max\{a(s)\}$
D_4	10	Examples in which weights are deliberately applied to the engine fan

manufacturers to result in very large vibration amplitude levels of the LP shaft, with approximately normal behaviour in other shafts. Novelty detection *applied to the HP shaft* should therefore not identify examples from sub-set D_4 as being novel.

3 Quantisation of Vibration Signatures

Increasing dimensionality of data requires exponentially increasing numbers of patterns within the data set used to construct a general model; this is termed the *curse of dimensionality* [2]. In order to avoid this problem, each signature is summarised by a *shape vector* $\mathbf{x}$. This is performed by computing a weighted average of the vibration amplitude values $a(s)$ over $N = 10$ speed sub-ranges [7]. The n^{th} dimension of shape vector $\mathbf{x}^n$, for $n = 1 \ldots N$, is defined to be:

$$\mathbf{x}^n = \int_{s_{\min}}^{s_{\max}} a(s)\omega_n(s)ds \tag{1}$$

in which the vibration amplitude $a(s)$ is integrated over the speed range $s : [s_{\min}\ s_{\max}]$, using weighting functions $\omega_n(s)$, for $n : 1 \ldots N$.

In setting the interval of speeds $s : [s_{\min}\ s_{\max}]$, the operational range of each shaft must be considered independently. For example, the gas-turbine engines of the class considered within this investigation are idle at HP shaft speeds below 60% of maximum rotational speed, during which no measurements of vibration amplitude are made. Thus, the interval [60% 100%] of maximum shaft speed is used when quantising vibration signatures from the HP shaft.

Component-wise normalisation [1, 3] (in which the normalised vibration signature has zero mean and unit variance at each quantised speed) was found to emphasise the separation between "good" (D_1, D_2) and "bad" engines (D_3). Labelling the data by comparison with a contractual limit is intended only as initial guide to model construction.

4 Visualisation Using a NeuroScale RBF Network

In order to confirm the results of normalisation, the data set was visualised by projecting the set of 10-dimensional shape vectors into 2 dimensions.

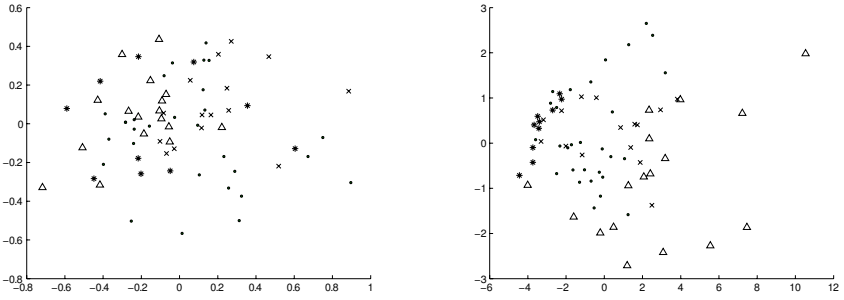


Fig. 2. Projections of HP shaft vibration signatures: (a) un-normalised; (b) component-wise normalisated. Data sub-sets $\{D_1 \ldots D_4\}$ are shown by $\{\bullet \times \triangle *\}$, respectively. Separation of "abnormal" D_3 patterns ($\triangle$) is clearly better in (b) than (a).

Topographic projection is a transformation that attempts to best preserve, in the projected space of lower-dimensionality (*latent space*, $\mathbb{R}^q$), distances between data in their original high-dimensional space (*data space*, $\mathbb{R}^d$). Typically $d > q$, $q = 2$ or 3. The *Sammon stress metric*[8] is based upon the distance d_{ij} between points (x_i, x_j) in $\mathbb{R}^d$, and the distance d^*_{ij} between projected points (y_i, y_j) in $\mathbb{R}^q$:

$$E_{\text{sam}} = \sum_{i=1}^{N} \sum_{j>i}^{N} (d_{ij} - d^*_{ij})^2 \tag{2}$$

in which the distance measure is typically Euclidean. The NeuroScale model [4, 5] trains a radial basis function (*RBF*) neural network to perform the mapping from $\mathbb{R}^d$ to $\mathbb{R}^q$, in which E_{sam} is minimised; i.e. distances between points are best preserved after projection.

A NeuroScale network was used for projecting shape vectors derived from the example data set described previously, with $d = 10$ inputs (corresponding to the number of elements in each shape vector) and $q = 2$ outputs (for 2-dimensional projection).

Projection of all 10-dimensional shape vectors before and after component-wise normalisation is shown in Figure 2, with clear separation of "abnormal" patterns (from sub-set D_3) evident.

5 Modelling Normality

The k-means clustering algorithm was used, as described in [6], to construct a model of normality from "normal" patterns (i.e. those from sub-sets D_1, D_2). In this method, the distribution of "normal" patterns is defined by $\mathbf{C}_k$ cluster centres in $\mathbb{R}^{10}$ space, each with an associated *cluster width* σ_k. A novelty score $z(\mathbf{x})$ may be computed for shape vector $\mathbf{x}$ with respect to the K cluster centres:

$$z(\mathbf{x}) = \min_{k=1}^{K} \frac{d(\mathbf{x}, \mathbf{C}_k)}{\sigma_k} \tag{3}$$

where $d(\mathbf{x}, \mathbf{C}_k)$ is Euclidean distance. We propose a new definition of width σ_k:

$$\sigma_k = \sqrt{\frac{1}{I_k} \sum_{i=1}^{I_k} d(\mathbf{x}_i, \mathbf{C}_k)^2}. \tag{4}$$

for the I_k points which have closest cluster centre $\mathbf{C}_k$. This allows an intuitive interpretation of the magnitude of novelty scores: novelty scores $z(\mathbf{x})$ computed using (3) are the number of standard deviations that pattern $\mathbf{x}$ lies from its closest cluster centre, relative to the distribution of training data about $\mathbf{C}_k$. A threshold H_z is applied to $z(\mathbf{x})$ such that all patterns $z(\mathbf{x}) \geq H_z$ are classified "abnormal". H_z is set to best separate the "normal" and "abnormal" patterns in the data set.

Investigation of the placement of cluster centres is possible using the NeuroScale RBF technique. The position of cluster centres with respect to the data set may be determined by projection using the NeuroScale network previously trained using the patterns from the example data set. Selecting a model of normality for use can be assisted through use of the projections generated by the neural network, such that the cluster centres accurately represent the distribution of patterns from the data set.

6 Results of Application to Example Engine Data

The NeuroScale RBF neural network method is further used to examine the results of novelty detection. Selected results are presented within this section from the HP shaft, and the radial drive shaft.

6.1 HP Shaft Results

Figure 3 shows all HP shaft patterns projected into 2 dimensions.

Of the 17 "abnormal" patterns (from sub-set D_3), 12 were classified as "abnormal" by the novelty detection scheme. From these 12 patterns, 5 corresponded to engines with vibration signatures that, though above the simple contractual limit (thus placing them into sub-set D_3, which would thus not pass product testing), exhibited very similar signature shapes to engines that were deemed fit for service. This indicates that the comparison of an engine's maximum vibration level to a fixed contractual limit, as is the standard means of pass-off testing, may not adequately detect abnormal engines.

Of those engines which were deemed fit for service, based on comparison to the simple threshold, two were classified "abnormal" by the scheme investigated in this article. These corresponded to engines which were identified by engine developers as requiring rebuilding due to high vibration levels within the HP shaft.

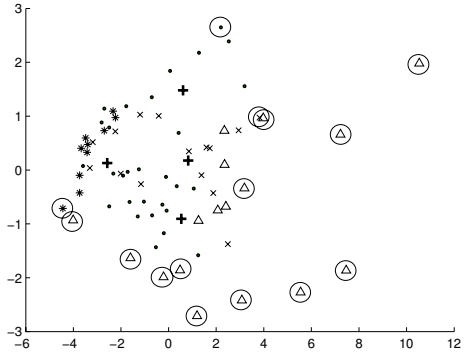


Fig. 3. Projection of all HP shaft patterns. Cluster centres C_K are shown as '+' symbols. Patterns classified as "abnormal" are circled. Data sub-sets $\{D_1 \ldots D_4\}$ are shown by $\{\bullet \times \triangle *\}$, respectively.

The NeuroScale projection also shows that engines that were deliberately unbalanced (from sub-set D_4) exhibit similar vibration signatures, which are classified "normal", agreeing with the observation that weights applied to the engine fan only affect vibration of the LP shaft.

6.2 Radial Drive Shaft Results

Due to no contractual vibration limit being specified for the radial drive shaft, patterns for this shaft could not be divided into the 4 sub-sets described previously. With no prior distinction made between "normal" and "abnormal" patterns, a different approach was taken, in which engines that were released into

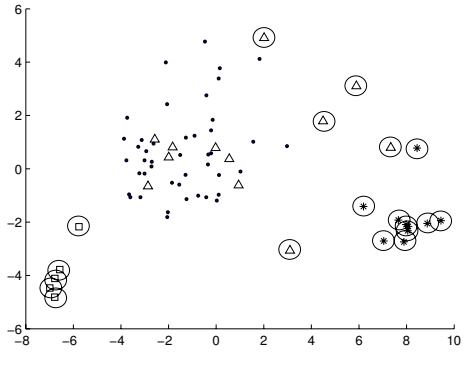


Fig. 4. Projection of all radial drive shaft patterns. Patterns classified as "abnormal" are circled. Engines that failed testing are shown as $\triangle$. Engines that were re-fitted during service are shown as $\square$. Engines with forced unbalance are shown as $*$.

service following pass-off testing formed the training set for the construction of the model of normality.

Figure 4 shows a NeuroScale projection of all patterns derived from radial drive shaft vibration signatures.

The NeuroScale projection shows that the older engines, re-fitted during service, form a distinct cluster in the lower-left of the plot, indicating that extended use of the engine in service results in a change in vibration characteristics. Furthermore, it can be seen that unbalanced patterns show significant difference to "normal" engines in the radial drive shaft, indicating that this condition is not limited only to the LP shaft.

It can also be seen that several engines which fail pass-off testing appear "normal" according to radial drive shaft vibration levels.

7 Conclusions

The method of vibration analysis presented in this paper has shown that the "shape analysis"process is effective for novelty detection in vibration data from new engine classes. Results from analysis of vibration signatures from each shaft of the example engine have shown that the majority of patterns derived from abnormal engine recordings are correctly classified, whilst false-positive detections associated with normal patterns are low.

The method presented provides a more accurate assessment of abnormality than comparison of vibration levels to a simple threshold, whilst visualisation by neural network allows changes in engine condition to be examined. Hence, increasing accuracy of engine assessment allows improved control of engine maintenance, and enhances the diagnostic process.

Acknowledgements

The authors gratefully acknowledge the support of the EPSRC, and the assistance of Dennis King, Mark Walters, and Steve King of Rolls-Royce Plc.; and Nicholas McGrogan of Oxford BioSignals Ltd.

References

1. Bannister, P.R., Tarassenko, L., Shaylor, I.: SPANN-04-PB1 Shape Analysis Tools, Department of Engineering Science, University of Oxford, Technical Report (2004)
2. Bishop, C.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)
3. Clifton, D.A.: Condition Monitoring of Gas-Turbine Engines. Department of Engineering Science, University of Oxford, Transfer Report (2005)
4. Lowe, D., Tipping, M.E.: Feed-Forward Neural Networks and Topographic Mappings for Exploratory Data Analysis. Neural Computing and Applications 4(2) (1996) 83-85
5. Nabney, I.T.: NETLAB, Algorithms for Pattern Recognition. Springer-Verlag, New York (2002)

6. Nairac, A., Corbett-Clark, T.A., Ripley, R.M., Townsend, N.W., Tarassenko, L.: Choosing an Appropriate Model for Novelty Detection. In: Proc. 5th IEE Int. Conf. on Artificial Neural Networks, Cambridge (1997) 117-122
7. Nairac, A., Townsend N., Carr, R., King, S., Cowley, P., Tarassenko, L.: A System for the Analysis of Jet Engine Vibration Data. Integrated Computer-Aided Engineering 6(1) (1999) 53-56
8. Sammon, J.W.: A Non-Linear Mapping for Data Structure Analysis. IEEE Transactions on Computers C-18(5) (1969) 401-409

Support Vector Machine in Novelty Detection for Multi-channel Combustion Data

Lei A. Clifton[1], Hujun Yin[1], and Yang Zhang[2]

[1] School of Electrical and Electronic Engineering,
l.wang-2@postgrad.manchester.ac.uk, h.yin@manchester.ac.uk
[2] School of Mechanical, Aerospace and Civil Engineering,
y.zhang@manchester.ac.uk
The University of Manchester, Manchester, M60 1QD, UK

Abstract. Multi-channel combustion data, consisting of gas pressure and two combustion chamber luminosity measurements, are investigated in the prediction of combustion instability. Wavelet analysis is used for feature extraction. A SVM approach is applied for novelty detection and the construction of a model of normal system operation. Novelty scores generated by classifiers from different channels are combined to give a final decision of data novelty. Comparisons between the proposed SVM method and a GMM approach show that earlier identification of combustion instability, and greater distinction between stable and unstable data classes, are achieved with the proposed SVM approach.

1 Introduction

Combustion instability, caused by the resonant coupling between heat (generated by combustion) and acoustic pressure, is a major problem in the operation of jet engines and power generators. Early warning of combustion instability is required in order to prevent catastrophic failure. Combustion image data from a high-speed camera have been investigated to predict instability in [12], in which a Gaussian mixture model (GMM) was constructed to identify novel data.

Two data sets generated by a Typhoon G30 combustor are investigated in this paper. The combustor was operated at atmospheric pressure, providing data in stable and deliberately unstable modes of combustion. Unstable combustion is achieved by increasing fuel flow rates above some threshold, with constant air flow rate. Both data sets represent measurements taken as the combustor moves from stable to unstable combustion modes.

Each data set consists of three channels (with sampling frequency of 1 KHz), the first of which is the gas pressure of the fuel propane (C_3H_8) in the pilot burner. For stable combustion, the swirl air flow rate was 0.039 Kgs^{-1}, the fuel supplied to the main and the pilot burners were fixed at flow rate 22.61×10^{-4} Kgs^{-1} and 10.20×10^{-4} Kgs^{-1} respectively. In order to initiate combustion instabilities, the flow rates of fuel supplied to the main and pilot burners were increased to 26.18×10^{-4} Kgs^{-1} and decreased to 4.37×10^{-4} Kgs^{-1} respectively. The second and third channels are luminosity measurements recorded within the

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 836–843, 2006.

combustion chamber. A bundle of fine optical fibres were mounted at the rear focal point of a Nikon 35 mm camera, such that all light passing through the front lens was collected. The flame luminosity from the combustion chamber was measured using this system. The fibre optic bundle was bifurcated, each channel connected to a photomultiplier (ORIEL model 70704). This allowed the measurement of chemiluminescent emitters of C_2 radicals (visible at light wavelength 513 nm), and the global intensity of unfiltered light, corresponding to the second and third channels in the data sets.

Support Vector Machines (SVMs) have been applied to many novelty detection problems, such as jet engine vibration testing [5], image retrieval [1], signal segmentation [3] and fMRI analysis [4]. The optical measurement methods described above have been applied to study flame dynamics of unstable combustion [9]. On similar data, we apply one-class SVMs to multi-channel combustion data; an averaging fusion method is then applied to combine SVM classifications from different channels. The whole system is applied in an on-line fashion.

2 Wavelet Analysis for Feature Extraction

For combustion data, frequency domain features alone are not suitable for on-line novelty detection [12]. Features from wavelet analysis are used in the proposed method. Wavelet analysis represents a function in terms of basis functions, localised in both location and scale [2]. It is capable of revealing behavioural trends or transient periods within the data. Wavelet decomposition can be regarded as a multi-level or multi-resolution representation of a function $f(x)$, where each level of resolution j (except the initial level) consists of wavelets Ψ_I^j or wavelet-families $\Psi_I^{\mu,j}$, with the same scale but differing locations. Wavelet decomposition of a function $f(x)$, can be written

$$f(x) = \sum_{k \in K^0} c_k^0 \phi_k^0(x) + \sum_{j=1}^{+\infty} \sum_{\mu} \sum_{I \in \zeta^{\mu,j}} d_I^{\mu,j} \Psi_I^{\mu,j}(x), \tag{1}$$

where ϕ_k^0 and $\Psi_I^{\mu,j}$ are respectively n-dimensional scaling functions and wavelets of different families and levels of resolution; c_k^0 and $d_I^{\mu,j}$ are corresponding scaling coefficients and wavelet coefficients.

Mallat [8] developed a filtering algorithm for Discrete Wavelet Transforms. Given a signal S of length N, wavelet decomposition of the first level produces two sets of coefficients: approximation coefficients cA_1 and detail cD_1, by convolving S with a low-pass filter for approximation, and with a high-pass filter for detail, followed by dyadic decimation (down-sampling). Wavelet decomposition of the next level splits the approximation coefficients cA_1 in two parts using the same scheme, replacing S by cA_1, and producing cA_2 and cD_2, and so on.

We investigate two three-channel combustion data sets in this paper, which contain 5800 and 7500 data points respectively in each channel. These data points are divided into 75 and 58 sections respectively, with 100 data points in each section. Wavelet analysis with the Harr wavelet is applied to each section.

The mean values of approximation and detail coefficients extracted on the first level are used as two dimensional features in the input space.

3 Support Vector Machines for Novelty Detection

Novelty detection, defined as detecting departures in behaviour from a model of system normality, is applicable to combusiton systems in which often only "normal" data are available.

The training data available are "normal" data in the feature domain $\mathcal{X} : X = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$, where m is the number of observations. The training data can be mapped into another feature space through a feature mapping $\mathbf{\Phi}$: $\mathcal{X} \rightarrow F$. The kernel function operates on the dot product of the mapping function

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{\Phi}(\mathbf{x}_i) \cdot \mathbf{\Phi}(\mathbf{x}_j)). \tag{2}$$

A Gaussian kernel function is used here to suppress the growing distances for larger feature spaces [11]

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2\right), \tag{3}$$

where σ is the width parameter associated with the kernel function.

The strategy developed in [10] maps the data into the feature space corresponding to the kernel function, and separates them from the origin with maximum margin. The decision function is found by minimising the weighted sum of a support vector type regulariser and an empirical error term depending on an overall margin variable ρ and individual errors ξ_i

$$\min_{w \in F, \xi \in \mathbb{R}^m, \rho \in \mathbb{R}} \quad \frac{1}{2}\|w\|^2 + \frac{1}{\nu m}\sum_{i=1}^{m}\xi_i - \rho \tag{4}$$

$$\text{subject to} \quad (w \cdot \mathbf{\Phi}(\mathbf{x}_i)) \geq \rho - \xi_i, \quad \xi \geq 0, \tag{5}$$

where w is a weight vector in the feature space; ν serves as an upper bound on the proportion of outliers (defined to be those data which lie outside the estimated region of normality).

Constructing a Lagrangian, and setting the derivatives with respect to w equal to zero, leads to

$$w = \sum_{i=1}^{m}(\alpha_i \mathbf{\Phi}(\mathbf{x}_i)) \tag{6}$$

$$\alpha_i = \frac{1}{\nu m} - \beta_i \leq \frac{1}{\nu m}, \quad \sum_{i=1}^{m}\alpha_i = 1, \tag{7}$$

where α_i and β_i $\{\ \alpha_i, \beta_i \geq 0\ \}$ are Lagrangian multipliers.

In (6), patterns $\mathbf{x}_i$ with $\alpha_i > 0$ are called Support Vectors. The dual formulation is

$$\min_{\alpha \in \mathbb{R}^m} \sum_{ij} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \tag{8}$$

$$\text{subject to} \quad \sum_i \alpha_i = 1, \quad 0 \leq \alpha_i \leq \frac{1}{\nu m}. \tag{9}$$

The overall margin variable ρ can be calculated by

$$\rho = (w \cdot \mathbf{\Phi}(\mathbf{x}_i)) = \sum_{i=1}^{m} \alpha_j k(\mathbf{x}_i, \mathbf{x}_j). \tag{10}$$

Typically, the use of SVMs for novelty detection includes investigation of the effect of varying parameter ν [10, 5] for particular data sets. Combustion data may have differing lengths and sampling frequencies, so that the number of observations m varies. We define a smoothing parameter $P = \frac{1}{\nu m}$, observing that ν and m vary together and may be considered a single variable.

We use the function

$$g(\mathbf{x}) = 1 - (w \cdot \mathbf{\Phi}(\mathbf{x})) + \rho \tag{11}$$

to assign novelty values to data, such that abnormal data (i.e. those outside the one training class) take values close to one.

4 Combination of Classifiers

Novelty values $g(\mathbf{x})$, computed using (11), from each of the three channels are combined. Each data set is considered independently.

Different classifier combination strategies have been studied in [6] and [7]. In this investigation, novelty scores generated by different classifiers are combined according to the mean rule [7]:

$$\hat{g}(\mathbf{x}) = \frac{1}{L} \sum_{i=1}^{L} g_i(\mathbf{x}), \tag{12}$$

where $g_i(\mathbf{x})$ is the novelty score generated by the ith classifier, $i = 1, \ldots, L$, and L is the number of classifiers.

5 Results and Comparisons

5.1 Comparison of SVM Classifiers with GMM Classifers

The series of wavelet features generated for the two data sets are two dimensional for each channel, containing 58 and 75 data points respectively. The combustor operated in stable combustion mode until data point 27 and 37, for data sets 1

and 2 respectively. This corresponds to a time of approximately 2.8 s, as shown in Figure 1(a), in which the three channels of data set 1 are plotted.

After this period of stable combustion, a period of transient behaviour is seen, prior to operation in unstable combustion mode for the remainder of both data sets. 80% of these stable data points (21 for data set 1, and 29 for data set 2) are used as training data. The remaining 20% stable data, and all of the transient and unstable data, are used as test data. A zero-mean, unit-variance transformation is applied to all data, for normalisation prior to the feature extraction.

For each SVM classifier, Gaussian kernel width parameter $\sigma = 0.9$ is used, and smoothing parameter P is set to 100. A threshold $H_s = 0.1$ is used, such that all data with novelty score $g_i(\mathbf{x}) > H_s$ are classified "abnormal" with respect to the one-class model, generated from the ith classifier.

In order to compare the proposed SVM approach with other methods such as GMM approaches [12], we replace SVM classifers with equivalent GMM classifiers. The estimated probilities generated by Gaussian classifers in each channel are also combined according to the mean rule to give final estimated probability $\hat{p}(\mathbf{x}) = \frac{1}{L}\sum_{i=1}^{L}\hat{p}_i(\mathbf{x})$, where $\hat{p}_i(\mathbf{x})$ is the estimated probability of normality generated by the ith GMM classifier. Each Gaussian classifier has two mixture components, with a probability threshold $H_g = 0.02$. Data with $\hat{p}(\mathbf{x}) < H_g$ are regarded "abnormal".

5.2 Results and Discussions

Figure 1 shows time-domain signals and results of novelty detection applied to data set 1. Two types of test data are considered here: unseen stable (normal) data and unseen unstable (novel) data. Figure 1(b) illustrates the use of novelty threshold H_s to separate the novelty scores of stable and unstable data for the combined SVM classifier. It can be seen that, as the transient period from stable to unstable combustion is reached at data index 30, novelty scores increase above H_s, and are thus classified as "abnormal" with respect to the SVM model of normality. Figure 1(c) shows the estimated probablities genearated by the combined GMM classifier. At data index 31, the probability dropped sharply to almost zero, indicating that these data are novel with respect to the GMM model of normality.

In Figure 1(b) and (c), test (unseen) stable data show similar results to training stable data, indicating both combined SVM and combined GMM classifications are correct for stable data. However, the combined SVM classifier gives more stable results than the combined GMM classifer, assigning novelty scores of approximately zero for all stable data, while estimated probabilities from the combined GMM classifier fluctuate greatly on the same data.

Figure 1(d)-(f) show contour plots of $g_i(\mathbf{x})$ generated by SVM classifiers in each of the three channels of data set 1. It can be seen that the modelled regions corresponding to normality, in which novelty scores are close to zero, closely correspond to the distribution of training stable data, while unseen unstable data are considerably separated from stable data.

Table 1. Data index of the first novel classification, according to each SVM or GMM classifier and combined efforts

	Channel 1		Channel 2		Channel 3		Combined classifier	
	SVM	GMM	SVM	GMM	SVM	GMM	SVM	GMM
Data set 1	31	17	29	16	30	30	30	31
Data set 2	38	4	37	34	31	31	31	40

Table 1 shows the first data index for which $g_i(\mathbf{x}) > H_s$; i.e. when data are first deemed "abnormal" by the SVM classifier trained for each channel, and for the combined SVM classifier. Novel data are identified during transient periods by every classifier in all three channels of both data sets. The final decision is made by the combined classifier. Note that for data set 2, the combined classifier first identifies novel data at index 31, which is the earliest novelty detection of all three individual SVM classifiers for that data set, occurring within the third channel. This indicates the comparatively high novelty of data from the transient period in the third channel.

Table 1 also shows the first data index for which $\hat{p}_i(\mathbf{x}) > H_g$ in GMM approach. Note that the first novel classifications using the GMM approach can be false positive detections, occurring within the period of stable combustion (e.g. indices 16 and 17 within data set 1, and index 4 within data set 2), indicating an over-sensitivity of the individual GMM classifiers in comparison to the SVM equivalents.

When classifications are combined, the combined SVM classifer detects novel data at data index 30 and 31 for data set 1 and data set 2 respectively, while the combined GMM classifier detects novel data at data index 31 and data index 40. Although the over-sensitivity of the GMM classifiers has been reduced by combining their outputs, the result is worse than the combined SVM classifiers. This is caused by the great fluctuations of $\hat{p}_i(\mathbf{x})$ within the GMM approach, in which the combination of highly variant and inaccurate probability estimates leads to poorer performance than the SVM approach.

There are three critical parameters for SVM classifiers: the smoothing parameter P, the width parameter σ for the Gaussian kernel and the threshold H_s for novelty scores. Variation of these parameters causes dramatic differences in novelty detection results. The parameters used within this investigation are empirical values which generally work well for multi-channel combustion data.

6 Conclusion

SVM classifiers for each of the three channels perform well in the classification of unseen multi-channel combustion data, including unseen stable data and unseen unstable data. Novel data are identified by the combined classifier at an early stage of unstable combustion, i.e. the transient stage, for both data sets exam-

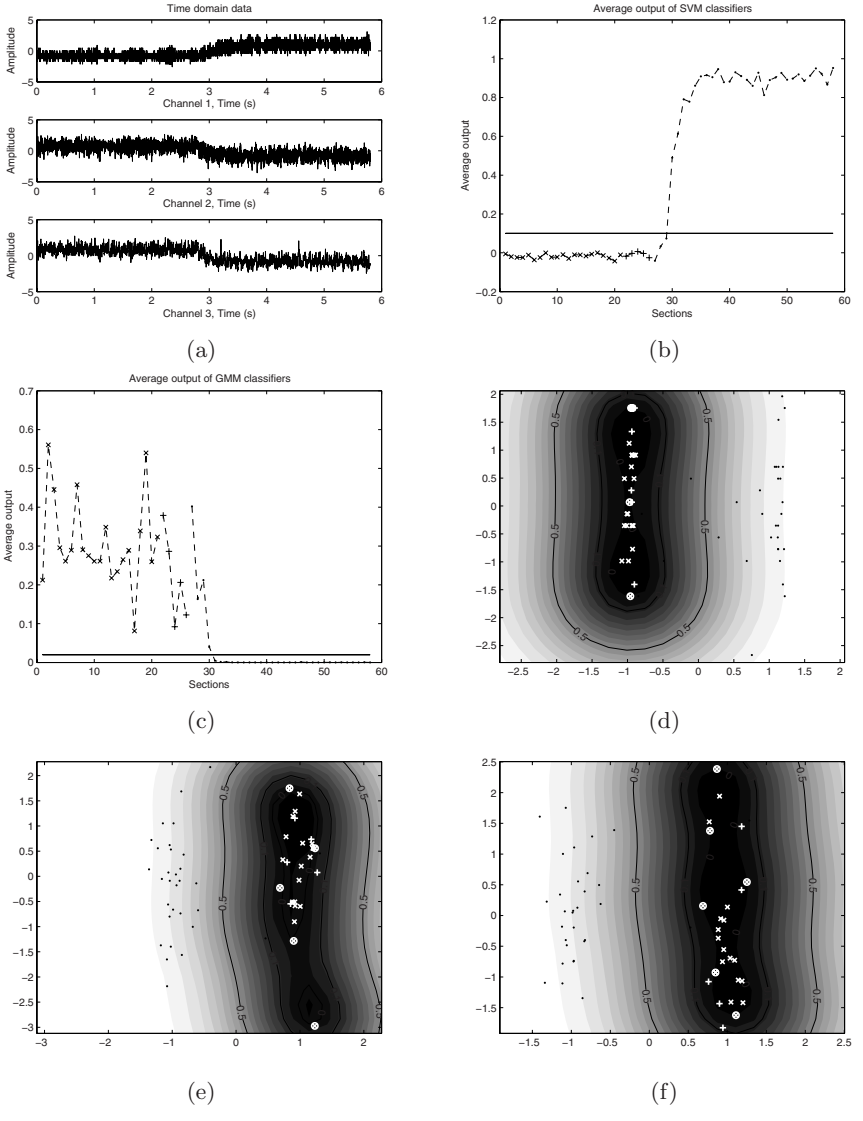


Fig. 1. Time-domain signals and SVM results for data set 1. (a) Three channel data in time-domain: gas pressure of the first channel in the upper figure, C_2 radical luminosity of the second channel in the middle figure, and global intensity of unfiltered light in the third channel in the lower figure. (b) Novelty scores generated by the combined SVM classifier. Threshold $H_s = 0.1$ is marked by the horizontal line. (c) Probabilities generated by the combined GMM classifier. Threshold $H_g = 0.02$ is marked by the horizontal line. (d)(e)(f) Contour plots of novelty scores generated by SVM classifers in the first, second and third channels respectively; support vectors are circled. In (b)-(f), stable data for training, stable data for test, transient and unstable data for test, are shown by $\{\times + \cdot\}$ respectively.

ined within this investigation. Novelty scores are assigned such that unstable combustion data are easily separated from stable combustion data.

This method of novelty detection provides good results when combined with the use of wavelet techniques for feature extraction of combustion data. This investigation emphasises the value of the SVM classification approach to on-line novelty detection in gaining early warning of possibly catastrophic combustion instabilities, such that preventative maintenance action may be taken to avoid equipment failure.

In comparison to the results obtained through application of a GMM-based approach, the SVM approach is shown to provide a robust early warning indication of unstable combustion, for both stable and unstable data.

Acknowledgement

The first author would like to thank David Clifton for his valuable discussions and comments.

References

1. Chen, Y., Zhou X.S., Huang, T.: One-class SVM for Learning in Image Retrieval. IEEE ICIP, Thessaloniki, Greece 1 (2001) 34-37
2. Daubechies, I.: Orthonormal Bases of Compactly Supported Wavelets. Comm. Pure and Appl. Math. 41 (1988) 909-996
3. Gretton, A., Desobry, F.: On-line One-class Support Vector Machines. An Application to Signal Segmentation. IEEE ICASSP, Hong-Kong, China (2003)
4. Hardoon, D.R., Manevitz, L.M.: fMRI Analysis via One-class Machine Learning Techniques. 19th IJCAI, Edinburgh, UK (2005) 1604-1605
5. Hayton, P., Tarassenko, L., Schölkopf, B., Anuzis, P.: Support Vector Novelty Detection Applied to Jet Engine Vibration Spectra. NIPS, London (2000) 946-952
6. Kittler, J., Hatef, M., Duin, R., Matas, J.: On Combining Classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(3) (1998) 226-239
7. Kuncheva, L.I.: A Theoretical Study on Six Classifier Fusion Stragies. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(2) (2002) 281-286
8. Mallat, S.: A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. IEEE Pattern Anal. and Machine Intell. 11 (1989) 674-693
9. Ng, W.B., Clough, E., Syed, K.J., Zhang Y.: The Combined Investigation of the Flame Dynamics of an Industrial Gas Turbine Combustor Using High-speed Imaging and an Ooptically Integrated Data Collection Method. Measurement Science and Tachnology 15 (2004) 2303-2309
10. Schölkopf, B., Platt, J., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the Support of a High-dimensional Distribution. Neural Computation 13 (2001) 1443-1471
11. Tax, D.M.J., Duin, R.P.W.: Data Domain Description Using Support Vectors. ESAN99, Brussels (1999) 251-256
12. Wang, L., Yin, H.: Wavelet Analysis in Novelty Detection for Combustion Image Data. 10th CACSC, Liverpool (2004) 79-82

A Levinson Predictor Based Compensatory Fuzzy Neural Network and Its Application in Crude Oil Distillation Process Modeling

Yongfeng He and Quanyi Fan

Department of Automation, Tsinghua University, Beijing 100084, China
starhe@gmail.com, fqy-dau@mail.tsinghua.edu.cn

Abstract. Levinson predictor based Compensatory fuzzy neural networks (LPCFNN), which can be trained by a back-propagation learning algorithm, is proposed as a modeling technique for crude oil distillation processes. This approach adds feedback to the input by using Levinson predictor. Simulation experiments are made by applying proposed LPCFNN on modeling for crude oil distillation process to confirm its effectiveness.

1 Introduction

In most cases, the nonlinearity and time delays are involved in the crude oil distillation processes, which increase difficulty for modeling for these processes. Multi-Layer back-propagation neural network (MLBPNN) has been shown to be able to model a nonlinear system more accurately than conventional algorithms. [1] It has been proved that a multi-layer neural network is a "universal approximator", which means that a multi-layer neural network can approximate any continuous nonlinear function by providing a sufficient number of hidden units in the hidden layers. However, conventional BPNN suffers from slow convergence to local minima and random settings of the initial values weights, which may make the neural networks have very poor mappings from inputs to outputs.

The compensatory fuzzy neural network (CFNN) is a hybrid system integrating compensatory fuzzy logic and neural network [2], [3].Because CFNN adopts compensatory fuzzy operators to perform fuzzy reasoning, the network can not only adaptively adjust input and output fuzzy membership functions, but also dynamically optimize fuzzy reasoning. This strategy makes the network more error-tolerant and makes the system more stable.

The Levinson-Durbin Recursion (LDR) describes a direct method for computing the prediction-error filter coefficients and prediction-error power by solving the augmented Wiener-Hopf equations. The algorithm is recursive in nature and makes it's suitable for online learning algorithms [4].

In this paper, a Levinson predictor based compensatory fuzzy neural network (LPCFNN) structure is proposed, in which the output is fed to the input through the Levinson predictor. Back-propagation algorithm is used to train the LPCFNN. Finally, simulation experiments are made by applying the LPCFNN on the modeling for

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 844 – 851, 2006.

a crude oil distillation process, and we also compare the results of three different kinds of network: BPNN, CFNN and LPCFNN.

2 Levinson Predictor Based Compensatory Fuzzy Neural Network

The structure of LPCFNN can be divided into two parts, one for the fuzzy-neural calculation and the other for the Levinson predictor. Fig.1 describes the basic structure of the model using the LPCFNN. Here the dynamic process refers to the steady crude oil distillation process; *x(t)* refers to the input at the *t*th step, and *y(t)* refers to the output of the dynamic process at the *t*th step; the delay unit refers to the sampling delay; *y"(t|t-d)* refers to the output of the Levinson predictor, $y_p(t)$ refers to the output of the compensatory fuzzy neural network.

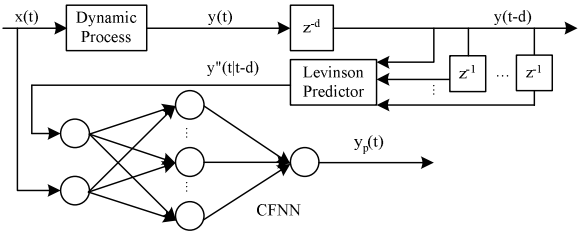


Fig. 1. This figure shows the structure of LPCFNN

Firstly, the Levinson predictor generates the estimation of *y(t)* based on the previous delayed output; then the estimation of *y(t)*, i.e. *y"(t|t-d)* is added to the input signals of the compensatory fuzzy neural network. The compensatory fuzzy neural network then calculates the output $y_p(t)$ according to its input *x* and *y"(t|t-d)*. After we have got the estimation of *y(t),* we then can train the CFNN using the learning algorithm such as back-propagation.

2.1 Structure and Learning Algorithm of CFNN

The structure of CFNN shown in Fig.2 comprises *n* input variables, *m* term nodes for each input variable, 2*l* rule nodes, *l* compensatory nodes and *p* output nodes. This CFNN consists of five layers: input layer, membership layer, rules layer, compensatory layer and output layer. Using u_i^k and o_i^k to denote the input and output of the *i*th node in the *k*th node respectively, the signal propagation and the operation functions are introduced as follows [1], [2], [5].

The input layer accepts input variables. Its nodes transmit input values to the next layer. Nodes in the membership represent the terms of respective linguistic variables. Each node performs a Gaussian membership function.

$$u_{ij}^2 = \exp(-((o_i^1 - a_{ij}) / \sigma_{ij})^2) \tag{1}$$

where we have $u_i^1 = o_i^1$, $i = 1,2, \ldots, n$, $j=1,2,\ldots,m$; u_{ij}^2 is the input for the jth term of the ith input variable; a_{ij} and σ_{ij} are the center and the width of the membership.

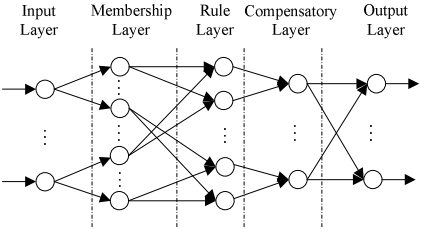


Fig. 2. Architecture of the Compensatory Fuzzy Neural Network

The rule layer forms the fuzzy rule base and realizes the fuzzy inference. This layer has l pairs of node. Each pair is corresponding to a fuzzy rule. Links before each node represent the preconditions of the corresponding rules, and the output of the node represents a subset of the output fuzzy set. Here the IF-THEN fuzzy rules are used and can be described as follows.

$$k\text{th Rule : IF } x_1 \text{ is } A_1^k, \ldots, x_n \text{ is } A_n^k \text{ THEN } y_1 \text{ is } B_1^k, \ldots, y_p \text{ is } B_p^k. \tag{2}$$

where $x = [x_1, x_2, ..., x_n]$ is the input vector of the network. A_i^k is the term of the ith in the kth rule; B_j^k is the term of the jth output in the kth rule. The membership functions of A_i^k and B_j^k are defined as equation (1).

Then one node of the kth node pair of the rule layer performs pessimistic operation and the other one performs optimistic operation as follows.

$$u_{2k}^3 = \prod_{i=1}^{n} o_{ik_i}^2 \quad , u_{2k+1}^3 = [\prod_{i=1}^{n} o_{k_i}^2]^{1/n} , o_j^3 = u_j^3 \tag{3}$$

where $o_{ik_i}^2$ denotes the membership function of x_i to A_i^k; k_i indicates the k_i th term node for x_i ($k_i = 1, 2, ..., m$); u_{2k}^3 is the input of the kth node that performs pessimistic operation; u_{2k+1}^3 is the input of the kth node that performs optimistic operation; $k = 1, 2, \ldots, l$.

The compensatory layer with l nodes performs compensatory operation as follows.

$$o_k^4 = (o_{2k}^3)^{1-\gamma^k} (o_{2k+1}^3)^{\gamma^k} \tag{4}$$

where $\gamma^k \in [0, 1]$ is the compensatory degree of rule, $k = 1, 2, \ldots, l$.

The output layer has p nodes and realizes the defuzzification operation as follows.

$$u_j^5 = \sum_{q=1}^{l} w_{sq} o_q^4 \ , o_s^5 = u_s^4 / \sum_{q=1}^{l} o_q^4 \tag{5}$$

where $s = 1, 2, \ldots, p$, $q = 1, 2, \ldots, l$, w_{sq} is the weight that represents the output action strength of the sth output associated with the qth rule.

2.2 Levinson Predictor

For a nonlinear dynamic system with pure delays, we can describe as follows:

$$y(t) = f(y(t-d), ..., y(t-d-n+1), x(t-1), ..., x(t-m)) \tag{6}$$

where f is a nonlinear function, d is the delay steps, n is the delay order of the output, m is the delay order of the input. We can use the following equation to approximate to this model:

$$y''(t|t-d) = g_L(y(t-d), y(t-d-1), \cdots, y(t-d-n+1)) \tag{7}$$

$$y_p(t) = f(y''(t|t-d), x(t-1), \cdots, x(t-m)) \tag{8}$$

where the Levinson predictor uses g_L to generate the prediction of y, which is a linear function about $y(t-d)$, $y(t-d-1)$, …, $y(t-d-n+1)$. $y''(t|t-d)$ is the output of the predictor and $y_f(t)$ is the output of the CFNN. For $d = 1$, we have:

$$y''(t \mid t-1) = -\sum_{i=1}^{n} a_i y(t-i) \tag{9}$$

Then we can approximately calculate the $y''(t|t-d)$ as:

$$y''(t|t-d) = -a_1 y''(t-1|t-d) - \cdots - a_{d-1} y''(t-d+1|t-d) - a_d y(t-d) - \cdots - a_n y(t-n) \tag{10}$$

a_i is calculated as the following recursive steps described as follows [6].

$$\gamma_k = \frac{-1}{P_{k-1}} \sum_{m=0}^{k} a_{k-1}(m) r_x(k-m) \tag{11}$$

$$P_k = P_{k-1}(1-\gamma_k^2) \tag{12}$$

$$a_k(m) = a_{k-1}(m) + \gamma_k a_{k-1}(k-m) \tag{13}$$

where $k = 1, 2 \ldots M$; $m = 1, 2 \ldots k$; $r_x(i)$ is the i-step autocorrelation of the input; γ_k is the reflection coefficient. And we initialize the algorithm with $P_0 = r_x(0)$, $a_0(0) = 1$, $a_0(1) = 1$. Careful observation of above equations will result in $a_k(m) = 0$ $(m > k)$, $a_k(0) = 1$, $a_k(k) = \gamma_k$.

3 Application in the Modeling of Crude Oil Distillation Processes

3.1 Construction of the Model

Crude oil distillation is in the first place of petroleum refining. The light oils yield ratio of the crude oil distillation process is one of the most important factors in planning for the production. Here the prediction for light oils yield ratio under the circumstances of atmospheric pressure distillation is considered. Light oils yield ratio of the

atmospheric distillation column is the ratio of the total amount of products from the top to the third side-stream of the distillation column to the feeding crude oil:

$$\eta = \sum_k P_k / \sum_j F_j \tag{14}$$

where P_k is the yield of the kth product, F_j is the jth feedstock and η is the light oil yield ratio which is the output of our model. The inputs are the controllable variables in the distributed control system installed in the atmospheric distillation column.

However, some variables are coupled to each other, which are too redundant to be used. If we treat them all these variables as the input variables of the LPCFNN, the network will be too complicated and additional error is prone to be introduced. Therefore, variables selection is a key issue in the prediction. Many approaches, such as PLS and PCA [7], can be used to select appropriate variables. Finally we select five most significant variables from forty-four variables. After analyzing the relationship between the five variables, we get five decision variables described in table 1. Then we define a five-input-one-output LPCFNN with its input variables as the five decision variables and its output as the light oil yield ratio.

Table 1. Decision variables as the input for the LPCFNN

Input variables	meaning
x_1	Top circulating reflux / Returned top reflux
x_2	Return temperature of second intermediate circulating reflux
x_3	Temperature of second side-stream draws
x_4	Temperature of third side-stream draws
x_5	Output of the Levinson Predictor

To eliminate the influence caused by the dimension and change range of technological factors, the raw data are standardized so that input variables will have zero means and unity standard deviation, and the output variable will range between 0 and 1. According to the fuzzy partition method of input space, the input data of the network are carefully analyzed. Thus we partition the standardized input vector spaces as table 2 shows.

Table 2. The partitioned spaces for input vector

Input variables	Range	Partition
x_1	(-2, 2)	(-2, -0.5) (-0.5, 1) (1, 2)
x_2	(-2, 2.5)	(-2, -0.5) (-0.5, 1) (1, 2.5)
x_3	(-4.5, 3)	(-4.5, -2) (-2, 0.5) (0.5, 3)
x_4	(-6, 8)	(-6, -3) (-3, 0) (3, 6) (6, 9)
x_5	(-4, 4)	(-4, -2) (-2, 0) (0, 2) (2, 4)

Input and output fuzzy membership functions of A_1^k, A_2^k, A_3^k, A_4^k, A_5^k and B^k in the IF-THEN rules described in section 2.1 are then constructed as follows($k = 1,2 \ldots 540$).

$$\mu_{A_1^k}(x_1) = \exp(-(x_1 + 1.25 - 1.5(k \bmod 3))^2) \tag{15}$$

$$\mu_{A_2^k}(x_2) = \exp(-(x_2 + 1.25 - 1.5([k/3] \bmod 3))^2) \tag{16}$$

$$\mu_{A_3^k}(x_3) = \exp(-(x_3 + 3.25 - 2.5([k/9] \bmod 3))^2) \tag{17}$$

$$\mu_{A_4^k}(x_4) = \exp(-(x_4 + 4.5 - 3([k/27]) \bmod 5)^2) \tag{18}$$

$$\mu_{A_5^k}(x_5) = \exp(-(x_5 + 3 - 2([k/135]) \bmod 4)^2) \tag{19}$$

$$\mu_B(y^p) = \exp(-(y^p - b^k)^2) \tag{20}$$

The center of output membership function b^k can be initialized as

$$\text{if } N_k \neq 0 \text{ then } b^k = \frac{1}{N_k}\sum_{i_k=1}^{N_k} y^{M_k(i_k)} \quad \text{, else } b^k = 0.5 \tag{21}$$

where N_k is the total number of data points in the *k*th subspace partitioned as table 2, and $y^{M_k(i_k)}$ is the corresponding output.

To construct the Levinson predictor, we should identify its order and calculate the coefficient a_i according to the Levinson algorithm described by equation (11)-(13). We get the relationship between the mean square of the output error and the order of the predictor by calculating the order from 1 to 20, as Fig3 shows.

As Fig.3 shows, when the order is larger than 3, we can see the mean square error decreases very slowly. Thus we choose 2 as the order of the Levinson predictor. Thus according to equation (11)-(13), we can get the predicting equation as the following:

$$y''(t \mid t-1) = 0.4176\, y(t-1) + 0.0707\, y(t-2) \tag{22}$$

where *y"(t|t-1)* is the estimation of current output based on last outputs.

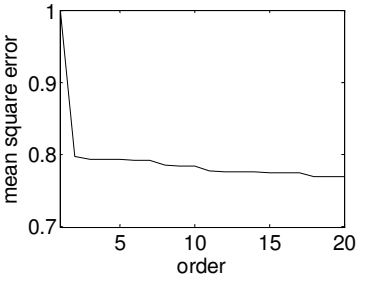


Fig. 3. This figure shows the relationship between the mean square error and the order of the Levinson predictor

The constructed LPCFNN is applied to predict light oil yield ratio. For the CFNN, we use (15)-(20) as the membership function of the five input variables and the single output, and we construct the CFNN part as Fig.1.shows. For the Levinson predictor, we use the model described by equation (22).

3.2 Simulation Results

In the simulation experiments, 1200 consecutive data points are collected, of which the first 800 data points are used to train the network, and the rest 400 data points are used for the simulation. We employ a three-layer BPNN with 20 hidden nodes, a CFNN without feedback and a LPCFNN to get the comparison of the performances of these networks. Table 3 shows the performance of these networks.

The input variables are calculated according to table 2, and then centered and normalized. In the Fig.4, it is obvious that the converging speed of LPCFNN is almost the same as BPNN and CFNN, but the mean square error of LPCFNN is smaller than

Table 3. This table details the performance of the simulation results

	BPNN	CFNN	LPCFNN
Mean of Error	0.7919	0.5481	0.4665
Standard Deviation of Error	0.9029	0.5309	0.4016
Error Distribution (<1%)	76.50%	87.00%	90.75%

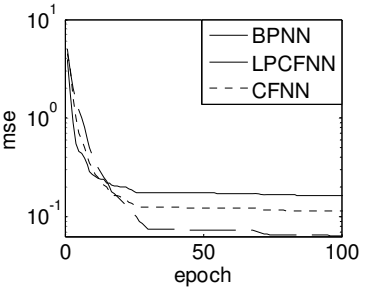


Fig. 4. This figure shows the training curve of BPNN, CFNN and LPCFNN

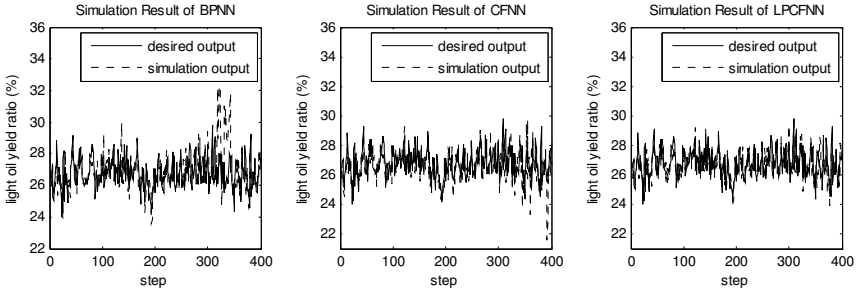


Fig. 5. This figure shows the simulation results of BPNN, CFNN and LPCFNN

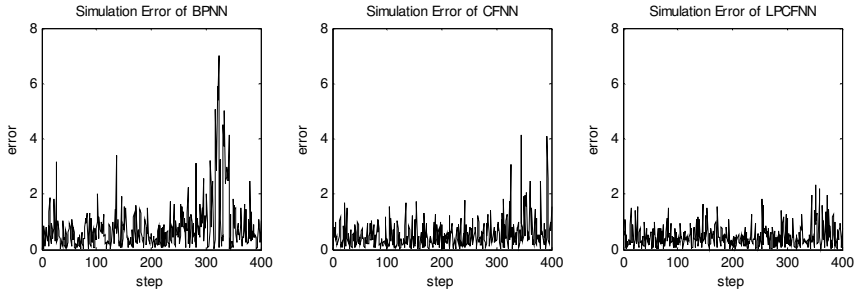


Fig. 6. This figure shows the simulation errors of BPNN, CFNN and LPCFNN.

the others. Simulation results are shown in Fig.5 and simulation errors are shown in Fig.6. From the simulation results and errors shown in Fig.5, Fig.6 and table 3, it is obvious that the performance of the proposed LPCFNN is better than others. LPCFNN achieves good effects in the prediction for the light oil yield ratio, and is superior to BPNN and CFNN in training epochs, prediction results, and the global error curve is more stable.

4 Conclusion

This paper proposed a Levinson predictor based compensatory fuzzy neural network (LPCFNN) using IF-THEN fuzzy rules. The Levinson predictor gives an optimal estimation of the current output and then the estimation is added to the input of the compensatory fuzzy neural network (CFNN). Thus this model is a kind of dynamic feedback system, as the simulation results show, this kind of dynamic model has more advantages in training epochs than BPNN and CFNN, and is more accurate than the other two kinds of neural networks.

References

1. Seker, H., Evans, D.H., Aydin, N., Yazgan, E.: Compensatory Fuzzy Neural Networks-Based Intelligent Detection of Abnormal Neonatal Cerebral Doppler Ultrasound Waveforms. IEEE Transactions on Information Technology in Biomedicine 5(3) (2001) 187 - 194
2. Zhang, Y.-Q., Kandel, A.: Compensatory Neurofuzzy Systems with Fast Learning Algorithms. IEEE Transactions on Neural Networks 9(1) (1998) 83 - 105
3. Sun, W., Wang, Y.: A Recurrent Neural Network Based Adaptive Control and Its Application on Robotic Tracking Control. Neural Information Processing 5(1) (2004) 19 - 26
4. Levinson, N.: The Wiener RMS (Root-Mean-Square) Error Criterion in Filter Design and Prediction. Journal of Mathematics and Physics 25(1) (1947) 261 - 278
5. Yu, W., Li, X.: Fuzzy Identification Using Fuzzy Neural Networks with Stable Learning Algorithms. IEEE Transactions on Fuzzy Systems 12(3) (2004) 411 - 420
6. Zhang, X.D.: Modern Signal Processing. Tsinghua University Press and Springer Press, Beijing (2002)
7. Haykin, S.: Neural Networks: A Comprehensive Foundation. 2nd edn. Prentice Hall, Upper Saddle River, New York (1999)

Laminar Cooling Process Model Development Using RBF Networks

Minghao Tan[1], Xuejun Zong[2], Heng Yue[3], Jinxiang Pian[3], and Tianyou Chai[3]

[1] School of Information Science and Engineering, Shenyang University of Technology, Shenyang, Liaoning 110023, China
mhtan@sut.edu.cn
[2] College of Information Engineering, Shenyang Institute of Chemical Technology, Shenyang, Liaoning 110142, China
xuejun8720@sohu.com
[3] Research Center of Automation, Northeastern University, Shenyang, Liaoning 110004, China
tychai@mail.neu.edu.cn

Abstract. Due to the complex nature (e.g., highly nonlinear, time varying, and spatially varying) of the laminar cooling process, accurate mathematical modeling of the process is difficult. This paper developed a hybrid model of the laminar cooling process by integrating Radial Basis Function (RBF) networks into the first principles dynamical model. The heat transfer coefficients of water cooling in the dynamical model were found by RBF networks. The developed model is capable of predicting the through-thickness temperature evolutions of the moving strip during the laminar cooling process. Experimental studies using real data from a hot strip mill show the superiority of the proposed model.

1 Introduction

Accurate modeling of the laminar cooling process is difficult because of its complex nature [1,2]. Various approaches have been proposed for modeling the laminar cooling process. Several authors used statistical methods and simplified or empirical heat transfer equations to model the laminar cooling process [2,3,4]. The through-thickness heat conduction of the strip was neglected in [5], where the laminar cooling process was modeled as a first-order time delay, while the real process is a distributed parameter system described by nonlinear partial differential equations. Ref. [6] developed a numerical model for the laminar cooling process. But it is difficult to determine the model parameters and the model can only be used for offline purposes.

This paper takes an intelligent approach toward modeling the laminar cooling process. The model structure is developed from first principles analysis, while the heat transfer coefficients of water cooling are determined by RBF networks. This paper is organized as follows. The hybrid modeling strategy of the laminar cooling process is discussed in Section 2. The results of an industrial case study are presented in Section 3 and conclusions are given in Section 4.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 852 – 857, 2006.

2 Hybrid Modeling Strategy of the Laminar Cooling Process

A detailed description of the laminar cooling process can be found in [7]. The boundary conditions of the laminar cooling process include the steel grade G_r, the strip gauge d, the strip speed v, the strip acceleration a, the strip length L, the entry temperature T_e, the water temperature T_w, and the ambient temperature T_a. The inputs of the process include the header flow rate q, the first activated top header H_t, the first activated bottom header H_b, the number of activated headers H, and the spraying pattern π. The output of the laminar cooling process is the strip temperature.

The proposed hybrid modeling strategy for the laminar cooling process is shown in Fig. 2, where T_s (T_0 or T_M) is the strip surface temperature, H_s is the first activated header, H_c is the current header, α_w is the heat transfer coefficient of water cooling, α_a is the heat transfer coefficient of air cooling, λ_j is the thermal conductivity, and a_j is the thermal diffusivity.

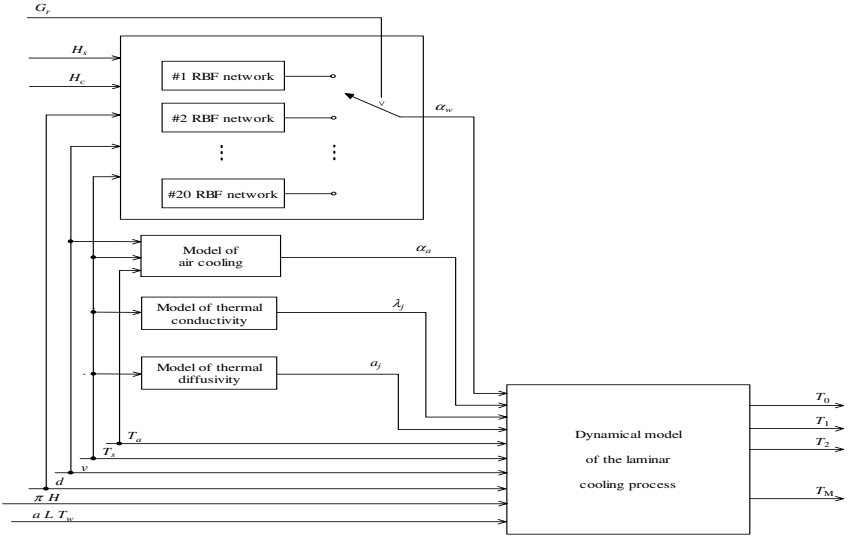


Fig. 1. Schematic of the hybrid modeling strategy for the laminar cooling process

The heat transfer coefficients of water cooling are found by one of twenty RBF networks, one for each existing steel grade. The RBF network to be used is selected by the steel grade G_r. The surface temperature, gauge and speed of the strip, the first activated header, and the current header are fed to the selected RBF network to find the heat transfer coefficient of water cooling α_w (α_0 or α_M) under the specified header H_c. By recursively solving the dynamical model the through-thickness temperature evolution and the coiling temperatures of the specified strip segment can be predicted.

2.1 Dynamical Model of the Laminar Cooling Process

The through-thickness temperature evolution of any strip segment is described by [7]

$$\begin{aligned}&(1+a_0\frac{M^2\Delta\Gamma}{d^2}+a_0\frac{M\Delta\Gamma\alpha_0}{d\lambda_0})T_0(t+1)-a_0\frac{M^2\Delta\Gamma}{d^2}T_1(t+1)=\\&(1-a_0\frac{M^2\Delta\Gamma}{d^2}-\frac{a_0M\Delta\Gamma\alpha_0}{d\lambda_0})T_0(t)+a_0\frac{M^2\Delta\Gamma}{d^2}T_1(t)+2\frac{a_0M\Delta\Gamma\alpha_0}{d\lambda_0}T_w\end{aligned} \quad (1)$$

$$\begin{aligned}&(2+2\frac{a_jM^2\Delta\Gamma}{d^2})T_j(t+1)-\frac{a_jM^2\Delta\Gamma}{d^2}T_{j-1}(t+1)-\frac{a_jM^2\Delta\Gamma}{d^2}T_{j+1}(t+1)\\&=\frac{a_jM^2\Delta\Gamma}{d^2}T_{j-1}(t)+(2-2\frac{a_jM^2\Delta\Gamma}{d^2})T_j(t)+\frac{a_jM^2\Delta\Gamma}{d^2})T_{j+1}(t)\end{aligned} \quad (2)$$

$$\begin{aligned}&(1+a_M\frac{M^2\Delta\Gamma}{d^2}+a_M\frac{\alpha_MM\Delta\Gamma}{d\lambda_M})T_M(t+1)-a_M\frac{M^2\Delta\Gamma}{d^2}T_{M-1}(t+1)=\\&a_M\frac{M^2\Delta\Gamma}{d^2}T_{M-1}(t)+(1-a_M\frac{M^2\Delta\Gamma}{d^2}-a_M\frac{M\Delta\Gamma\alpha_M}{d\lambda_M})T_M(t)+2a_M\frac{M\Delta\Gamma\alpha_M}{d\lambda_M}T_w\end{aligned} \quad (3)$$

where j: the j^{th} through-thickness layer (j=0, 1, … M), T: the strip temperature, t: the t^{th} time step, $\Delta\Gamma$: time step size, a_j: thermal diffusivity at layer j; λ_0, λ_M: thermal conductivities; α_0, α_M: heat transfer coefficients at top and bottom surface.

For the case the current header H_c is deactivated, the heat transfer coefficient at the top surface, α_0, is computed as follows [7,8]

$$\alpha_0=\bar{\sigma}\times\bar{\varepsilon}\times\frac{(T_0^4-T_a^4)}{T_0-T_a}+5\times v^{0.9} \quad (4)$$

The heat transfer coefficient of air cooling at the bottom surface is calculated by [7]

$$\alpha_M=\bar{\sigma}\times\bar{\varepsilon}\times\frac{(T_0^4-T_a^4)}{T_0-T_a} \quad (5)$$

where $\bar{\sigma}$ is the Stefan-Boltzmann constant, $\bar{\varepsilon}$ = 0.82 is the emissivity.

The thermal conductivities at the top and bottom surface λ_j (j=0, M) are found by [7]

$$\lambda_j = 56.43 - 0.0363\,T_j \qquad (j=0, M)\,. \quad (6)$$

The calculation of the thermal diffusivity is omitted due to limited space [7]. This paper uses RBF networks to determine α_w according to the operating conditions.

2.2 Setup of the Radial Basis Function Network

The network input vector $X\in R^5$ is defined as $X=[d,T_s,v,H_s,H_c]^T$. The network output vector $Y\in R$ is the heat transfer coefficient of water cooling α_w (α_0 or α_M). Then the mapping $R^5\rightarrow R$ of the RBF network is given by

$$Y(X)=Z^T(X)W=w_0+\sum_{i=1}^{m}w_i\phi_i(\|X-C_i\|) \quad (7)$$

where m is the number of basis functions, ϕ_i is the ith radial basis function ($(i=1,\cdots,m)$, $Z=[1,z_1,z_2,...,z_m]^T$ is the output vector of the radial basis functions, $\|.\|$ denotes the Euclidean distance, $C_i \in R^5$ is known as the RBF center, and $W=[w_0,w_1,w_2,...,w_m]^T$ is the weighting vector. ϕ_i is chosen as the Gaussian function, so the output of the hidden layer is computed as

$$z_{ip}=\phi_i\left(\left\|X_p-C_i\right\|\right)=\exp\left[-\sum_{k=1}^{5}\frac{(x_{kp}-c_{ik})^2}{2\sigma_i^2}\right], \tag{8}$$

where z_{ip} is the output of ϕ_i for the p^{th} sampled input X_p, and σ_i is the width of ϕ_i.

2.3 Learning of the RBF Network

The RBF network is learned in two stages. First the network centers are determined by unsupervised learning. Then the output weights are tuned using the RLS algorithm [9, 10].

(1). Initialization: let $m=1$, choose initial values of the RBF centers $C_i(0)$, the learning rate $\beta(0)$ $(0<\beta<1)$, and the error tolerance ε.

(2). Clustering: For every sample X_p (p=1,…, N) find the best-matching center C_k for the input sample X_p by using a minimum-distance Euclidean criterion:

$$b_i(n)=\left\|X_p-C_i(n)\right\|,\ 1\le i\le m \tag{9}$$

$$k=\arg\min_{1\le i\le m}\{b_i(n)\} \tag{10}$$

Then adjust the RBF centers and the learning rate with the following update rules

$$C_i(n+1)=C_i(n),1\le i\le n,i\ne k \tag{11}$$

$$C_k(n+1)=C_k(n)+\beta(n)[X_p-C_k(n)] \tag{12}$$

$$\beta(n+1)=\beta(n)/\sqrt{1+\text{int}(n/m)} \tag{13}$$

(3). If $\sum_{i=1}^{m}\left\|C_i(n+1)-C_i(n)\right\|<\varepsilon$, go to the next step, otherwise return to step (2).

(4). Calculate the width of the radial basis function ϕ_i, $\sigma_i=\frac{D}{\sqrt{2m}}$, where D is the maximum distance between two RBF centers.

(5). Find the weighting vector W using the recursive least squares algorithm [9].

(6). Calculate the accumulated square error of the RBF network predictions

$$E=\frac{1}{2}\sum_{p=1}^{N}(Y_p-Z^TW)^2\ , \tag{14}$$

where Y_p is the sampled output for the p^{th} sampled input X_p. If $E<\varepsilon$, then stop learning, else let $m=m+1$ and return to step (2).

3 Industrial Case Study

The boundary conditions for the experiment are shown in Table 1. The training data for α_w are obtained from a combination of real measurements, first principles analysis, empirical data and trial-and-error. Table 2 lists the calculated heat transfer coefficients at the first activated headers.

Table 1. Boundary conditions for the experiment

G_r	d(mm)	L(m)	T_w(℃)	H_t	H_b
320	12	231	29	17	19

Table 2. Experiment results and comparison

$\alpha_0(17)$	$\alpha_M(19)$	Measurements ±10℃		Measurements ±30℃	
		This paper	Ref. [11]	This paper	Ref. [11]
3270	3201	53	11	60	40

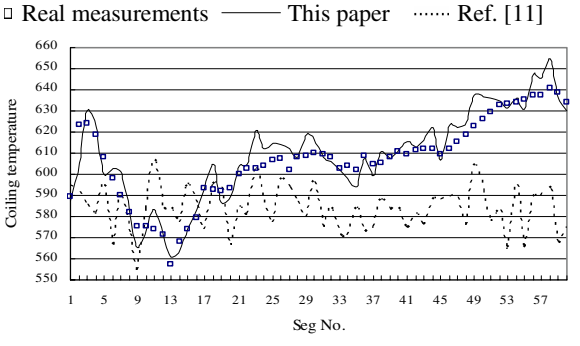


Fig. 2. Comparison of coiling temperature predictions

The model predictions of this paper, Ref [11], and the real measurements are plotted in comparison in Fig. 2. It is evident that the model developed in this paper is very good at tracking the evolution of the strip temperature and capable of much better accuracy than the model in [11].

4 Conclusions

The lack of understanding of the heat exchange during the laminar cooling process has made it very difficult to develop an accurate physical model. This paper has introduced a novel hybrid approach to modeling the laminar cooling process using first principles analysis and RBF networks. Experimental studies based on industrial data have demonstrated that the hybrid modeling approach contributes significantly to

improved model accuracy. The temperature profiles provided by the proposed model provide useful insights into the heat transfer of the laminar cooling process. The results in this paper can be generalized to a wide range of similar processes.

Acknowledgment

This work was partly supported by the Ph.D. Startup Fund of Shenyang University of Technology, the Funds for Creative Research Groups of China under Grant No.60521003, and the China National Key Basic Research and Development Program under Grant No.2002CB312201.

References

1. Chai, T.Y., Tan, M.H., Chen, X.Y., Li, H.X.: Intelligent Optimization Control for Laminar Cooling. In: Camacho, B., Puente, D. (eds.): Proc. of the 15th IFAC World Congress. Elsevier, Amsterdam (2003) 691-696
2. Groch, A.G., Gubemat, R., Birstein, E.R.: Automatic Control of Laminar Flow Cooling in Continuous and Reversing Hot Strip Mills. Iron and Steel Engineer 67(9) (1990) 16-20
3. Leitholf, M.D., Dahm, J.R.: Model Reference Control of Runout Table Cooling at LTV. Iron and Steel Engineer 66(8) (1989) 31-35
4. Moffat, R.W.: Computer Control of Hot Strip Coiling Temperature with Variable Flow Laminar Spray. Iron and Steel Engineer 62(11) (1985) 21-28
5. Ditzhuijzen, V.G.: The Controlled Cooling of Hot Rolled Strip: A Combination of Physical Modeling, Control Problems and Practical Adaptation, IEEE Trans. Aut. Contr. 38(7) (1993) 1060-1065
6. Evans, J.F., Roebuck, I.D., Howard, R.W.: Numerical Modeling of Hot Strip Mill Runout Table Cooling. Iron and Steel Engineer 70(1) (1993) 50-55
7. Tan, M.H.: Case-based Modeling and Control of the Laminar Cooling Process on the Runout Table. Ph.D. Dissertation. Northeastern University, Shenyang (2003)
8. Thomas, G.: A Combined Feedforward-Feedback Computer System for Hot Strip Mill. C.R.M. Metallurgical Rep. 52(1) (1978) 17-23
9. Haykin, S.: Neural Networks: A Comprehensive Foundation. 2nd edn. Prentice Hall, Upper Saddle River, New Jersey (1999)
10. Moody, J., Darken, C.: Fast Learning in Networks of Locally-Tuned Processing Units. Neural Computat. 1(2) (1989) 281-294
11. Shan, X.: Transformation and Development of the Cooling Control System of the 2050mm Baosteel Hot Strip Mill. In: Ren, D. (eds.): Development of Science and Technology in Metallurgy. Metallurgical Industry Press, Hangzhou China (1999) 1-4

Hybrid Intelligent Control Strategy of the Laminar Cooling Process

Minghao Tan[1], Shujiang Li[1], and Tianyou Chai[2]

[1] School of Information Science and Engineering, Shenyang University of Technology, Shenyang, Liaoning 110023, China
{mhtan, lisj}@sut.edu.cn
[2] Research Center of Automation, Northeastern University, Shenyang, Liaoning 110004, China
tychai@mail.neu.edu.cn

Abstract. Performance of controlled laminar cooling is usually poor because of the difficulty in continuous online temperature measurement and the complex nature of the laminar cooling process (e.g., highly nonlinear, time varying). This paper developed a hybrid control strategy for the laminar cooling process that integrates Radial Basis Function (RBF) networks and Case-Based Reasoning (CBR). The spraying pattern and the first activated headers are found by a case-based reasoner, while the number of activated headers is calculated in real time by RBF networks. Experimental studies using production data from a hot strip mill show the superior performance of the proposed control strategy.

1 Introduction

In a hot strip mill, the laminar cooling process is controlled to ensure good mechanical properties for the strip. However due to the difficulty of continuous temperature measurement, the complex process nature and frequent variations of operating conditions, the control performance is very poor and human intervention is often required [1,2].

Many of the controllers that have been reported use either empirical or statistical approaches that rely on simplified heat transfer equations [2,3,4]. Ditzhuijzen has proposed a minimum settling time feedback controller by modeling the process as simply a first-order time delay, while the real laminar cooling process is a distributed parameter system described by nonlinear partial differential equations [5]. The feedforward controllers proposed in [4,6,7] use an approximate model of the laminar cooling process to predict the strip temperature and estimate the amount of water needed. All the above controllers are only effective for a specific laminar cooling process, and must be revised if they have to be applied to a different laminar cooling process.

This paper introduces an intelligent control approach for the laminar cooling process implemented by a combination of case-based reasoning (CBR) [8,9] and radial basis function (RBF) networks [10]. The paper is organized as follows. The hybrid control strategy of the laminar cooling process, including the case-based reasoner, and the RBF networks-based real time controller is discussed in section 2. The experimental results are presented in section 3 and conclusions are given in section 4.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 858 – 863, 2006.

2 Hybrid Intelligent Control Strategy

Detailed description of the laminar cooling process is omitted in this paper due to limited space [11]. The objective of the laminar cooling process is to ensure uniform temperature and cooling rate for the whole strip. This is achieved by adjusting the number and distribution of the activated headers. In particular the cooling rate is controlled through the use of different spraying patterns [1]. The inputs of the laminar cooling process include the number of the activated headers H, the first activated top header H_t, the first activated bottom header H_b, and the spraying pattern π. The output of the laminar cooling process is the strip temperature.

The proposed hybrid intelligent control strategy for the laminar cooling process is shown in Fig. 2, where $T_0 \ldots T_M$ are the through-thickness temperatures of the strip, G_r is the steel grade, d is the strip gauge, v is the strip speed, a is the strip acceleration, L is the strip length, T_e is the entry temperature, T_c is the coiling temperature setpoint, T_w is the water temperature, and T_a is the ambient temperature.

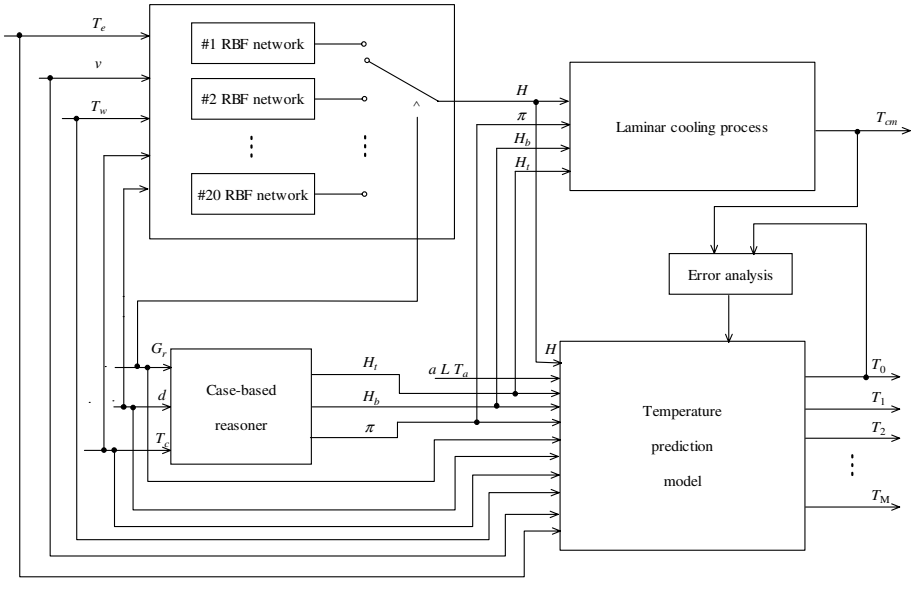


Fig. 1. Schematic of the hybrid control strategy for the laminar cooling process

The first activated headers H_t, H_b and the spraying pattern π are found by the case-based reasoner before the strip is cooled. The most similar case is retrieved from the case base according to the steel grade, the strip gauge and the coiling temperature setpoint. Then the solution of the retrieved case is reused for the control of the current strip. The number of activated headers H is recalculated in every sampling cycle by one of twenty RBF networks, to be selected by the steel grade. The temperature prediction model is described in detail in [11].

2.1 Case-Based Reasoner

The case structure is shown in Table 1.

Table 1. Case Structure

Case descriptors			Case solutions		
f_1	f_2	f_3	s_1	s_2	s_3
G_r	T_c	d	H_t	H_b	π

Case Retrieval and Reuse. We denote the operating condition of the strip as C_{in}. The descriptors of the stored case C_i^p (i=1,…n) are $F_i^p = (f_{i,1}^p, f_{i,2}^p, f_{i,3}^p)$, where n is the number of cases in the case base. The solutions of the case C_i^P are defined as $S_i^p = (s_{i,1}^p, s_{i,2}^p, s_{i,3}^p)$. The similarity between C_{in} and the stored case C_i^p (i=1,…n) is found as follows [8]

$$\mathrm{SIM}_i = \mathrm{SIM}(C_{in}, C_i^p) = \sum_{j=1}^{3} \omega_j \times \mathrm{sim}_j(f_j, f_{i,j}^p) \quad . \tag{1}$$

where ω_j's (j=1,2,3) are the weights on the respective case descriptors and defined as

$$\omega_1=0.4,\ \omega_2=0.28,\ \omega_3=0.32 \tag{2}$$

The local similarity metrics are defined as follows

$$\mathrm{sim}_1 = \mathrm{sim}(f_1, f_{i,1}^p) = \begin{cases} 1 & f_1 = f_{i,1}^p \\ 0.9 & 1 < \left|f_1 - f_{i,1}^p\right| \le 3 \\ 0.7 & 3 < \left|f_1 - f_{i,1}^p\right| \le 10 \\ 0.3 & 10 < \left|f_1 - f_{i,1}^p\right| \le 25 \\ 0 & \left|f_1 - f_{i,1}^p\right| > 25 \end{cases}, (i=1,\ldots n) \tag{3}$$

$$\mathrm{sim}_2 = \mathrm{sim}(f_2, f_{i,2}^p) = \begin{cases} 1 & \left|f_2 - f_{i,2}^p\right| < 2 \\ 0.95 & 2 \le \left|f_2 - f_{i,2}^p\right| < 5 \\ 0.9 & 5 \le \left|f_2 - f_{i,2}^p\right| < 10 \\ 0.8 & 10 \le \left|f_2 - f_{i,2}^p\right| < 20 \\ 0.6 & 20 \le \left|f_2 - f_{i,2}^p\right| < 30 \\ 0.2 & 30 \le \left|f_2 - f_{i,2}^p\right| < 40 \\ 0 & \left|f_2 - f_{i,2}^p\right| > 40 \end{cases}, (i=1,\ldots n) \tag{4}$$

$$\mathrm{sim}_3 = \mathrm{sim}(f_3, f_{i,3}^p) = \begin{cases} 1 & f_3 = f_{i,3}^p \\ 0.9 & 0 < \left|f_3 - f_{i,3}^p\right| \le 0.3 \\ 0.8 & 0.3 < \left|f_3 - f_{i,3}^p\right| \le 1 \\ 0.7 & 1 < \left|f_3 - f_{i,3}^p\right| \le 2 \\ 0.5 & 2 < \left|f_3 - f_{i,3}^p\right| \le 5 \\ 0.3 & 5 < \left|f_3 - f_{i,3}^p\right| \le 10 \\ 0 & \left|f_3 - f_{i,3}^p\right| > 10 \end{cases}, (i=1,\ldots n) \tag{5}$$

The stored case with the greatest similarity to the operating condition C_{in} is retrieved as the matching case. The first activated headers and the spraying pattern of the retrieved case are reused, executed in the laminar cooling process and also sent to the temperature prediction model.

2.2 RBF Network Controller

The network input vector $X \in R^5$ is defined as $X = [d, T_c, v, T_e, T_w]^T$. The network output $Y \in R$ is the number of activated headers *H*. The network mapping $R^5 \to R$ of the RBF network is defined as

$$Y(X) = Z^T(X)W = w_0 + \sum_{i=1}^{m} w_i \phi_i(\|X - C_i\|) \cdot \tag{6}$$

where m is the number of basis functions, ϕ_i is the *i*th radial basis function ($i = [1, \cdots, m]$), $Z = [1, z_1, z_2, ..., z_m]^T$ is the output vector of the radial basis functions, ||.|| denotes the Euclidean distance, $C_i \in R^5$ is known as the RBF center, and $W = [w_0, w_1, w_2, ..., w_m]^T$ is the vector of weights that connect the hidden layer and the output layer. ϕ_i is chosen as the Gaussian function, and the output of the hidden is

$$z_{ip} = \phi_i(\|X_p - C_i\|) = \exp\left[-\sum_{k=1}^{5} \frac{(x_{kp} - c_{ik})^2}{2\sigma_i^2}\right] \cdot \tag{7}$$

The learning algorithm of the network is described in detail in [11].

3 Experimental Studies

Experiments were conducted based on the cooling data of more than 1000 strips collected from a steel company. After the initial case base was established, the RBF networks were trained with selected data samples of respective grades. In the following experiment we use the data of fifty five segments, not included in the training data set, in a comparative study with the controller in [12]. The boundary conditions for the experiment are shown in Table 2, where *N* is the number of data segments used.

Table 2. Boundary conditions for the experiment

G_r	*d*(mm)	*L*(m)	T_h(℃)	v_h(m/s)	T_c(℃)	*N*
250	15.8	166.6	830	2.14	580	55

Table 3. Comparative statistics

H_t	H_b	π	*H*(8)	T_{cm}(8)(℃)	T_0(8)(℃)	580 ±10(℃)		580 ±30(℃)	
						This paper	Ref [12]	This paper	Ref [12]
9	91	A	82	618	581	55	15	55	49

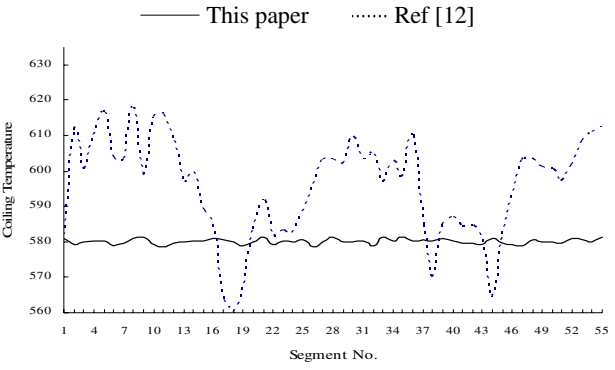


Fig. 2. Coiling temperature curves in comparison with [12]

Table 4 gives the first top activated header, the first bottom activated header, and the spraying pattern of the 8th segment as calculated by the case-based reasoner. $H(8)$ is the number of activated headers for the 8th segment calculated by the RBF network. The coiling temperature curves of this paper vs. ref [12] are plotted for comparison in Fig. 2.

We can see from Table 4 that ref [12] has the coiling temperatures of 49 segments within 30℃ of the coiling temperature setpoint, but only 15 segments within 10℃ of the setpoint. In contrast, the hybrid control strategy in this paper shows superior performance: 100% (55/55) of the coiling temperatures are within 10℃ of the coiling temperature setpoint. It is evident that the proposed control strategy is capable of much better accuracy than the controller in ref [12].

4 Conclusions

One of the key problems with the control of laminar cooling processes is that the final performance is not under closed loop control. This paper has presented an effective intelligent strategy for controlling the laminar cooling process that is able to adapt to changing operating conditions. Case-based reasoning is used to find the first activated headers and the spraying pattern. The RBF network is used to calculate the number of activated headers. Experimental studies based on industrial data have demonstrated that the hybrid control strategy contributes to significant performance enhancement. The control strategy proposed in this paper can be extended to a wide range of similar processes.

Acknowledgment

This work was partly supported by the Funds for Creative Research Groups of China under Grant No.60521003, the China National Key Basic Research and Development Program under Grant No.2002CB312201, and the Ph.D. Startup Fund of Shenyang University of Technology.

References

1. Chai, T.Y., Tan, M.H., Chen, X.Y., Li, H. X.: Intelligent Optimization Control for Laminar Cooling. In: Camacho, B., Puente, D. (eds.): Proc. of the 15th IFAC World Congress. Elsevier, Amsterdam (2003) 691-696
2. Groch, A.G., Gubemat, R., Birstein, E.R.: Automatic Control of Laminar Flow Cooling in Continuous and Reversing Hot Strip Mills. Iron and Steel Engineer 67(9) (1990) 16-20
3. Leitholf, M.D., Dahm, J.R.: Model Reference Control of Runout Table Cooling at LTV. Iron and Steel Engineer 66(8) (1989) 31-35
4. Moffat, R.W.: Computer Control of Hot Strip Coiling Temperature with Variable Flow Laminar Spray. Iron and Steel Engineer 62(11) (1985) 21-28
5. Ditzhuijzen, V.G.: The Controlled Cooling of Hot Rolled Strip: A Combination of Physical Modeling, Control Problems and Practical Adaptation, IEEE Trans. Aut. Cont. 38(7) (1993) 1060-1065
6. Iwamoto, M., Jeyerajan, M., Wada, H.: Process Control Computer System of Fukuyama No.2 Hot Strip Mill. Nippon Kokan Fukuyama Works Tech. Rep. 46 (1986)
7. Miyake, Y., Nishide, T., Moriya, S. Device and System for Controlled Cooling for Hot Strip Mill, Transactions ISIJ. 20(2) (1980) 496-503
8. Kolodner, J.L.: Case-Based Reasoning. 1st edn. Morgan Kaufmann, New York (1993)
9. Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches, AI Comm. 7(1) (1994) 39–59
10. Haykin, S.: Neural Networks: A Comprehensive Foundation. 2nd edn. Prentice Hall, Upper Saddle River, New Jersey (1999)
11. Tan, M.H., Chai, T.Y.: Laminar Cooling Process Model Development Using RBF Networks. ISNN 2006, In Press
12. Shan, X.: Transformation and Development of the Cooling Control System of the 2050mm Baosteel Hot Strip Mill. In: Ren, D. (eds.): Development of Science and Technology in Metallurgy. Metallurgical Industry Press, Hangzhou (1999) 1- 4

Application of Adaptable Neural Networks for Rolling Force Set-Up in Optimization of Rolling Schedules

Jingming Yang, Haijun Che, Yajie Xu, and Fuping Dou

Institute of Electrical Engineering, Yanshan University,
Qinhuangdao, P.R. China

Abstract. This paper presents two optimization procedures--single and multi objective optimization for 1370mm tandem cold rolling schedules, in which back propagation (BP) neural network is adopted to predict the rolling force instead of traditional models. Analysis and comparison with existing schedules are offered. The results show that the proposed schedules are more promising.

1 Introduction

Automation system for tandem cold rolling mills are continuously being improved due to today's stringent high throughput, quality and low scrap loss requirements for product [1]. Rolling schedule is an important aspect in the operation of tandem cold rolling mills. It refers to stand reductions, tensions, rolling forces, roll torques, mill speeds, and threading adjustments. The optimized schedules should lead to improved thickness, surface finish and shape performance of the products.

Since the rolling process is a nonlinear and closed coupling process with multiple variables, the accuracy of the models is not high. In this paper, we present an investigation into optimal schedules for five-stand tandem cold rolling mills based on BP neural network with Levenberg-Marquardt algorithm in the set-up of the rolling force. Two designs are proposed. The first design is with the energy-saving objective and the second one is to get relatively even load together with fine strip shape.

2 Objective Functions and Constraints

2.1 Objective Function

Based on the practical requirements, the objective functions are selected as follows:

Power cost function: The exit thickness in each stand h_i is taken as the state variable the minimum power cost is taken as the objective function.

$$min\, N = \sum_{i=1}^{n} N_i(h_{i-1}, h_i) \tag{1}$$

where n is the number of stands and $n = 5$ in this paper.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 864 – 869, 2006.

Relative even load function: The function is used to make the load even based on the physical capability of rolling mills. It is defined as follows:

$$min\ S = \sum_{i=1}^{n} (N_i - N_i^{'})^2 \tag{2}$$

where N_i and $N_i^{'}$ are the rated power and real power of the actuator at stand i, respectively. $N_i^{'} = \lambda N_i$, λ is the load factor.

Fine shape function: It is obtained when the strip is uniformly rolled across the strip width. This means that the deformed roll profile should perfectly match the incoming transverse strip thickness profile geometrically. The main influence factor is the rolling force, so objective function is selected as follows:

$$min\ S = \sum_{i=1}^{n} (P_i - P_{opti})^2 \tag{3}$$

where P_i and P_{opti} are the real rolling force and the calculated force for fine shape at stand i respectively.

2.2 Constraints for Validity Checks

Roll force: $P_i \le P_{max}$, where P_{max} is the maximum value due to the mechanical design limits imposed by manufacturers of the rolling mill and electrical drive motors.

Roll speed: $V_{min} \le V \le V_{max}$, where V_{min} and V_{max} are lower and upper limits corresponding to the production requirement and actuator references respectively.

Overload limit: $M_i \le K_\lambda \cdot M_H$, where K_λ is the overload coefficient, M_i and M_H are the roll torque and rated torque at stand i respectively.

3 Models of Rolling Process

Since there are many uncertainties in the rolling models of the coming coils[2], we take an artificial neural network(ANN) instead of the empirical rolling force model in rolling schedules. So a lot of work can be avoided with neural computing.

(1) Neural Network Modeling for Rolling Force

In this work an ANN leaning under supervision is used. A multi-layer BP network is selected, where the Levenberg-Marquardt algorithm is used for off-line training. This algorithm avoids the direct calculation of Hansen Matrix, and thus it needs less calculation works and saves memories. The resulting network structure is 7-10-7-1.

Input: $p = [h_0, h_1, t_b, t_f, K, B, R]$, Output： $t = P(i)$. Where h_0 and h_1 are the entry and exit thickness at the stand respectively, t_b and t_f are the front and back tension stresses respectively, K is the yield stress at exit points of roll bite, B is the strip width R is the work roll radius.

To avoid the non-convergence of the network, the transfer function between the input layer and the first hidden layer is $f(x) = tanh(x)$, and the transfer function

between hidden layers is $g(x)=1/(1+e^{-x})$. $Purelin(x)$ is selected as the active function in output layer. All input variables are normalized as follows:

$$\hat{x}=\frac{1.6}{x_{max}-x_{min}}(x-x_{min})-0.8 \tag{4}$$

where x is input variable, x_{max} and x_{min} are the minimum and maximum value of x respectively where the x_{max} and the x_{min} are selected by considering the physical limits and the range of logged data. These values are fixed in all the training periods.

The process data gathered in the data logging system are transferred to a local development system by the Ethernet. Here, the measurement lag between the exit thickness and the rolling force is compensated. Thus, the data's correspondence is realized indeed. The training set has 5000 samples, which are collected from cold tandem rolling over 12 months. The training process is shown in Fig.1.

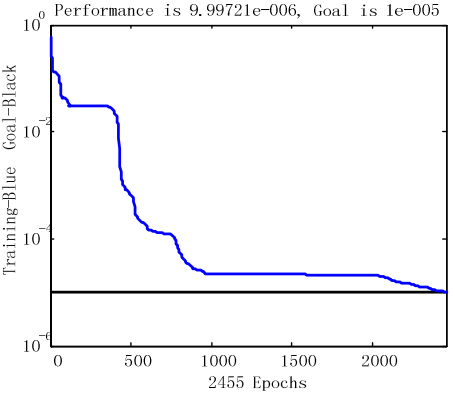


Fig. 1. Training result of BP L-Mt algorithm

Since the database can receive the data on-line, the network parameters can be continuously adjusted off-line. Then, the prediction accuracy for the next coil will be improved. Therefore, the schedules will adapt to the correction of the rolling force.

(2) Friction Coefficient

The friction coefficient μ dependents not only on the physical and chemical nature of the lubricant, but also on the rolling process parameters, such as reduction ratio, work roll speed, roughness between the work roll and strip. It can be described as:

$$\mu=\sqrt{\frac{\Delta h}{2R}}[0.5+(K1-0.5)e^{-K2V}] \tag{5}$$

Where Δh is the reduction, $K1$ is the ratio of μ to $\sqrt{\Delta h/2R}$ on the condition that the roller smoothness is 0.254μm, $K2$ is the influence factor of the roll speed to μ.

(3) Tension

The tension stress should be less than half of the yield stresses limit σ_s. It always is $(0.2\sim0.4)\sigma_s$. Besides, we should consider the power splitting between the stands and the coiler. Normally, the tension of continuous operation strain can be described as: $T = c\sigma_s B h_i$, $c = \beta(0.33\text{-}0.14h_i + 0.02h_i^2)$, where c is the ratio of the actual stress σ_0 to σ_s, β is the rolling mill strain coefficient. For the continuous annealing and coated lines, $\beta = 0.5$.

(4) Roll Torque and Power

The total roll torque can be described as follows: $M_\Sigma = M_P + M_f + M_X \pm M_d$. Based on the results of the roll speed and the roll torque, the power at each stand is estimated as: $N = 0.103 M_\Sigma n_p$, where M_P is the roll torque, M_f, M_X, M_d are the addictive friction torque, no-load motion torque and kinetic torque respectively, n_p is rotational speed of the motor shaft.

4 Optimized Schedules

(1) Single Objective Optimization

Since the tandem cold rolling is with the character of multistage decision, Bellman dynamic programming algorithm is used in the optimization in which energy saving is taken as the objective. Here, every rolling stand is taken as a decision process, and then the whole rolling process can be seen as a multistage decision process [3]. So it is suitable for the condition that the state and decision variables are both discrete. Therefore, every variable should be discrete before optimization. The shorter discrete interval is, the more the points are and the longer optimization time is.

Firstly, the variables are discrete at 0.002 mm equal interval in vertical direction based on the initial schedule. Then the whole rolling interval spacing at each stand together with the output thickness after discrete process can be calculated under the constraints. Next, we get other rolling parameters corresponding to each reduction. Finally, based on dynamic programming, we can choose a set of gauge distribution which leads to the total power least. The flow process chart is shown in Fig.2 (a).

(2) Multiple Objective Optimization

For 1370 tandem cold rolling mill, pursue of energy saving will bring the power distribution imbalance in five stands. The large-scale change of the roll force will increase the difficulty for shape adjustment. Thus, it is necessary to make use of the load distribution to keep shape [4]. The first four stands should run with as even load as possible. At the last stand S5, multi-objective load distribution is applied and the control system is in the C mode, i.e. S5 is only used as a planishing mill. There is minute reduction at S5 and the rolling force is constant during the rolling process, so the finished strip can be with fine shape. The flow chart is shown in Fig.2 (b).

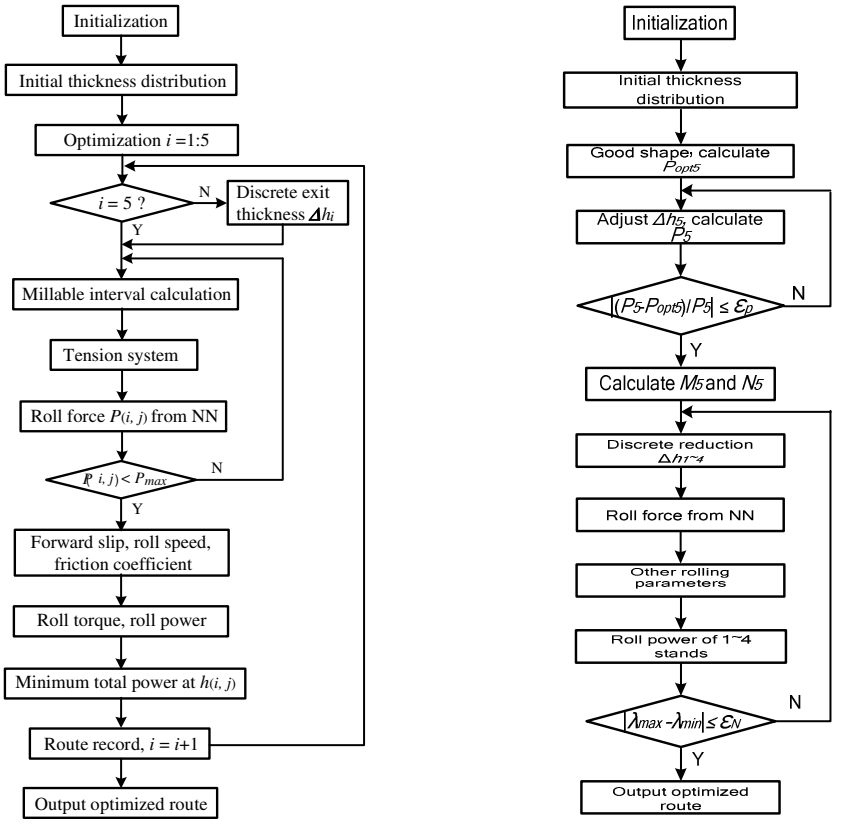


Fig. 2. (a) Flow chart of optimized schedule 1; (b) Flow chart of optimized schedule 2

(3) Application Example

We apply the optimization methods to 1370 five-stand tandem cold mill in one steelworks of Tangshan. Table 1 shows the mill parameters. Table 2 and 3 show the comparison results with different optimization objectives. From the results in Table 2 we can see the optimized schedule 1 consumes less power than the initial schedule by follows:

$$C = \frac{N_{old} - N_{opt}}{N_{old}} \times 100\% = 11.6\% \tag{6}$$

Table 1. Parameters of 1370 tandem cold mill

Mill parameters	Stand number				
	1	2	3	4	5
Rated power /kW	2572.5	2572.5	2572.5	3675	3675
Motor roted speed / r/min	135/305	175/375	225/445	250/500	250/500
Max roll force /kN	20000	20000	20000	20000	20000
Work roll diameter /mm	520.7	528.0	526.8	529.7	531.6
Backup roll diameter /mm	1380	1380	1310	1310	1310

The optimized schedule 1 shows obvious energy-saving performance. However, it brings big difference of the load coefficient between the 1st and 4th stand. To make good use of the mill motors, we proposed the multi-objective schedule 2, which gives attention to both load distribution and good shape. The results are shown in Table 3.

Table 2. The comparison with schedule 1

Rolling schedule		Exit gauge /mm	Reduction /%	Roll force /kN	Roll power /kW	Load factor
1	Initial	1.710	24.00	6899	1383.8	0.603
	Optimized	2.010	10.67	3720	537.3	0.209
2	Initial	1.190	30.41	8795	1832.7	0.815
	Optimized	1.490	25.87	7400	1466.4	0.570
3	Initial	0.850	28.57	8707	1805.2	0.839
	Optimized	0.930	37.58	9953	2312.6	0.899
4	Initial	0.580	31.76	9831	2441.3	0.822
	Optimized	0.522	43.87	12157	3667.8	0.998
5	Initial	0.500	13.79	11264	2733.1	0.760
	Optimized	0.500	4 . 21	5994	1026.4	0.279
Total power /kW				10196.1 9010.4		

Table 3. The comparison with schedule 2

Rolling schedule	Exit gauge /mm	Reduction /%	Roll force /kN	Roll power /kW	Load factor
Initial	1.710	24.00	7536	1551.9	0.603
Optimized	1.545	31.33	8513	1897.4	0.738
Initial	1.190	30.41	9883	2095.4	0.815
Optimized	1.070	30.74	9102	1905.4	0.740
Initial	0.850	28.57	9967	2159.6	0.839
Optimized	0.765	28.50	8906	1899.4	0.738
Initial	0.580	31.76	11746	3022.2	0.822
Optimized	0.520	32.02	10195	2685.6	0.731
Initial	0.500	13.79	11486	2793.4	0.760
Optimized	0.500	3.85	5994	1026.4	0.280
$\Delta\lambda_i$ $(i=1\sim4)$			0.007		

5 Conclusion

In this paper, two optimal rolling schedules method are proposed. The neural network is introduced to improve the model prediction ability for rolling force set-up in 1370 tandem cold mill schedules. Since the network learns from the on-line data, the revise schedule is more rational. Schedule 1 improves the energy-saving performance and schedule 2 makes power distribution balanced as well as good flatness. The optimization makes good use of the effect from the rolling force and the power distribution to strip shape, and lead to better combination of gauge and shape control. The example illustrates the performance of the proposed method.

References

1. Wang, D.D., Tieu, A.K., de Boer, F.G.: Toward a Heuristic Optimum Design of Rolling Schedule for Tandem Cold Rolling Mills. Engineering Appl. of Artificial Intelligence 13(4) (2000) 397-406
2. Lee, D.M., Choi, S.G.: Application of On-line Adaptable Neural Network for Rolling Force Set-up of A Plate Mill. Engineering Appl. of Artificial Intelligence 17(5) (2004) 557-565
3. Zhao, H.J., Zhang, Y.H., Hu, H.T.: The Optimized Design of Copper Strips Rolling Rules by Dynamic Programming Method. Journal of Southern Institute of Metallurgy 22(4) (2001) 243-246
4. Di, H.S., Xu, J.Z., Gong, D.Y.: Effect of Load Distribution on Strip Crown in Hot Strip Rolling. J. Maser. Sci. Technol. 20(3) (2004) 330-334

Multiple Neural Network Modeling Method for Carbon and Temperature Estimation in Basic Oxygen Furnace

Xin Wang[1], Zhong-Jie Wang[2], and Jun Tao[3]

[1] Institute of Information & Control Technology,
Center of Electrical & Electronic Technology, Shanghai Jiao Tong University,
Shanghai, P.R. China, 200030
wangxin26@sjtu.edu.cn
[2] Department of Control Science & Engineering, Tongji University, Shanghai, 200092
[3] Baosight Software Corporation, Shanghai, 201900, China

Abstract. In this paper, a novel multiple Neural Network (NN) models including forecasting model, presetting model, adjusting model and judgment model for Basic Oxygen Furnace (BOF) steelmaking dynamic process is introduced. The control system is composed of the preset model of the dynamic requirement for oxygen blowing and coolant adding, bath [C] and temperature prediction model, and judgment model for blowing-stop. In this method, NN technology is used to construct these models above; Fuzzy Inference (FI) is adopted to derive the control law. The control method of BOF steelmaking process has been successfully applied in some steelmaking plants to improve the bath Hit Ratio (HR) significantly.

1 Introduction

Basic Oxygen Furnace (BOF) steelmaking is one of the key processes in the iron & steel industry. The process of BOF steelmaking is a complex physical-chemical process, which takes hot metal (HM), pig iron and scrap as materials [1]. In order to decrease impurity compositions level (such as carbon, [P], [Mn] *etc*.) and raise bath temperature to catch the tapping aim, oxygen is ejected into the bath through the oxygen lance [2]. The tapping aim slot is determined mainly by the aim bath carbon [C], the temperature and sometimes by the bath [P], [Mn] *etc*. Because of the poor condition of BOF process, bath compositions can't be measured online [3]. So a sublance is added to measure the bath. Unfortunately, [C], the temperature, [P] and [Mn] of steel bath can't be measured continuously and operation conditions vary frequently, which makes it difficult to control the BOF end-point bath precisely. Actually, it often happens that operators have to re-melt the steel bath due to the low control precision of end-point bath. So improving the control precision of BOF steelmaking end-point is quite important [4]. In our earlier work, multiple models method was introduced into the field of BOF endpoint [P] & [Mn] estimation according to the analyzing speed of in-blow sublance sample [5]. For the [C] and the temperature estimation, intelligent method including fuzzy inference, neural network (NN) and expert system is proposed to improve the control precision of BOF steelmaking end-point [6]. However, a lot of expert experiences must be needed to set up the expert system.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 870 – 875, 2006.

In this paper, a novel multiple NN models method is presented for Basic Oxygen Furnace (BOF) steelmaking dynamic process, by combining Neural Network, Fuzzy Inference with dynamic process control method of BOF steelmaking. The control system is composed of the preset model of the dynamic requirement for oxygen blowing and coolant adding, bath [C] and temperature prediction model, and judgment model for blowing-stop. The control method of BOF steelmaking process has been successfully applied in some steelmaking plants and improves the bath Hit Ratio (HR) significantly.

2 Intelligent Control Method

In BOF steelmaking dynamic process, the controlled variables are carbon content and temperature of bath steel, and the controlling variables are the amount of oxygen blown-in and that of coolant added in the process. In this intelligent control method, one DCS is used in the controlling of oxygen blowing and coolant adding, and a set of method is applied to determine the set values of dynamic oxygen and coolant, and finally return the set values to the dynamic control system.

The control processes are mainly as follows: On the base of sub-lance measurement result and the tapping target, the initial set value of dynamic oxygen is determined firstly under the condition of no coolant added. Secondly, the calculated quantity of dynamic oxygen and dynamic coolant (initially zero) and other relative information are all transmitted to the forecast model, which precalculates the bath end-point [C] and temperature under the condition of the current calculated quantities of oxygen and coolant. Thirdly adjusting model will modify the quantities of oxygen and coolant with the difference between the forecasting result of bath [C], temperature and their tapping targets. Then the adjusted value is transmitted to the forecast model to predict the bath [C] and temperature under the new operation condition. The process above is recycled until the judgment model for BOF blowing-stop demonstrates that the bath [C] and temperature have hit the tapping targets, then the amount of oxygen and that of coolant obtained are just the final set values and will be sent to the corresponding DCS.

2.1 Forecasting Model

During the procedure of BOF steelmaking, forecasting model, based on SL1 sublance measurement results $\left(y_{1,k}(t_0), y_{2,k}(t_0)\right)$ (here k presents the kth heat) and the amount of dynamic controlling variables $\left(u_{1,k}(t), u_{2,k}(t)\right)$, predict the bath [C] and temperature real-timely, and finally realize the soft-measurement of the BOF bath.

Three Three-Level Back Propagation (BP) NNs are applied to predict [C] and temperature of the steel bath in the BOF dynamic process. The hidden level of the first NN adopts the Gaussian function as its responding function, which reflects the character of the middle part of the data set; While the hidden level of the second and the third NN adopt the mode of Gaussian function and Tanh function as its responding function respectively, which reflect the character around the upper and lower part of the data set.

2.2 Presetting Model

The task of presetting model is to determine the dynamic supplementary oxygen requirement with no coolant added, to warrant bath [C] level decreasing from BOF dynamic process start point $\left(y_1^k(t_0), y_2^k(t_0)\right)$ and hitting the tapping aim $\left(y_{1AIM}^k, y_{2AIM}^k\right)$. Radial Basis Functions (RBF) NN is adopted here.

There are two stages in the training of RBF NN, one is the learning of the RBF centres, and the other is the training of the connection weight of the output layer [7]. In the learning of the RBF centres, Fuzzy C-means method is adopted [8], and the classification object function is to minimize the sum of the square of the distances between the samples and cluster centres. The approximation function of RBF NN is $F(x) = \sum_{i=1}^{c} w_i G(\|x - v_i\|)$, where: $G(x)$ is Gaussian Function.

Self-adaptation module of forecasting model is executed to implement on-line learning function (adjusting connection weight) of BP NN by feedback learning of the network error, which is the difference between the RBF NN output and the actual measurement results of the bath end-point [C] and temperature.

The oxygen requirement for BOF dynamic steelmaking process is relative to bath [C] level of sublance blowing-in measurement, the [C] tapping target $y_{1AIM}^k(t_{end})$ and the quantity of the coolant added during the BOF steelmaking dynamic procedure. As a result, the input node number of the presetting NN is 3 and the node number of hidden layer is determined in training and the output node number is 1, which is corresponding to the oxygen requirement $\Delta u_{1SET,1}^k$ for BOF dynamic steelmaking process.

After the sample learning and training, the presetting RBF NN can implement the function of presetting the BOF dynamic oxygen requirement. The coolant amount is zero when determining the amount of the supplementary oxygen.

In the process of presetting calculation, the quantity of the added dynamic coolant is set with zero and the amount of the supplementary oxygen obtained from the NN is only an initial value. Based on the initial value of dynamic oxygen requirement, adjusting model will regulate the final quantity of oxygen and coolant for BOF dynamic process.

2.3 Adjusting Model

Presetting model only takes bath [C] content as the controlled variable, and determines the dynamic oxygen requirement under the condition of no coolant added. The bath temperature is controlled with adding coolant, but there exists solid oxygen in the coolant so it is necessary to modulate the dynamic oxygen volume to balance the influence of the coolant on the bath [C] content during the controlling of the bath temperature by adding coolant. Therefore adjusting model includes two sections: the adjusting model of determining the coolant requirement and that of modifying dynamic oxygen volume.

In adjusting model, the control errors $\Delta y_{1,j}^k, \Delta y_{2,j}^k$ of bath [C] and temperature are both selected as model inputs, and the adjusting results $\Delta u_{1SET,j}^k, \Delta u_{2SET,j}^k$ as model

outputs. Here, the two models both adopt Fuzzy Inference technology and Fuzzy Inference Rules are adopted *T-S* rules as follows [9]

$$\text{If } y \text{ is } A_i \text{ then } \Delta u \text{ is } u_i .$$

The input variable of the adjusting model is $\Delta y_{1,j}^k = y_{1AIM}^k - y_{1,j}^k(t)$ or $\Delta y_{2,j}^k = y_{2AIM}^k - y_{2,j}^k(t)$, the output variable is $\Delta u_{1,SET}$ or $\Delta u_{2,SET}$ and the membership functions of input and output variables are triangular.

$$\begin{aligned}
\Delta Y_1 &= \{NVB, NB, NM, NS, ZE, PS, PM, PB, PVB\} \\
&= \{-4, -3, -2, -1, 0, 1, 2, 3, 4\} \\
\Delta Y_2 &= \{NB, NM, NS, ZE, PS, PM, PB\} \\
&= \{-15, -10, -5, 0, 5, 10, 15\} \\
\Delta U_{1SET} &= \{NVB, NB, NM, NS, ZE, PS, PM, PB, PVB\} \\
&= \{20, 10, 5, 2, 0, -2, -5, -10, -20\} \\
\Delta U_{2SET} &= \{NB, NM, NS, ZE, PS, PM, PB\} \\
&= \{0.1, 0.05, 0.02, 0, -0.02, -0.05, -0.1\}
\end{aligned}$$

The relationship between the quantity of dynamic oxygen and bath [C], and that between coolant and bath temperature, is obvious, so the adjusting rules of the quantity of dynamic oxygen and coolant can be easily obtained. The output of the adjusting model is obtained by fuzzy inference (Mamdani method here) and defuzzization operation (Gravity method).

2.4 Judgment Model for BOF Blowing-Stop

Based on the soft-measurement result $\left(y_1^k(t), y_2^k(t)\right)$ of the forecasting model and the taping aim slot $\left(y_{1AIM}^k, y_{2AIM}^k\right)$, judgment model for BOF blowing-stop accomplishes the BOF blowing-stop function by judging whether the bath [C] content and temperature has hit the tapping target with Fuzzy Inference technology. When the tapping target [C] and temperature are both hit, this model will send the blowing-stop command to DCS of oxygen blowing system if computer control style is selected.

The intelligent control method of BOF steelmaking dynamic process can be concluded as:

- applying one DCS to control the oxygen blowing-in circuit and the coolant adding circuit;
- determining quantity of oxygen to be blown-in and that of coolant to be added with a set of intelligent method;
- sending the determined value to homologous control circuits.

3 Industrial Test

The intelligent control method above for BOF steelmaking dynamic process was applied successfully to one control system of 250t BOF, which was established in 1992.

The computer network is constructed by Ethernet technology, and the operation system is VMS and the server is Alpha DS20. The original control model was built by mechanism and its hit ratio of bath [C] and temperature was about 83%.

In general, in the industrial test there are two procedures, establishing the control model and validating the control model. In the process of establishing control model, 180 heat records of the BOF steelmaking are collected firstly, which contains bath weight, sublance blowing-in measurment result, bath end-point [C] content and temperature, expenditures of oxygen and coolant in BOF dynamic process and etc. Secondly, the bath [C] and temperature forecast models (the BP NNs) are established, then the presetting RBF NN (presetting model) is trained with the collected records.

In the industrial test, experimental records of 116 heats are obtained to validate the BOF intelligent dynamic control system. The results are shown in Fig.1 and Fig. 2, where "◇" denotes the bath end-point carbon content or temperature and "*" denotes the bath taping target.

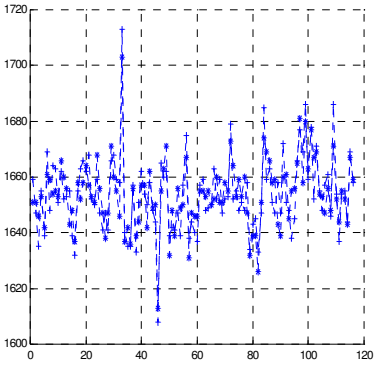


Fig. 1. The Bath end-point Temperature

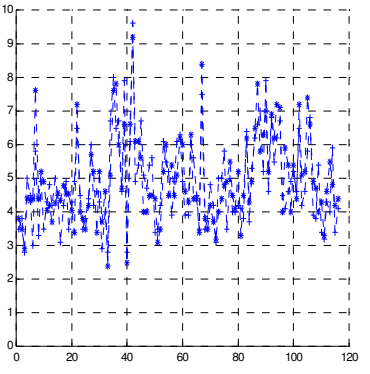


Fig. 2. The Bath end-point carbon

Among those test heats above, when the hit aim slots are selected as ±1.8% for temperature and ±14℃ for [C], there are 111 heats of bath end-point carbon hitting the target [C], and 107 heat of bath end-point temperature hitting the target temperature, and 103 heats of both bath [C] and bath temperature hitting the taping targets. Statistically, the Hit Ratio of bath [C], bath temperature and both of them, are 95.69%, 92.24% and 88.79% respectively. It demonstrates that the new method of BOF intelligent dynamic control introduced in this paper can improve the bath Hit Ratio greatly (about by 4~5%), and also shows wonderful ability of control the end-point bath.

4 Conclusion

For BOF steelmaking dynamic process, a novel multiple NN modeling method is presented by combining Neural Network, Fuzzy Inference with dynamic process control method of BOF steelmaking. The control system is composed of the preset

model of the dynamic requirement for oxygen blowing and coolant adding, bath [C] and temperature prediction model, and judgment model for blowing-stop, which has been applied in some steelmaking plants and improves the bath Hit Ratio (HR) significantly.

Acknowledgements

This work is supported by National Natural Science Foundation (60504010), SJTU Research Foundation and Municipal Technological Super-Star Project of Shanghai (No. 04QMX1429)

References

1. Sumi, I., Kawabata, R., Kikuchi, Y., *et al.*: Technique of Controlling Dust Generation during Oxygen Top Blowing in BOF. Steel Research., 74 (2003) 14–18
2. Swift, T.: BOF Bath Level Measurement for Lance Height Control at the Sparrows Point Plant. Iron and Steelmaker 29 (2002) 37–40
3. Nirschel, W.F., Stone, R.P., Carr, C.J.: Overview of Steelmaking Process Control Sensors for the BOF. Iron and Steelmaker 28 (2001) 61–65
4. Dai, Y.G., Li, W.X., Long, T.C.: Modern BOF Steelmaking. Northeastern University Press, Shenyang (1998) (in Chinese)
5. Wang, X., Li, S.Y., Wang, Z.J., Tao, J., Liu, J.X.: A Multiple RBF NN Modeling Approach to BOF Endpoint Estimation in Steelmaking Process, In: Yin, F., Wang, J. and Guo, C. (eds.): Advances in Neural Networks. Lecture Notes in Computer Science, Vol. 3174. Springer-Verlag, Berlin Heidelberg New York (2004) 175–180
6. Tao, J., Wang, X., Chai, T.Y., *et al.*: Intelligent Control Method and Application for BOF Steelmaking Process Proceedings of the IFAC World Congress (2002) 1071–1076
7. Mao, K.Z.: RBF Neural Network Center Selection Based on Fisher Ratio Class Separability Measure, IEEE Transactions on Neural Networks 13 (2002) 1211–1217
8. Pedrycz, W.: Conditional Fuzzy Clustering in the Design of Radial Basis Function Neural Networks. IEEE Transactions on Neural Networks 9 (1998) 601–612
9. Azeem, M.F., Hanmandlu, M., Ahmad, N.: Generalization of Adaptive Neuro-fuzzy Inference Systems, IEEE Transactions on Neural Networks 11 (2000) 1332–1346

Air-Fuel-Ratio Optimal Control of a Gas Heating Furnace Based on Fuzzy Neural Networks

Heng Cao[1], Ding Du[1], Yunhua Peng[2], and Yuhai Yin[1]

[1] School of Mechanical and Power Engineering,
East China University of Science and Technology, Shanghai, China
[2] Shanghai Baosteel Chemical Co. Ltd., Shanghai, China

Abstract. Based on Neural Network BP algorithm and self-optimizing control, taking gas heating furnace air-fuel-ratio optimized control as goal, a new heating furnace intelligent control algorithm is raised and applied in the practice. Comparing fuzzy neural network hybrid algorithm and PID control algorithm, with gas heating furnace energy-saving control reconstruct, new algorithm can achieve function of automatic tracking calorific value variable and adjusting air-fuel-ratio. The characteristics of this algorithm are high precision and reliability, and suitable for project application.

1 Introduction

The commonly methods of heating furnace air surplus coefficient examination and control are divided into two types, feedback control according to the combustion product remnant oxygen content and the feed-forward according to calorific value measurement result. Because of life, reliability and price of these two methods examination measurement, application is limited. There are many gas heating furnaces air-fuel-ratio are adjusted according to experience, for feed-forward control method on-line analysis was still not mature and also cannot meet the needs. Thus, the production stability, reducing consumption and product quality can be influenced badly, especially when gas calorific value fluctuating frequently and greatly, the calorific value measurement reliability is low, meanwhile, when feedback control is used, its application is constrained by big time-delay. At present, the convention control system cannot automatic track and adjust proper air-fuel-ratio. It is main barrier which influences heating furnace control technology development. With increasing of intelligent control technology, and heating furnace air-fuel-ratio control effect can be improved by the fuzzy control, more research are carried on.

With the fuzzy neural network, the 9101$^{\#}$ sweeping oil gas heating furnace energy-saving reconstruction of a chemical plant of Baoshan Steel Corporation is carried on in the laboratory; the PID control is instead by fuzzy neural network control. The gas heating furnace which should be reconstructed is a sweeping oil heating furnace, in the chamber is negative pressure burning, therefore the control difficulty is big. The negative pressure in the chamber can raise flame; causing burning sufficiently, at the same time, the negative pressure can be used to pump air needed by burning. Because of the

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 876 – 884, 2006.

multi- inputs/outputs characteristics of the negative pressure furnace, the control process is especially complex. The present production situation is: The operator only can observe the burning condition through the experience then adjust the import gas and air content, lacks the essential quantification process, most of the time cannot use energy maximum; because the scene examination is not timely, therefore burning over-sufficient or burning under-sufficient often happen, which waste energy and pollute environment.

The artificial intelligence theory and the on-line monitor technology are combined to use in the heating furnace process parameter observation, according to off-line or on-line process information reconstructed by soft-measuring technique; the running status, working condition, temperature and air (gas) flow should be monitored and controlled; real-time performance and stability of key parameter monitor is enhanced greatly.

2 Optimizing Heating Furnace Burning Control with Fuzzy Neural Network

2.1 Heating Furnace Control Requirements

Gas heating furnace production process and sensor configuration are shown as fig. 1:

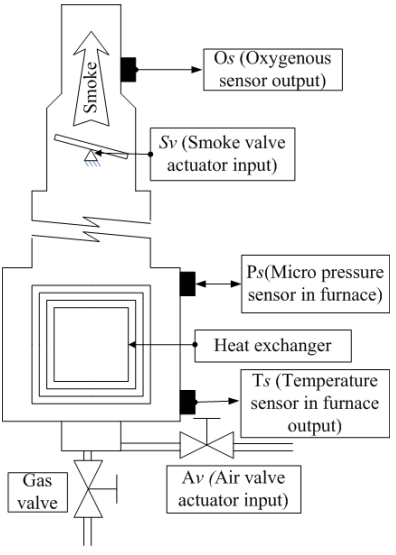


Fig. 1. The production process and sensor configuration of gas heating furnace

Where, O*s* - remnant oxygen content; S*v* - The actuator angle of flue tilting plate open; P*s* – micro-pressure value of the furnace chamber; T*s* – temperature of chamber; A*v* – airflow rate.

2.1.1 Temperature Control Process

The common heating furnace temperature control goal is: When heating furnace temperature is invariable, gas/air flow rate is constant; when furnace temperature deflect from the provided value, gas/ air flow must make some adjustment to enable the heat energy supply and demand to achieve a new balance, thus maintains the temperature invariable. The gas flow pressure of reconstructed furnace fluctuated slightly in the production process; therefore gas flow rate almost does not need to make any adjustment. But negative pressure value (P*s*) in chamber is affecting temperature (T*s*) directly, P*s*, as an assistant index of combustion condition, mainly demonstrate the flame raise. In the chamber negative pressure may guarantee heating sufficient, only flame raising, the furnace upside can obtain full heating, and maintains T*s* to be constant. But on the other hand, if P*s* value is excessively high, flame raise excessively high, the air input also increases, and heat of chamber can take away by the air, then T*s* also drops.

2.1.2 Air-Fuel-Ratio Control

A main goal of heating furnace control is guaranteeing the combustion process on the condition of perfect air-fuel-ratio. When air-fuel-ratio is excessively high, heat of chamber can take away by remnant air, heat energy loss increases; when air-fuel-ratio is excessively low, combustion is not sufficient, then wasting fuel and polluting environment. Thus air-fuel-ratio is a important link in the heating furnace control process. Air is pumped in the chamber by negative pressure, so air-fuel-ratio cannot be controlled directly through adjusting airflow valve, but through examining remnant oxygen content (O*s*), the reasonability can be judged, experiment indicated reasonable O*s* value is 2% ~ 6%. At present, most heating furnace temperature control and air-fuel-ratio control uses the PID controller, but it often appear overstep, longer adjusting time, control effect is not good, in the coupling system, the PID controlled variable is not easy to tuning, but easy to make vibration even out of control. It cannot meet request of heating furnace energy-saving and environment-protecting.

2.2 The Application of Fuzzy Neural Network in Heating Furnace Air-Fuel-Ratio Control

The intelligent control deduced from the artificial intelligence, emphasizes massive qualitative prior knowledge, uses fuzzy, inference, logical knowledgebase and inference method, through off-line training, on-line study, the intelligence dynamic control system is established, instead of the traditional controller. The fuzzified process is fuzzifying furnace temperature deviation e and furnace temperature deviation rate ec into linguistic value E and EC. When fuzzifying, the universe of e is divided into 7 grades; the universe of ec is divided into 5 grades. The linguistic value E contains 7 subsets: {PB (positive big), PM (positive middle), PS (positive small), 0, NS (negative small), NM(negative middle), NB (negative big)}; the linguistic value EC contains 5 subsets: {PB, PS, 0, NS, NB}; Output variable: the gas flow output em, fuzzy linguistic value EM contains 7subsets: {PB (positive big), PM (positive middle), PS (positive small), 0, NS (negative small), NM (negative middle), NB (negative big)}; The heating furnace control rule may be concluded as the following sentence:

$$\text{If } E = E_i \text{ and } EC = EC_i \text{ then } EM = EM_i$$

E_i, EC_i, EM_i is respectively one of E, EC and EM subsets. The fuzzy inference decision is searching fuzzy control rule through fuzzified linguistic variable E and EC, and finding fuzzy control strategy EM. After EM un-fuzzified, the precise control value em is obtained, and then em is applied in the heating furnace.

The fuzzy control biggest problem is how to determine the membership function and the fuzzy control rule. According to specialized knowledge and operating experience, the obtained fuzzy control rule often cannot adapt the scene complex condition, but fuzzy control can use the neural network through auto-adapt to solve this problem, then a new fuzzy control method is created.

2.3 Auto-adaptive Fuzzy Neural Network Structure

The artificial neural network is non-linear dynamic input/ output system which is combined by massively node, that is neuron; it can simulate the human brain to process the knowledge, and has very strong classification ability, association ability and learning ability, thus in the aspect of processing people's experience knowledge, it has high intelligence quotient. There are five layers of neural network to be used to realize every fuzzy controller part: e and ec fuzzifying, fuzzy inference decision and eM un-fuzzified, the structure is shown as fig.2.

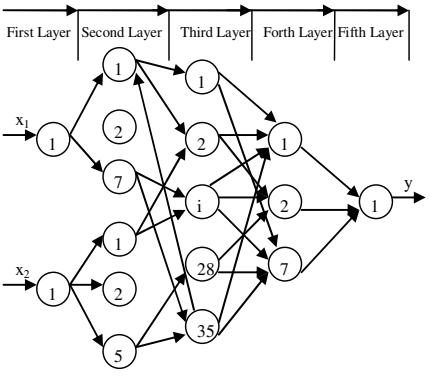


Fig. 2. Five layers neural network fuzzy controller

First layer: The first layer neuron is input node, as fuzzy controller input signal. This neural network has two input nodes, representing temperature deviation and deviation rate. The first layer neuron only transports the input value to next layer. Therefore:

$$\text{Net}^{(1)} = x_i \,, \ (\ i = 1, \ \ 2\ ) \tag{1}$$

$$\text{Out}^{(1)} = x_i \tag{2}$$

The first layer connection weight is 1.

Second layer: This layer denotes the linguistic value of input signal linguistic variable, each neuron represents one kind of linguistic value, and expresses as a

membership function. Then each neuron output should be the corresponding membership function.

$$Net_i^2 = Out_j^{(i)}, i = 1, 2, \wedge, 12 \tag{3}$$

$$j = 1,\ i = 1,\ 2, \Lambda,\ 12$$

$$Out_i^{(2)} = \exp\left\{\left[\frac{Net_i^{(2)} - m_i^{(2)}}{\sigma_i^{(2)}}\right]^2\right\}\ i = 1,\ \ 2, \Lambda,\ \ 1 \tag{4}$$

Where, m_i and σ_i respectively represents the jth linguistic value Gauss membership function center and width of input x_i linguistic variable. The second layer connection weight is $m_i^{(2)}$.

Third layer: Altogether have 35 nodes, each node represents a fuzzy rule; the connection between first and second layer represents the matching fuzzy rule, excitation strength of each rule is determined by output. That is:

$$Net_i^{(2)} = \min(out_j^{(2)}, out_k^{(2)})\ \ i = 5(j-1) + (k-7) \tag{5}$$

$$j = 1, 2, \Lambda, 7;\ k = 8, \Lambda, 12$$

$$Out_i^{(3)} = Net_j^{(3)}, i = 1, 2, \wedge, 35 \tag{6}$$

The third layer connection weight is 1.

Fourth layer: Altogether have 7 nodes, 7 fuzzy division number of gas flow output variable. Each node executes fuzzy "or" to synthesizes rule with same output; this function represents weight W_{ij}, which denotes the connection strength between the ith output linguistic value and jth rule; W_{ij} value only takes 0 or 1.

$$Net_j^{(4)} = \sum_{j=1}^{7} W_{ij} Out_j^{(3)} \tag{7}$$

$$Out_i^{(2)} = \min(1, Net_k^{(4)}),\ \ i = 1,\ 2, \Lambda,\ 7 \tag{8}$$

Fifth layer: the neuron is used to solute fuzzy. If fuzzy subset membership function center and width of the fourth layer jth neuron are mj and σ_j , then the following function can be used to imitate the gravity model to solute fuzzy:

The fifth layer connection weight is m (4)

$$Net_j^{(5)} = \sum_{j=1}^{7} W_{ij} Out_j^{(3)} \cdot \tag{9}$$

2.4 Hybrid Learning Algorithm of Fuzzy Neural Network System

Based on the above structure, a hybrid learning algorithm is raised; the whole system construction process is divided into three steps: The determination of initial membership function, the rule extraction and adjusting membership function optimized.

2.4.1 The Determination of Initial Membership Function

$$\| x(t) - m_{closest} \| = \min \{ \| x(t) - m_i(t) \| \} \tag{10}$$

$$m_{closest}(t +1) = m_{closest}(t) + a(t)[x(t)- m_{closest}(t)$$
$$m_i(t+1) = m_i(t),\ m_i \neq m_{closest}$$

$$\sigma_i = \frac{| m_i - m_{closest} |}{r}$$

Where, k is division number of x; α(t) is learning rate; r is overlapping parameter. According to the provided training data $x_i(t)$, i = 1, 2,Λ, n, the ideal output is $y_i(t)$, i = 1, 2,Λ, m; The fuzzy division number of x and y as well as the ideal membership function shape, use self-organization feature map algorithm to determine membership function central point and width value; taking gauss function as an example, that m_i and σ_i is determined, i = 1, 2,Λ, n.

2.4.2 Establishment of Rule

There is only one fuzzy rule conclusion, therefore need to find a algorithm to determine which conclusion node of fourth layer should connect to the rule node of third layer, then deletes this rule node with other conclusions connection, which makes reasonable form. After the membership function parameter is determined, training data can be transported from fuzzy inference system to second and fourth layer, then node output of second layer can reach the rule node, so it can get excitation strength of every node rule; based on excitation strength of rule node and fourth layer output, fuzzy rule can be found through correct and reasonable connection of every rule node is determined. At the beginning, the fourth layer connection is completely, W_{ij} represents the connection weight of ith node and jth node. To each training data set, competition learning principle is used to revise the weight: $W_{ij}=O_j^{(4)}(-w_{ij}+o_j^{(3)})$, $O_j^{(4)}$ represents the output of fourth layer jth node. Thought of this principle is the victor learning. Under extreme situation, if $O_j^{(4)}$ is a 0 ~ 1 universe function, then the above principle mean only the victor can learn. Through the above algorithm, fuzzy rule is determined, and some rule nodes influencing output little can be deleted, thus number of node is reduced, and fuzzy inference system structure scale is reduced too.

2.4.3 Optimizing Membership Function

After fuzzy rule and inference structure are determined, parameters of membership function need to be optimized. Most of time, optimization is to find optimal solution in a bunch of reasonable solutions through optimized principle. According to the provided training data $x_i(t)$, i = 1, 2,Λ, n, the ideal output is $y_i(t)$, i = 1, 2,Λ, m; The fuzzy division number of x and y as well as initial membership function shape and fuzzy rule, through adjusting membership function parameter, change universe covering situation of respective fuzzy subset of variable x, y, thus achieve the fuzzy inference model output approach the constructed fuzzy system output infinitely in the mean-square deviation. This algorithm is auto-adapted decoupling BP algorithm. The index function is:

$$E = \frac{1}{2}(Y_d - Y)^2 \tag{11}$$

Where, Y_d is network expectation output, and then the adjusting function of network parameter is shown as (taking the fourth layer jth neuron which represents the center m_j and width σ_j of fuzzy subset membership function as example):

$$m_j(k+1) = m_i(k) - a\frac{\partial \sum}{\partial m_j}\eta \Delta m_j(k)$$

$$\Delta m_j(k) = m_j(k+1) - m_j(k)$$

α is learning rate, η is momentum factor, both of them have auto-adapted ability.

3 Selection of Training Sample Set

According to gas heating furnace production log, selecting group of samples (100) as training sample set, as table 1 shown. Feedback input variable: T*s* - furnace temperature, O*s* – remnant oxygen content.

Control output variable: P*s* - negative pressure in chamber (indirectly denote flue tilting plate valve open angle), A*v* - air flow (directly denote air valve open angle).

Table 1. Training sample set

No.	P*s* (Pa)	A*v* (m^3/h)	O*s* (%)	No.	P*s* (Pa)	A*v* (m^3/h)	O*s* (%)	No.	P*s* (Pa)	A*v* (m^3/h)	O*s* (%)
1	−70	250	4.1%	10	−70	255	5.0%	92	−80	275	4.5%
2	−75	250	3.8%	11	−75	255	4.6%	93	−85	275	3.7%
3	−80	250	3.4%	12	−80	255	3.9%	94	−90	275	2.8%
4	−85	250	3.0%	13	−85	255	3.3%	95	−95	275	2.6%
5	−90	250	2.5%	14	−90	255	2.9%	96	−100	275	2.4%
6	−95	250	2.6%	…	…	…	…	97	−105	280	2.9%
7	−100	250	2.7%	…	…	…	…	98	−110	280	3.7%
8	−105	250	3.1%	90	−70	270	6.0%	99	−115	280	4.5%
9	−110	250	3.9%	91	−75	270	5.5%	100	−120	280	4.6%

4 Simulation and Experiment Result

The DCS control system of FOXBORO Company which is used by this fuel gas heating furnace is directly used in the scene of industry control, then for the production safety, it is not suitable for debugging in the scene while algorithm is not mature. So a set of hardware in-the-loop simulation debugging and test system is established with Siemens (S7-300) PLC. At the same time, Matlab Simulink Box is also used. When the controlled object is not typical non-linear, the mathematical model is $y(t+1) = \frac{y(t)}{1+y^2(t)} + u^3(t)$, shown as fig.3., curve y1 is a step response curve of reference model; Curve y2 is response curve of FNNC control method; Curve y3 is error curve of the previous two curves. The controller based on fuzzy neural network can along with output of reference model, and obtains good control effect.

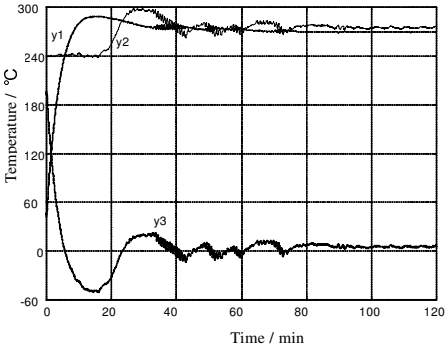


Fig. 3. The in-the-loop simulation test curve

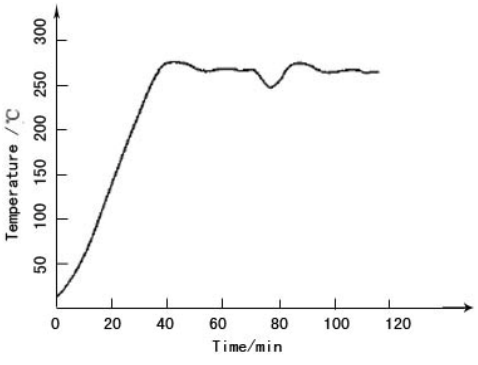


Fig. 4. Temperature real-time control response curve

After the massive simulations experiment, the algorithm is proved. Through real-time on-line appropriate adjusting each parameter, and add all kinds of disturbance and experimental system itself lagging 120 seconds, the satisfied control effect can be obtained. When temperature is 270℃, a disturbance is added, the system can reach stability again, in the situation, the temperature real-time control response curve is shown as fig.4. which indicate that this method has strong robustness to the big lag system.

5 Conclusion

The gas heating furnace energy-saving reconstruction of a chemical plant of Baoshan Steel Corporation demonstrated that, when fuzzy neural network technique is applied in gas heating furnace flue remnant oxygen content monitor and control, combustion efficiency might be enhanced, energy is saved, pollution is reduced, and meanwhile the furnace temperature optimizing control is realized. The project reconstruction indicate that, comparing with PID algorithm, the hybrid control algorithm has high precision and reliability, quick inference, good tracking, strong disturbance, feasibility and the usability, suitable for project application.

References

1. Vesloki, T.A., et al.: Application of a Dynamic Model of a Reheating Furnace. Iron and Steel Engineer 59(4) (1982) 46-51
2. Holland, F.D.: Development and Performance of On-line Computer Control in a 3–Zone Reheating Furnace Iron and Steel Engineer 59(1) (1982) 44-52
3. Las, K.: Advance Process Control Handbook I~VII. Hydrocarbon Processing (1986-1992)
4. Ronald, R Y.: Implementing Fuzzy Logic Controllers Using a Neural Network Framework. Fuzzy Sets and Systems 48(8) (1992) 53-64.
5. Lee, H.M., Chen, C., Huang, T.C..: Learning Efficiency Improvement of Back-propagation Algorithm by Error Saturation Prevention Method. Neuron Computing 41 (2) (2002) 125-143
6. Smith, O.J.M.: Closer Control of Loops with Dead Time. Chemical Engineering Progress Trans. 53(5) (1997) 217-219
7. Majhi, S., Atherton, D.P.: Obtaining Controller Parameters for a New Smith Predictor Using Autotuning. Automatica 36(6) (2000) 161-168
8. Tan, K.K., Lee, T.H., Leu, F.M.: Predictive PI Versus Smith Control for Dead-time Compensation. ISA Transactions (2001) (40)17—29
9. Li, W., Chang, X.G., Wahl, F.M. et al.: Tracking Control of a Manipulator under Uncertainty by Fuzzy PID Controller. Fuzzy Sets and Systems 15(23) (2002)125-137
10. Gao, X., Li, S., Chai, T.: Modeling and Intelligent Compound Control of Direct Current Electric Arc Furnace. Control Theory and Applications 6(16) (1996)49-52
11. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning Internal Representations by Error Propagation in Parallel Distributed Processing. Instrumentation in the Power Industry 33(5) (1996) 17-29
12. Maffezoni, C.: Boiler-turbine Dynamics in Power-plant Control. Control Engineering Practice 5(3) (1997) 303-304
13. Koo, K.-D., Lee, O.-J..: Technical Review of Enhanced Process Digital Control and Automation at the Poryong Thermal Power Plant. Industry Computing Conference 5(3) (1997) 1-6
14. Emoto, G.: Integrated Large-scale Multivariable Control and Real-time Optimization of a Power Plant 16(8) (1995) 11-18

An Evolutionary Hybrid Model for the Prediction of Flow Stress of Steel

Ai-ling Chen, Gen-ke Yang, and Zhi-ming Wu

Department of Automation, Shanghai Jiao Tong University,
Shanghai 200240, China
chengchengcal@sjtu.edu.cn

Abstract. A new hybrid model combining evolutionary artificial neural network (EANN) and mathematical models (MM) is proposed to improve the prediction precision of flow stress of 45 steel. In EANN, the optimal parameters are obtained by chaotic particle swarm optimization (CPSO) algorithm. CPSO adopts chaotic mapping and combines local search and global search, possessing high search efficiency and good performance. The results obtained from the computational study have shown that the proposed model can correctly recur to the flow stress in the sampled data and it can also predict well the non-sampled data. The efficiency and accuracy of the proposed model are demonstrated in comparison with the model combining BP networks with mathematical models (BPN-MM) used in much literature.

1 Introduction

In hot rolling process, setting reasonable rules of rolling technics is a critical factor to obtain the rolled products of higher quality. In order to set reasonable rules of rolling technics, it is necessary to correctly obtain the flow stress of metals under different processing conditions. Over the past decades, considerable researches [1-3] have been done in order to develop mathematical models to establish the relation between flow stress and strain, strain rate, temperature. However, during the hot deformation process, there are many factors that influence flow stress. The influence is very complex and most of the factors are non-linear, so it is very difficult to establish an advisable regression model. In recent years, owing to the need for higher efficiency and precision in prediction, artificial intelligent methods have been used because they possess massive parallel computing capability and strong ability to simulate non-linear data.

In the area of the flow stress prediction, one of the most popular intelligent models is BP network (BPN) [4-6]. However, BPN also suffers from some disadvantages, such as local minima problem, slow rates of convergence, and the danger of over-fitting, etc. In the paper, an evolutionary hybrid prediction model is proposed to improve the prediction precision of flow stress. The hybrid model makes use of chaotic particle swarm optimization (CPSO) algorithm to automatically choose the parameters of ANN. The algorithm avoids jamming and improves the precision of prediction.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 885 – 890, 2006.

2 Method Description

2.1 Artificial Neural Networks

The back-propagation (BP) algorithm has been widely used for training feed-forward neural networks. It is popular and applied to many fields. However, owing to the update disharmony between weights connected to the hidden layer and the output layer caused by the saturation behavior of the activation function used for the network layers, the random perturbations of the search direction and various kinds of stochastic adjustments to the current set of weights, the convergence rate of network is slow and the network is easy to become trapped in the local optimum. To overcome the disadvantages of BPN and improve the prediction precision of the model, chaotic PSO is used to choose the parameters of the model.

2.2 Chaotic Particle Swarm Optimization (CPSO)

The particle swarm optimization (PSO) is a parallel population-based computation technique proposed by Kennedy and Eberhart [7] in 1995, which is motivated by the behavior of organisms such as fish schooling and bird flocking. The global optimizing model proposed by Shi & Eberhart (1999) [8] is described as follows:

$$V_{id}(t+1) = W * V_{id}(t) + C_1 * R_1 * (P_{best}(t) - X_{id}(t)) + C_2 * R_2 * (G_{best}(t) - X_{id}(t)) \quad (1a)$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t) \quad (1b)$$

where V_{id} is the velocity for particle i; t is the number of iterations; X_{id} is the particle position; W is the inertial weight; C_1 and C_2 are the positive constant parameters; R_1 and R_2 are the random functions in the range [0,1]; P_{best} is the best position of the ith particle and G_{best} is the best position among all particles.

In general, the parameters W, C_1, C_2, R_1 and R_2 are the important factors to influence the convergence of the PSO. However, parameters R_1 and R_2 cannot guarantee the optimization's ergodicity entirely in phase space because they are absolutely random in the traditional PSO [9]. Therefore, chaotic mapping with certainty, ergodicity and the stochastic property is introduced into PSO to improve the global convergence. R_1 and R_2 are chosen as follows:

$$R_i(t+1) = 4.0 \times R_i(t) \times (1 - R_i(t)) \quad (2)$$

where $R_i(t) \in (0,1) \quad i = 1,2$.

The inertia weight W is very important for the convergence behavior of PSO. A suitable value usually provides a balance between global and local exploration abilities and consequently results in a better optimum solution. We use the following equation to adjust W to enable quick convergence:

$$W = W_{\max} - \frac{W_{\max} - W_{\min}}{k_{\max}} \times k \quad (3)$$

where $W_{\max}$ is the initial weight, $W_{\min}$ is the final weight, k is the current generation and $k_{\max}$ is the maximum number of generation.

Fitness is used to evaluate the performance of particles. In general, mapping an original objective function value to a fitness value. In PSO operation, the root mean squared error (*RMSE*) is selected as the fitness function:

$$Fit(W_{BP}, r) = \sqrt{1/n * \sum_{i=1}^{n} (t_i - o_i)^2} \tag{4}$$

where t_i is the real value, o_i is the predicted value, n is the number of samples, W_{BP} is the weight of the network, r is the number of neurons in the hidden layer.

The objective is to minimize *RMSE*, i.e. *Fit*. So the particle with the lowest fitness will be superior to other particles and should be reserved in search process.

2.3 Evolutionary Artificial Neural Network (EANN)

The generalization performance of ANN depends on a good setting of the parameters of the network. Therefore, it is crucial to properly set the parameters of ANN. PSO algorithm is an evolutionary computation technique and it combines local search and global search, possessing high search efficiency. In the research, EANN – a hybrid approach which combines PSO algorithm with ANN is proposed to automatically choose the parameters of ANN. The algorithm avoids jamming and improves the prediction precision. The flow of the algorithm is described as follows:

```
Begin
  Step1: Set parameters
  Step2: Learning and computation
    Initialize a swarm of particles;
    Input the sample sets;
    ANN model learns with the training sample set;
    Compute the fitness of each particle;
    Initialize G_best position and P_best position;
    While (maximum generation or precision is not met)
    Do {Generation: = Generation+1;
        Generate the next swarm;
        ANN model learns with the training sample set;
        Compute the fitness of each particle in the swarm;
        Update G_best of the swarm and P_best of each particle;}
  Step3: Prediction and output the results
    ANN model predicts testing sample set with the parame-
    ters obtained from step2 and output the results;
End
```

2.4 Approach Combining EANN and Mathematical Models (EANN-MM)

Traditional mathematical models for flow stress prediction actually are theory-experiment models; they only consider some main influencing factors. Therefore, the accuracy of the flow stress from the mathematical models is low.

EANN considers the influence of other factors such as chemical compositions besides considering the influence of strain, strain rate and temperature. Therefore, EANN is used to adjust the error of the flow stress. Mathematical models reflect the varying direction of flow stress and they are used to predict the main value of the flow stress. The prediction model is described as follows:

$$\sigma = \sigma_M + D_{EANN} \tag{5}$$

where σ_M is the output of mathematical model; D_{EANN} is the output of EANN and it is the difference value between the real value and the calculated value.

In the paper, the following equation is chosen as mathematical model of 45 steel:

$$\sigma_M = \sigma_0 \exp(a_1 + a_2 T)\left(\dot{\varepsilon}/\dot{\varepsilon}_0\right)^{a_3+a_4 T}\left[a_5(\varepsilon/\varepsilon_0)^{a_6} + (1-a_5)(\varepsilon/\varepsilon_0)\right] \tag{6}$$

where $T = (t + 273)/1000$ is temperature, K; σ_0 is standard flow stress, MPa and σ_0=134MPa; ε_0 is true strain when σ is equal to σ_0 , and $\varepsilon_0 = 0.4$; $\dot{\varepsilon}_0$ is strain rate when σ is equal to σ_0, and $\dot{\varepsilon}_0 = 10s^{-1}$; $a_1 \sim a_2$ are regression coefficients.

3 Computational Results

3.1 Sample Selection

The continuous compression tests for 45 steel are performed on a Gleeble 1500 Thermal Simulator. The specimen size is $\phi 8mm \times 15mm$. In the hot compression tests, flow stress is tested and the data are collected in the sample database. Finally, 180 data patterns are selected and they are randomly divided into two subsets: 'training' and 'testing' sets. 100 groups are used as training samples to train the network, 80 groups are used as testing samples to check the generalization performance of the network.

Normalizing the data before applying EANN-MM is very important. In the paper, each attribute is scaled by the following method:

$$x_i^{'} = \frac{(x_i - \mu)}{\rho} \tag{7}$$

where x_i is a certain attribute value, and μ is the mean value of the attribute, and ρ is the standard deviation.

3.2 Models and Parameters

The inputs of EANN-MM are chosen as follows: the contents of C, Si, Mn, P, S, Cr and Ni; strain, strain rate and temperature. C, Si, Mn, P, S, Cr and Ni are the chemical components. The output is the flow stress. The weights of the network and the number of neurons in hidden layer are obtained by CPSO algorithm. Through many

experiments, the parameters of EANN-MM is chosen as follows: swarm size: 50, maximal generation: 200, $W_{\max} = 1.2$, $W_{\min} = 0.4$, $C_1 = 2.0$, $C_2 = 2.0$.

In BPN-MM, a three-layer BP network is used as a benchmark. The network inputs and output are the same as EANN-MM. Through many experiments, the parameters of BPN-MM: maximal generation: 6000, hidden nodes: 18, momentum term: 0.85, learning rate: 0.01.

3.3 Results of Different Models

In the research, we also choose BPN as a benchmark to predict the flow stress. Through 20 runs, the best results and average results of *RMSE* and the best results of relative error rate of the testing set for each algorithm are shown in Table 1.

Table 1. Comparison results of the three methods

Model		EANN-MM	BPN-MM	BPN
RMSE	Average results	3.973	6.134	6.696
	Best results	3.296	5.328	5.749
Relative error rate	$\beta \le 0.01$	36.24%	22.06%	13.42%
	$0.01 < \beta \le 0.05$	54.88%	60.98%	59.05%
	$0.05 < \beta \le 0.10$	7.75%	12.31%	18.86%
	$\beta > 0.10$	1.13%	4.65%	8.67%

where $\beta = |t - o|/t$ is relative error rate, t is real value and o is predicted value.

Generalization performance is the most important factor to evaluate the performance of a model, and the accuracy and precision of the testing set exactly reflect generalization performance of the model. From Table 1, it can be observed that the *RMSE* obtained from EANN-MM is far smaller than that from other approaches, which indicates that prediction precision and generalization performance of EANN-MM outperform other methods. On the other hand, the *RMSE* acquired form BPN-MM is smaller than that acquired from BPN, which shows that mathematical models also play an important role in the flow stress prediction. For the testing data, i.e. the non-sampled data, the relative errors between real values and predicted values from EANN-MM are nearly all within 0.1. It can be seen that EANN-MM is able to predict the flow stress of 45 steel very accurately. So the EANN-MM is a feasible and promising technique in the prediction of flow stress.

4 Conclusions and Discussions

We have discussed a new hybrid approach combining physically based models and evolutionary back-propagation neural network for the flow stress prediction. The performance of the new approach is evaluated in comparison with the results obtained from other methods. The obtained results show that significant improvements have been obtained in the prediction precision of EANN-MM, especially, the generalization performance of this model is greatly improved.

The flow stress prediction of 45 steel is not the only application field of the proposed approach, there are a large number of research directions that can be considered as useful extensions of this research. Applying the approach to engineering practice should be a more significant subject. The further research can consider using the model to production practice and can also consider applying the model to other prediction fields.

Acknowledgements

The work is supported by National Natural Science Foundation of China (No: 60174009, 60574063).

References

1. Hatta, N., Kokado, J.I., Kikuchi, S., Takuda, H.: Modelling on Flow Stress of Plain Carbon Steel at Elevated Temperatures, Steel Res. 56 (11) (1985) 575-582
2. Laasraoui, A., Jonas, J.J.: Prediction of Flow Stresses at High Temperatures and Strain Rates, Metall. Trans. A. 22A (7) (1991) 1545-1558
3. Serajzadeh, S., Taheri, A.K.: Prediction of Flow Stress at Hot Working Condition. Mechanics Research Communications 30 (2003) 87-93
4. Rao, K.P., Prasad, Y.K.D.V.: Neural Network Approach to Flow Stress Evaluation in Hot Deformation. Journal of Material Processing Technology 53 (1995) 552-556
5. Madakasira, P.P., Ashok, K.L.: The Applicability of Neural Network Model to Predict Flow Stress for Carbon Steels. Journal of Materials Processing Technology 141 (2003) 219–227
6. Li, P., Xue, K.M., Lu, Y., Tan, J.R.: Neural Network Prediction of Flow Stress of Ti–15–3 Alloy under Hot Compression. Journal of Materials Processing Technology 148 (2004) 235–238
7. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. Proceeding of the 1995 IEEE International Conference on Neural Networks (1995) 1942-1948
8. Shi, Y., Eberhart, R.: Empirical Study of Particle Swarm Optimization. Proceedings of Congress on Evolutionary Computation, (1999) 1945-1950
9. Jiang, C.W., Etorre, B.: A Self-Adaptive Chaotic Particle Swarm Algorithm for Short Term Hydroelectric System Scheduling in Deregulated Environment. Energy Conversion and Management 46 (2005) 2689-2696

Meta-Learning Evolutionary Artificial Neural Network for Selecting Flexible Manufacturing Systems

Arijit Bhattacharya[1], Ajith Abraham[2], Crina Grosan[4], Pandian Vasant[3], and Sangyong Han[2]

[1] The Patent Office, Bouddhik Sampada Bhawan
CP-2, Sector V, Salt Lake , Kolkata 700 091, West Bengal, India
[2] School of Computer Science and Engineering, Chung-Ang University
221, Heukseok-dong, Dongjak-gu, Seoul 156-756, Korea
ajith.abraham@ieee.org, hansy@cau.ac.kr
[3] Universiti Teknologi Petronas, 31750 Tronoh, BSI, Perak DR, Malaysia
[4] Department of Computer Science, Babes-Bolyai University, Romania

Abstract. This paper proposes the application of Meta-Learning Evolutionary Artificial Neural Network (MLEANN) in selecting flexible manufacturing systems (FMS) from a group of candidate FMS's. First, multi-criteria decision-making (MCDM) methodology using an improved *S*-shaped membership function has been developed for finding out the 'best candidate FMS alternative' from a set of candidate-FMSs. The MCDM model trade-offs among various parameters, namely, design parameters, economic considerations, etc., affecting the FMS selection process in multi-criteria decision-making environment. Genetic algorithm is used to evolve the architecture and weights of the proposed neural network method. Further, a back-propagation (BP) algorithm is used as the local search algorithm. The selection of FMS is made according to the error output of the results found from the MCDM model.

1 Introduction

Flexible manufacturing system (FMS) is a set of integrated computer controlled automated material handling equipments and numerical controlled machine tools capable of processing a variety of part types. FMS provides a low inventory environment with unbalanced operations unique to the conventional production environment. Process design of FMS consists of a set of crucial decisions that are to be made carefully. It requires decision-making, e.g., selection of CNC machine tool, material handling system, product mix, etc. The selection of a FMS thus requires trading-off among the various parameters of the FMS alternatives. The selection parameters are conflicting in nature. Therefore, there is a need for sophisticated and applicable technique to help the decision-makers for selecting the proper FMS in a manufacturing organization. AHP is widely used for tackling FMS selection problems due to the concept's simplicity and efficiency 8. Ayag 3 uses the AHP technique for the evaluation of the hardware and software components for a targeted computer-aided system and uses a simulation generator integrated with the AHP in order to try the alternatives that are ranked by the AHP study, on a real-life product organization

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 891 – 897, 2006.

model of a company, until a model is found that provides the best performance values as determined by the company's management. In Abdi and Labib's 1 work, AHP is employed for structuring the decision-making process for selection of a manufacturing system among feasible alternatives based on the Reconfigurable Manufacturing System (RMS) study. Most of the published works in the field of MCDM application to select best possible FMS alternative from a group of candidate-FMSs contain data with hidden errors. Thus, an attempt has been made in this paper using Meta-Learning Evolutionary Artificial Neural Network (MLEANN) 2 approach to select the best possible FMS from a group of candidate-FMSs. The selection is made trading off the errors of output data while using the fuzzy-MCDM approach based on AHP.

2 FMS Selection Problem

The following nomenclature was used in the MCDM model for FMS selection problem. α: Level of satisfaction of DM, OFM: Objective Factor Measure, SFM: Subjective Factor Measure, OFC: Objective Factor Cost, SI: Selection Index, β: Fuzzy parameter which measures the degree of vagueness and $\beta = 0$ indicates crisp. Six different types of objective cost components have been identified for the selection problem. The total costs of each alternative are nothing but the OFCs of the FMSs as illustrated in Table 1. The task is to select best candidate-FMS among the five candidate-FMSs.

The subjective attributes influencing the selection of FMS are shown in Table 2. The study consists of five different attributes, namely, flexibility in pick-up and delivery, flexibility in conveying system, flexibility in automated storage and retrieval system, life expectancy / pay back period and tool magazine changing time. One may

Table 1. Cost factor components (in US \$ x 10^5)

FMS → / OFCs ↓	S_1	S_2	S_3	S_4	S_5
1. Cost of Acquisition	1.500	0.800	1.300	1.000	0.900
2. Cost of Installation	0.075	0.061	0.063	0.053	0.067
3. Cost of Commissioning	0.063	0.052	0.055	0.050	0.061
4. Cost of Training	0.041	0.043	0.046	0.042	0.040
5. Cost of Operation	0.500	0.405	0.420	0.470	0.430
6. Cost of Maintenance	0.060	0.070	0.065	0.054	0.052
Total Cost (OFC)	2.239	1.431	1.949	1.669	1.550
Objective Factor Measure (OFM_i)	0.154	0.241	0.177	0.206	0.222

Table 2. Attributes influencing the FMS selection problem

Factor I	Factor II	Factor III	Factor IV	Factor V
Flexibility in pick-up and delivery	Flexibility in conveying system	Flexibility in automated storage and retrieval system	Life expectancy / pay back period	Tool magazine changing time

consider other attributes appropriate to selection of FMS. The most important task for a decision-maker is the selection of the factors. Thorough representation of the problem indicating the overall goal, criteria, sub-criteria (if any) and alternatives in all levels maintaining the sensitivity to change in the elements is a vital issue. The number of criteria or alternatives in the proposed methodology should be reasonably small to allow consistent pair-wise comparisons. Matrix D is the decision matrix based on the judgemental values from different judges. Matrices A_1 to A_5 show comparisons of the weights for each of the attribute. Matrix G consolidates the results of the earlier tables in arriving at the composite weights, i.e., SFM_i values, of each of the alternatives. In the proposed methodology, the unit of OFC is US $, whereas OFM is a non-dimensional quantity. Correspondingly, the SI is also a non-dimensional quantity. Higher the SI values, the better would be the selection.

$$D = \begin{bmatrix} 1 & 5 & 3 & 4 & 5 \\ \frac{1}{5} & 1 & \frac{1}{3} & \frac{1}{2} & 1 \\ \frac{1}{3} & 3 & 1 & 3 & 5 \\ \frac{1}{4} & 2 & \frac{1}{3} & 1 & 3 \\ \frac{1}{5} & 1 & \frac{1}{5} & \frac{1}{3} & 1 \end{bmatrix} \quad A_1 = \begin{bmatrix} 1 & 3 & 2 & 5 & 4 \\ \frac{1}{3} & 1 & \frac{1}{3} & 5 & 2 \\ \frac{1}{2} & 3 & 1 & 4 & 3 \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{4} & 1 & \frac{1}{3} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{3} & 3 & 1 \end{bmatrix} \quad A_2 = \begin{bmatrix} 1 & 7 & 3 & 5 & 6 \\ \frac{1}{7} & 1 & \frac{1}{4} & \frac{1}{3} & \frac{1}{2} \\ \frac{1}{3} & 4 & 1 & 3 & 4 \\ \frac{1}{5} & 3 & \frac{1}{3} & 1 & 2 \\ \frac{1}{6} & 2 & \frac{1}{4} & \frac{1}{2} & 1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 1 & 4 & 1 & 3 & 7 \\ \frac{1}{4} & 1 & \frac{1}{4} & \frac{1}{2} & 5 \\ 1 & 4 & 1 & 2 & 7 \\ \frac{1}{3} & 2 & \frac{1}{2} & 1 & 3 \\ \frac{1}{7} & \frac{1}{5} & \frac{1}{7} & \frac{1}{3} & 1 \end{bmatrix} \quad A_4 = \begin{bmatrix} 1 & \frac{1}{3} & 5 & 3 & 6 \\ 3 & 1 & 5 & 7 & 6 \\ \frac{1}{5} & \frac{1}{5} & 1 & 2 & 3 \\ \frac{1}{3} & \frac{1}{7} & \frac{1}{2} & 1 & 2 \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{3} & \frac{1}{2} & 1 \end{bmatrix} \quad A_5 = \begin{bmatrix} 1 & \frac{1}{3} & 5 & 7 & 4 \\ 3 & 1 & 5 & 6 & 4 \\ \frac{1}{5} & \frac{1}{5} & 1 & 2 & \frac{1}{2} \\ \frac{1}{7} & \frac{1}{6} & \frac{1}{2} & 1 & \frac{1}{3} \\ \frac{1}{4} & \frac{1}{4} & 2 & 3 & 1 \end{bmatrix}$$

$$G = \begin{bmatrix} 0.471 & 0.076 & 0.259 & 0.131 & 0.063 \\ 0.408 & 0.512 & 0.366 & 0.273 & 0.305 \\ 0.159 & 0.051 & 0.104 & 0.501 & 0.458 \\ 0.279 & 0.246 & 0.338 & 0.103 & 0.074 \\ 0.050 & 0.117 & 0.151 & 0.075 & 0.047 \\ 0.103 & 0.075 & 0.040 & 0.047 & 0.116 \end{bmatrix}$$

The value of objective factor decision weight (α) lies between 0 and 1. For $\alpha = 0$, SI = SFM, i.e., selection is solely dependent on subjective factor measure values found from AHP and SFM values dominate over OFM values. There is no significance of considering the cost factor components for $\alpha = 0$. For α=1, SI = OFM, i.e., OFM values dominate over the SFM values and the FMS selection is dependent on OFM values only. For α =1, the cost factors get priority than the other factors. Keeping this in mind, the values of α are taken in between 0 and 1. The selection of the best candidate-FMS alternative is based on the error output of the results found from this MCDM model. Readers may refer to 56 for detailed description of the model and its analysis. The output data of MCDM is treated as input to MLEANN.

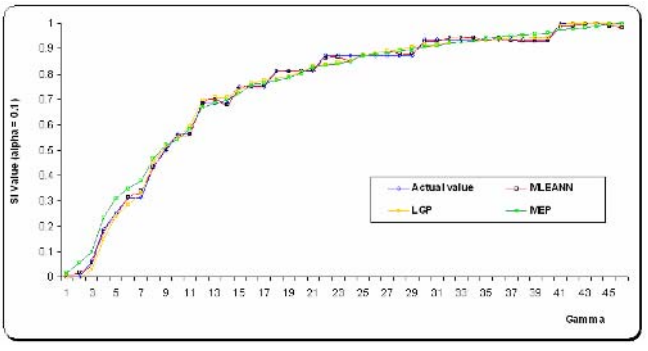


Fig. 1. Performance comparison for FMS_1 ($\alpha = 0.1$)

Table 3. FMS ranking

Candidate-FMS	SI_i values	Rank #
FMS_1	0.249	#1
FMS_2	0.224	#2
FMS_3	0.210	#3
FMS_4	0.155	#5
FMS_5	0.162	#4

3 Experiment Results

We used the Meta-Learning Evolutionary Artificial Neural Network (MLEANN) framework 2 and compared the performance with two genetic programming techniques namely Linear Genetic Programming (LGP) 4 and Multi Expression Programming (MEP) 7. Evolutionary algorithms are used to first locate a good region in the space and then a local search procedure is used to find a near optimal solution in this region. Back-propagation (BP) algorithm is used as the local search algorithm. The learning rate and momentum of the BP algorithm are adapted according to the problem. For performance comparison, we used the same set of training and test data that were used for experimentations with conventional design of neural networks. The parameters used in our experiments were set to be the same for all the problems. Fitness value is calculated based on the RMSE achieved on the test set. We considered the best-evolved neural network as the best individual of the last generation. All the genotypes were represented using real coding and the initial populations were randomly generated with the following settings. Population size = 30; maximum no. of generations = 25; number of hidden nodes = 5 - 9; activation functions = tanh, logistic, sigmoidal, tanh-sigmoidal and log-sigmoidal; output neuron = linear; training epochs = 500; initialization of weights = +/- 0.1; ranked based selection= 0.50; learning rate = 0.15-0.01; momentum = 0.15-0.01; elitism = 5 %; initial mutation rate = 0.70. Parameters used by LGP are population size = 100; mutation frequency = 50%; Crossover frequency = 95%; number of demes = 10; initial program size = 80 and maximum program size = 1000. Parameters used by

Table 4. RMSE and CC values for the different FMS using 4 different algorithms

	FMS_1	FMS_2	FMS_3	FMS_4	FMS_5
			MLEANN		
RMSE α=0.1	0.0082	0.0065	0.0067	0.0084	0.0045
RMSE α = 0.5	0.0065	0.0075	0.0056	0.0054	0.0063
RMSE α = 0.9	0.0056	0.0087	0.0067	0.0056	0.0056
CC α = 0.1	0.999	0.998	0.999	0.998	0.999
CC α = 0.5	0.999	0.999	0.998	0.999	0.998
CC α = 0.9	0.998	0.999	0.998	0.998	0.999
			ANN		
RMSE α = 0.1	0.022	0.0365	0.0267	0.0284	0.0245
RMSE α = 0.5	0.0265	0.0275	0.0256	0.0254	0.0263
RMSE α = 0.9	0.0256	0.0287	0.0267	0.0256	0.0256
CC α = 0.1	0.997	0.996	0.997	0.998	0.996
CC α = 0.5	0.997	0.996	0.997	0.998	0.998
CC α = 0.9	0.998	0.997	0.996	0.999	0.996
			MEP		
RMSE α = 0.1	0.0263	0.0196	0.0201	0.0154	0.0175
RMSE α = 0.5	0.0168	0.0199	0.0223	0.0185	0.019
RMSE α = 0.9	0.0236	0.0287	0.0176	0.0164	0.0177
CC α = 0.1	0.998	0.998	0.998	0.997	0.999
CC α = 0.5	0.996	0.998	0.999	0.998	0.998
CC α = 0.9	0.997	0.997	0.998	0.999	0.998
			LGP		
RMSE α = 0.1	0.1820	0.1965	0.1767	0.1840	0.1745
RMSE α = 0.5	0.0987	0.0295	0.0324	0.0354	0.02863
RMSE α = 0.9	0.0216	0.0248	0.0257	0.0216	0.0246
CC α = 0.1	0.998	0.995	0.996	0.996	0.998
CC α = 0.5	0.996	0.998	0.998	0.999	0.997
CC α = 0.9	0.998	0.998	0.996	0.999	0.996

MEP are population size = 50; number of mutations per chromosome = 3; crossover probability = 0.8; code length = 30; number of generations = 30 and tournament size = 4. FMS ranking is illustrated in Table 3. Empirical results (Root Mean Squared Error – RMSE, and Correlation Coefficient – CC) using the three methods and a direct back-propagation approach are illustrated in Table 4.

5 Conclusions

It is seen from the MCDM model combining both cardinal and ordinal factors for selecting FMS that at lower level-of-satisfaction (α) the chances of getting involved higher degree of fuzziness (β) increase. Therefore, a decision maker's (DM) level-of-satisfaction should be at least moderate in order to avoid higher degree of fuzziness while making any kind of decision using the MCDM model.

One underlying assumption of the MCDM methodology was that the selection is made under certainty of the information data. In reality, the information available is highly uncertain and sometimes may be under risk also. The fuzzy *S*-curve MF helps in reducing the level of uncertainty as validated further by introducing the MLEANN framework shown in Table 4. The RMSE and CC as compared and a trade off is made to select the error levels in the said MCDM model's decision. It is found that using the MLEANN framework the following decision depicted in Table 4 can be consolidated with DM's α value of $\alpha = 0.42$

Acknowledgements

This research was supported by the International Joint Research Grant of the IITA Institute of Information Technology Assessment) foreign professor invitation program of the MIC (Ministry of Information and Communication), Korea.

References

1. Abdi, M.R. and Labib, A.W.: A Design Strategy for Reconfigurable Manufacturing Systems (RMSs) Using Analytical Hierarchical Process (AHP): A Case Study. International Journal of Production Research 41(10) (2003) 2273–2299
2. Abraham A., Meta-Learning Evolutionary Artificial Neural Networks, Neurocomputing 56c (2004) 1-38
3. Ayag, Z.: An Analytic-Hierarchy-Process Based Simulation Model for Implementation and Analysis of Computer-Aided Systems. International Journal of Production Research 40(13)(2002) 3053–3073
4. Banzhaf. W., Nordin. P., Keller. E. R., Francone F. D.: Genetic Programming: An Introduction on The Automatic Evolution of Computer Programs and its Applications. Morgan Kaufmann Publishers, Inc., (1998)
5. Bhattacharya, A., Abraham, A., and Vasant, P.: FMS Selection Under Disparate Level-of-Satisfaction of Decision Maker Using Intelligent Fuzzy-MCDM Model, In: C. Kahraman (Ed.) Fuzzy Multi-Criteria Decision-Making Theory and Applications with Recent Developments, Kluwer Academic Publishers (2006)

6. Bhattacharya, A., Abraham, A., and Vasant, P.: Measurement of Level-of-Satisfaction of Decision Maker in Intelligent Fuzzy-MCDM Theory: A Generalised Approach, In: C. Kahraman (Ed.) Fuzzy Multi-Criteria Decision-Making Theory and Applications with Recent Developments, Kluwer Academic Publishers (2006)
7. Oltean, M., Grosan, C.: Evolving Evolutionary Algorithms using Multi Expression Programming, Proceedings of the 7th European Conference on Artificial Life, Dortmund, Germany (2003) 651-658
8. Saaty, T.L., How to Make a Decision: The Analytic Hierarchy Process, European Journal of Operational Research 48(1) (1990) 9–26

A Multi-Criteria Decision Making Procedure Based on Neural Networks for Kanban Allocation

Özlem Uzun Araz, Özgür Eski, and Ceyhun Araz

Dokuz Eylul University, Faculty of Engineering, Industrial Engineering Dept.,
35100 Bornova-Izmir, Turkey
ozlem.uzun@deu.edu.tr

Abstract. In this study, we proposed a methodology for determining optimal number of kanbans for each station in a JIT manufacturing system. In this methodology, a backpropagation neural network is used in order to generate simulation meta-models, and a multi-criteria decision making technique (TOPSIS) is employed in order to evaluate kanban combinations with respect to the relative importance of the performance measures. The proposed methodology is applied to a case problem and the results are presented. The results show that the methodology can solve this type of problems effectively and efficiently.

1 Introduction

Just-in-Time (JIT) is a management philosophy aimed at eliminating manufacturing wastes by producing only the right amount and combination of parts at the right place at the right time. One of the major elements of JIT philosophy is the kanban system. A kanban system is a decentralized system that harmoniously controls the production of the necessary products in the necessary quantities at the necessary time [1]. The inventory levels are determined by the total number of kanbans at each step of manufacturing. Hence, determining the appropriate number of kanbans at each workstation is a critical decision. There are several studies have been done to find the number of kanbans required in JIT system.

Most of the existing studies in the literature are modeled by using a mathematical programming (see e.g. [2], [3]), Markov chain (see e.g. [4], [5]) or simulation approaches. The problem of allocation of kanbans to the stations is complicated due to the variations in demand, processing times and product types. For that reason, simulation is used as a tool to analyze such complex problems.

Gupta and Gupta [6], and Savsar [7] presented simulation models to analyze JIT systems under various operating conditions. However, the use of simulation is time consuming and not practical due to the combinatorial nature of the problem. In order to overcome these limitations of simulation, regression meta-modeling approaches have been widely used. Aytug et al. [8] determined the number of kanbans in a JIT system using a simulation meta-modeling approach.

A neural network is a proven tool in providing excellent response predictions in many application areas and it outperforms regression analysis for a wide array of applications. Hurrion [9] developed a neural network meta-model to search for the

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 898 – 905, 2006.

optimal kanban combination for a two-station pull system. Savsar and Choueiki [10] extended the study of Hurrion and proposed a Generalized Systematic Procedure (GSP) that integrates experimental design concepts with simulation and neural network modeling for solving kanban allocation problems.

Generally, in simulation of manufacturing systems, more than one performance measures are considered, usually in conflict. If a decision maker were interested in optimizing all objectives simultaneously, an aggregated weighted function would have to be defined. Some researchers use cost function as aggregated weighted function in order to evaluate all possible kanban combinations with respect to the objective function (see e.g. [10]). However, in real life, construction of the cost function is a difficult task because of incommensurable nature of some performance measures such as average customer delay time, maximum customer delay time, the number of demand backordered etc.

In this study, we proposed, as the first time, a multi-criteria decision making (MCDM) procedure that uses neural network based meta-modeling for determining the number of kanbans. The function representing the relationships between the possible kanban combinations and simulation responses is generated from the back-propagation neural network meta-models. Instead of using an aggregated cost function, The Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), which is a well-known MCDM technique, is employed to evaluate kanban combinations with respect to the relative importance of the performance measures.

This paper is further organized as follows. Section 2 contains a description of TOPSIS method. Section 3 is devoted to explain the proposed methodology. A case problem is introduced in section 4. Section 5 provides an application of the proposed methodology to the case problem. Final section includes concluding remarks.

2 TOPSIS

TOPSIS [11] is one of the well-known MCDM techniques. The basic concept of this method is that the selected best alternative should have the shortest distance from the ideal solution and the farthest distance from the negative-ideal solution. The TOPSIS method consists of the following steps[12,13]:

Step 1: Normalization of performance measures. This step converts the various attribute dimensions into non-dimensional attributes. An element a_{ij} of the normalized decision matrix A is calculated as follows:

$$a_{ij} = \frac{x_{ij}}{\max_i a_{ij}} \quad i = 1, 2, ..., m. \quad j = 1, 2, ..., n. \tag{1}$$

Where, x_{ij} is the performance measure of the *i*th alternative in terms of the *j*th criterion.

Step 2: Determine the ideal and the negative-ideal solutions. The ideal a^M and the negative-ideal solutions a^m are defined as follows:

$$a^M = (a_1^M, \ldots, a_j^M, \ldots a_n^M) . \tag{2}$$

where a_j^M is the best value for the *j*th attribute among all available alternatives. The composite of all best attribute ratings attainable is the ideal solution, whereas the negative-ideal solution is composed of all worst attribute ratings attainable. The negative-ideal solution a^m is given as

$$a^m = (a_1{}^m, \ldots, a_j{}^m, \ldots a_n{}^M) . \tag{3}$$

where a_j^m is the worst value for the *j*th attribute among all alternatives.

Step 3: Calculate the separation measure. The weighted distances of each alternative from the ideal one (a^M) and the worst one (a^m) is then respectively, according to the chosen metric p:

$$d^M{}_p(a_i) = \left[\sum_j w_j^p \left| a^M{}_j - a_{ij} \right|^p \right]^{1/p} . \tag{4}$$

$$d^m{}_p(a_i) = \left[\sum_j w_j^p \left| a^m{}_j - a_{ij} \right|^p \right]^{1/p} . \tag{5}$$

where w_j is the weight of *j*th criteria.

Step 4: Calculate the relative closeness (similarity ratio) to the ideal solution. The relative closeness of *i*th alternative with respect to the ideal solution a^M is defined as follows:

$$D_p(a_i) = \frac{d^m{}_p(a_i)}{d^M{}_p(a_i) + d^m{}_p(a_i)} . \tag{6}$$

*Step 5: Rank the preference order***.** Rank the preference order according to the-descending order of similarity ratio $D_p(a_i)$.

3 Proposed Methodology for Determining Number of Kanbans

In this study, we proposed a design methodology incorporating BP neural network based meta-modeling and TOPSIS based multi-criteria decision making procedure for determining number of kanbans. The steps of methodology are given below.

Step 1. Selection of possible kanban combinations. First, subject to some system constraints such as maximum number of kanbans in a station, the set of all possible kanban combinations is identified. Selection of a limited set of combinations can be made by using randomized, one-factor-at-a-time or full factorial experimental design.

Step 2.Simulation modeling of manufacturing system. The second step of the methodology includes building a simulation model of JIT manufacturing system, setting the parameters of system such as arrival time and processing time distributions and defining the performance measures to be evaluated. Finally, the simulation model is run for each selected kanban combinations in step 1.

Step 3. Building, training and testing of neural network model. An ANN is built, trained and tested by using the selected set of kanban combinations as inputs and performance measures obtained by simulation as outputs.

Step 4. Full enumeration: The performance measures of all possible kanban combinations are obtained by using ANN model built in step 3.

Step 5. Evaluation of Kanban combinations by using a multi-criteria decision making approach. The performance measures conflicting in nature are evaluated by using TOPSIS approach, instead of using an aggregated cost function.

4 A Case Study for a JIT Manufacturing System

The manufacturing system under consideration is an assembly line constituted of three stations that process two types of products. End products are labeled as E and F and manufactured in three adjacent work-centers. Product E is also used to produce end product F. The dual-card kanban controlled system considered in this study is illustrated in Figure 1.

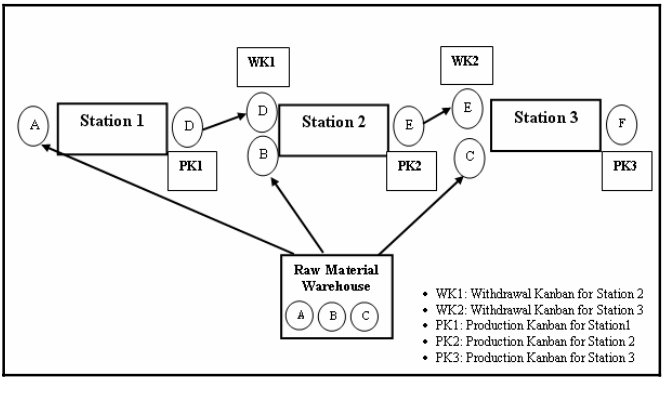


Fig. 1. 3-Station Dual-Card Kanban System

The time between two successive demands is randomly chosen according to an exponential distribution with a mean of 9.17. The product types are assigned from a discrete probability distribution for each arriving order with 40% type E and 60% type F. The first workstation uses raw material A and produces intermediate product D. The second workstation uses product D and raw material B to assemble product E. Finally, third station uses raw material C and product E to produce product F.

Containers are used to transport the products and one kanban is attached to each container. The transport times of containers are the same for each type of product and specific to each section of the line. The processing times and transport times are given in Table 1.

Problem assumptions are as follows.

- An infinite supply of raw materials A, B, C.
- Transfer time for raw materials and production kanbans are negligible.

- The kanbans are released as soon as their containers are empty.
- The number of production Kanban associated with intermediate product D is in range of 1-8 and the number of other production kanbans and all withdrawal kanbans are in range of 1-9.
- Due to the lack of production floor space following additional constraints must be hold: $WK1 + PK2 \leq 10$ and, $WK2 + PK3 \leq 10$.
- Acceptable customer delay time is in the range of 0- 480 min.

Table 1. The Distributions of Processing and Transportation Times

Processing Times (Minute)	
Station 1	Normal (8.85, 1.15)
Station 2	Normal (5.25, 0.98)
Station 3	Normal (7.27, 1.78)
Transportation Time (Minute)	
From Station 1 to Station 2	9.25
From Station 2 to Station 3	8.75

In this study, three important performance measures, namely average work-in process level (WIP), average customer delay time (CDT), number of containers, are considered.

In the next section, proposed methodology is applied to the JIT manufacturing system mentioned above.

5 Application of Proposed Methodology

Step 1. In this step, the set of all possible kanban combinations is identified considering some system constraints such as maximum number of kanbans in a station. The simulation run of each kanban combinations (for 10 replications) takes approximately 5-7 minutes of input/output and computation time. Running 16200 combinations requires 1400 to 1900 hours. A neural network based meta-model requires much less time than the related simulation model to generate the simulation outputs. Selection a set of combinations is made by using randomized experimental design. Out of 16200 possible combinations, 500 combinations are selected randomly.

Step 2. The simulation model of three station JIT production line is constructed with ARENA 2.2 Simulation Software and run for each combinations selected in Step 1. A total of 10 runs were made for each combination and the average WIP levels and customer delay times were computed.

Step 3. After obtaining the corresponding performance measures for 500 kanban combinations, a multilayer feed-forward neural network model is built as illustrated in Figure 2. The number of withdrawal and production kanbans in each station are introduced as input nodes. Output layer consists of two nodes that estimates the average WIP levels and average customer delay time.

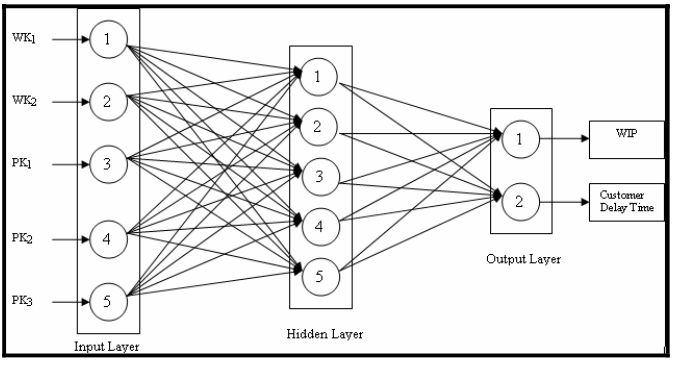


Fig. 2. Neural Network model of the JIT manufacturing system

The performance of neural network depends on several design parameters such as number of hidden layers, the number of nodes in the hidden layer, the type of transfer function, learning rate, momentum rate etc. In general, identification of these design parameters is done by trial and error [10]. Hence the design parameters are obtained by trial and error in our study. We trained the backpropagation neural network which has a single hidden layer with 5 nodes. Hyperbolic tangent transfer function is used and learning rate and momentum rate are chosen 0.1 and 0.5 respectively. For neural network model 400 and 40 out of 500 input-output pairs are randomly taken and used for training and cross validation. When the number of learning epochs is larger than 10.000 or the mean square error is less than 0.001 the learning process stops. Remaining 60 combinations are used for testing.

The mean square error is used as the measure of accuracy of the neural network. Table 2 shows some measure of accuracy such as mean square error; % mean absolute error and % root mean square error.

Table 2. Some measure of accuracy for WIP levels and Customer delay time (MSE: Mean Square Error; RMSE= Root Mean Square Error)

Error Measures	Customer Delay Time	WIP Level
% Abs Error	0.00886	0.02695
MSE	0.26369	0.16901
% RMSE	0.00126	0.00559

Step 4. The average WIP levels and average customer delay times for each kanban combination (except for the 500 combinations used for training and testing) are obtained by using the neural network model built in step 3.

Step 5. After obtaining customer delay time and WIP levels for all kanban combinations, TOPSIS multi-criteria decision making technique is used in order to choose optimum kanban combination considering the relative importance of the performance measures. It is known that, the customer delay time more than 480 min is not allowed.

Thus, 457 combinations violating this constraint are eliminated. Then TOPSIS is applied for remaining 15.743 kanban combinations. Three performance measures, the customer delay time, WIP levels and the number of containers, are considered as the decision criteria. All of these performance measures are wanted to be minimized.

First, all data are normalized between 0 and 1. Second, Ideal and anti-ideal points for all criteria are identified as shown in Table 3. Third, the weighted Euclidean distances $d^M{}_p(ai)$ and $d^m{}_p(a_i)$ to the ideal and anti-ideal points are computed from equations (4) and (5). At the fourth step, similarity ratio is calculated from equation (6). This ratio varies from $D_p(a^m) = 0$ for the anti-ideal point to $D_p(a^M) = 1$ for the ideal point. Finally, the ranking of the alternatives is obtained according to the descending order of similarity ratio $D_p(a_i)$.

Table 3. Ideal and anti-ideal points for all criteria

Criteria	Ideal Points	Anti-Ideal Points
Customer Delay time	0.0532	1
WIP	0.2067	1
Number of containers	0.2143	1

We assumed that customer delay time has the highest priority among all measures. In order to reflect the relative importance of performance measures for different decision makers (DM), different weight structures are generated. For given set of weights, the alternative with the highest D_p is chosen. Table 4 shows the alternative solutions according to DM's point of view.

As seen in Table 4, if customer delay time is used as a sole criterion for choice, the kanban combination (*3-1-8-7-9*) with minimum customer delay time is selected. It is interesting to note in Table 4 that the higher is the number of the production kanban at the second and third stations, the shorter is the customer delay time. This is probably because the second and third stations produce end products, and serve as a buffer in meeting the demand.

It is obvious that a decision maker can find different solutions by using different weight structure that reflects her/his preferences.

Table 4. Optimum solutions for different set of weights

Weights			Alternatives					Criteria		
CDT	WIP	# of con.	WK4	WK5	PK4	PK5	PK6	CDT	WIP	# of con.tainers
0.5	0.4	0.1	3	1	1	2	1	91.915	4.625	8
0.8	0.1	0.1	3	1	2	3	9	50.278	9.957	18
0.33	0.33	0.33	3	1	1	1	1	96.015	4.298	7
0.5	0.25	0.25	3	1	1	1	2	91.784	4.681	8
1	0	0	3	1	8	7	9	26.421	15.323	28

6 Conclusion

In this study, we proposed a methodology for determining optimal number of kanbans for each station in a JIT manufacturing system. In this methodology, a backpropagation neural network is used in order to generate simulation meta-models. In order to evaluate kanban combinations a multi-criteria decision making technique (TOPSIS) is employed. The proposed methodology can solve kanban allocation problem effectively and efficiently. The further research of this study should be investigation of applicability and effectiveness of the methodology on different industrial problems.

References

1. Ramanan, G.V., Rajendran, C.: Scheduling in Kanban-Controlled Flowshops to Minimise the Makespan of Containers. Int. J. Adv. Manuf. Technol. 21 (2003) 348–354.
2. Bard, J.F., Golony, B.: Determining the Number of Kanbans in a Multiproduct, Multistage Production System. Int. J. of Production Research 29 (1991) 881–895.
3. Mitwasi, M.G., Askin, R.G.: Production Planning for A Multi-item, Multi-stage Kanban System. Int. J. of Production Research 32 (1994) 1173–1195.
4. Wang, H., Wang, H.P.: Determining the Number of Kanbans: A Step toward Non-Stock-Production. Int. J. of Production Research 28 (1990) 2101–2115.
5. Askin, R.G., Mitwasi, M.G., Goldberg, J.B.: Determining the Number of Kanbans in Multi-Item Just-In-Time Systems. IIE Transactions 25 (1993) 89-97.
6. Gupta, Y.P., Gupta, M.C.: A System Dynamics Model for A Multi-Stage, Multi-Line Dual-Card JIT-Kanban System. Int. J. of Production Research 27 (1989) 309-352.
7. Savsar, M.: Effects of Kanban Withdrawal Policies and Other Factors on the Performance of JIT Systems: A Simulation Study. Int. J. of Production Research 34 (1996) 205-214.
8. Aytug, H., Dogan, C.A., Bezmez, G.: Determining the Number of Kanbans: A Simulation Metamodeling Approach. Simulation 67 (1996) 23-32.
9. Hurrion, R.D.: An Example of Simulation Optimization Using A Neural Network Metamodel: Finding the Optimum Number of Kanbans in Manufacturing System. Journal of the Operation Research Society 48 (1997) 1105-1112.
10. Savsar, M., Choueiki, M.H.: A Neural Network Procedure for Kanban Allocation in JIT Production Control Systems. Int. J. of Production Research 38 (2000) 3247-3265.
11. Hwang, C.L., Yoon, K.: Multiple Attribute Decision Making: Methods and Applications, A State-of-the-Art Survey. Springer-Verlag, Berlin (1981).
12. Pomerol, J.C., Romero, S.B.: Multicriterion Decision in Management: Principles and Practice. Kluwer Academic Publishers, USA (2000).
13. Triantaphyllou, E., Lin, C.T.: Development and evaluation of five fuzzy multi-attribute decision making methods. Int. J. of Approximate Reasoning, 14 (1996) 281-310.

On-Line Measurement of Production Plan Track Based on Extension Matter-Element Theory

Zhi-Lin Sheng[1], Song-Zheng Zhao[1], Xin-Zheng Qi[1], and Chen-Xi Wang[2]

[1] Northwestern Polytechnical University, Xi'an, 710072, China
wyszl@mail.wzyb.com.cn
[2] Chang'an University, Xi'an, 710064, China

Abstract. Based on the features of production plan and control in core enterprise under supply chain, according to extension matter-element theory, the paper presents an automatic on-line measuring method of distributed production plan track using the multi-sensor and a new extension measurement method which has two BP networks for realizing the compensation of the measurable matter-element and matter-element transform respectively and uses the D-S Evidence theory for matter-element focus to realize the machining accuracy data of a working procedure that has finished justly and the faults of the machine tools and cutting-tools etc. By the data and the faults, it will reschedule Job-Shop production plan, so as to realize the right time to finish the production plan and to supply data guarantee for the production plan and control in core enterprise under supply chain. The result presented shows that the method is feasible and efficient.

1 Introduction

Under supply chain, the production of enterprises depends on the real-time feedback of information. The real-time feedback of the information makes the enterprises eliminate the uncertainty and the effect of information lag to supply chain and assure relativity and consistent of production of fluctuation enterprises. This requires core enterprise to supervise roundly and intendance and to inspect in phase the production order during they implementing production plan and control under supply chain and to send all information to the information center that other enterprises can share. And all supply enterprises can real-time adjust their production plan and control according to the fluctuant information so as to realize the 5R production (Right Product, Right Quantity, Right Price, Right Time, and Right Place). Whereas Job-Shop production plan track is the important segment of real-time feedback of the information during course of the production plan and control, and real-time gaining of the data of Job-Shop production plan track is the bottleneck of real-time feedback of the information in supply chain system, because it is difficult to directly measure all of the information on-line [1][2].

The extension subject is a type science to study contradictive problems. The originality of the extension measurement is that the measurement problems are solved making use of the invert method of contradiction. It is a new method different from the traditional method. The purpose of the extension measurement is to realize the

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 906 – 913, 2006.

measurement of immeasurable information and to improve the measurement precision of the immeasurable information.

Artificial neural network system composed of mass processing units is non-linear, cosmically parallel processing system. The BP network has ability of the non-linear map, according to the map theorem. The three layers of BP network with one hidden layer can close in upon any mapping function to complete the mapping role given [3][4][5].

As D-S evidence theory has the characteristic of reasoning under uncertainty. It uses the collected information as evidences, and sets up a corresponding basic certainty factor in decision-making target sets, thus D-S evidence theory can use the special form of Dempster's Rule of Combination to yield the combined mass in the same decision-making frame.

Based on the analysis and the emerging technology, the distributed production plan track systems could be modernized. In this paper, we will introduce an on-line measuring method of the distributed production plan track using a high precision measuring method of immeasurable information based on extension matter-element theory. Following the infrastructure, the BP network is adopted to compensate the measurable matter-element. Then the fusion result is sent to the BP network to realize the matter-element transformation, and D-S evidence theory is used to take the matter-element focus and precision compensation.

2 Rationale of the System

2.1 The Rationale of the Extension Measurement

The basic principle of the extension measurement bases on the extension matter-element concept, its key point is measurable matter-element and immeasurable matter-element and their transform.

Matter-Elements: Assuming n matter-elements $R_i = (N, c_i, v_i)$, where N is object, c_i is the feature of the object and v_i is the feature value, if the feature value of the matter-element can be measured using a sensor that is existing now, the matter-element is called measurable matter-element. Reversely the feature value of the matter-element cannot be measured, and then the matter-element is called immeasurable matter-element. In the measuring course, the measured matter-element is named target matter-element. The target matter-element may be measurable matter-element or immeasurable matter-element. In the extension measurement, the target matter-element is generally immeasurable.

Matter-Element Transform: While solving the contradiction problems, by transforming the feature and feature value of the objects, the method of the incompatible problems becoming the compatible problems is called matter-element transform.

By the matter-elements transform between the measurable matter-element of the object $R_i^* = (N, c_i^*, v_i^*)$ and immeasurable matter-element of the object $R_j' = (N, c_j', v_j')$, the formula (1) can be obtained:

$$R_j^{'} = f(R_1^*, R_2^*, \cdots, R_n^*) \quad j = 1,2,\cdots,m \tag{1}$$

Where c_i^* and v_i^* are the true feature and the feature value of the i-th measurable matter-element, $c_j^{'}$ and $v_j^{'}$ are the true feature and the feature value of the j-th immeasurable matter-element. Passing through the transform above, the feature of the immeasurable matter-element can be expressed by the feature of the measurable matter-element.

Matter-Element Focus: By the noise-removing function of the feature value of the measurable matter-element, the method of calculating feature value is named as matter-element focus.

2.2 The Rationale of D-S Evidence Theory

The D-S evidence theory assumes that there is a fixed set of mutually exclusive and exhaustive elements called the environment and symbolized by the Greek letter Θ. Θ={θ1, θ2, · · ·, θn}, Θ is named frame of discernment too.

1. The mass function m: $2^{\Theta} \rightarrow [0,1]$ meet: $m(\phi) = 0$ and $\sum_{A \subset \Theta} m(A) = 1$, then m is regarded as the basic probability assignment or mass function in Θ.
2. If: $Bel(\phi) = 0$ and $Bel(\Theta) = 1$ and $\forall A_1, A_2, \cdots, A_n \subset \Theta$ (n is any natural number) meet: $Bel\left(\bigcup_{i=1}^{n} A_i\right) \geq \sum_{I \subset \{1,2,\cdots,n\}, I \neq \Phi} (-1)^{|I|+1} Bel\left(\bigcap_{i \in I} A_i\right)$, then the function $Bel : 2^{\Theta} \rightarrow [0,1]$ is called belief function, its expression is:

$$Bel(A) = \sum_{B \subset A} m(B) \tag{2}$$

Where m is the mass function in Θ

3. If: $Pl(\phi) = 0$ and $Pl(\Theta) = 1$ and $\forall A_1, A_2, \cdots, A_n \subset \Theta$ (n is any natural number) meet: $Pl\left(\bigcap_{i=1}^{n} A_i\right) \geq \sum_{I \subset \{1,2,\cdots,n\}, I \neq \Phi} (-1)^{|I|+1} Pl\left(\bigcup_{i \in I} A_i\right)$, then the function $Pl : 2^{\Theta} \rightarrow [0,1]$ is called plausibility function, its expression is:

$$Pl(A) = \sum_{B \cap A \neq \Phi} m(B) \tag{3}$$

Where m is the mass function in Θ

4. Assuming $e_1, e_2, \cdots, e_m$ as m evidences, for $e_j (j \leq m)$, if $m(\bullet / e_j)$ is the mass function in Θ and $P(e_j)$ is the probability, then:

$$m(A) = \sum_{j=1}^{m} m(A / e_j) P(e_j) \quad A \subset \Theta \tag{4}$$

Where m is the mass function in Θ. In fact, $[Bel(A), Pl(A)]$ and $[0, Bel(A)]$ separately express the uncertain and totally believable interval of the proposition A, and $[0, Pl(A)]$ expresses the implicit interval that A is true.

5. The evidences can be combined using the following special method of D-S theory: assuming m_1 and m_2 as two mass functions in Θ, for and

$$m(A) = \frac{1}{N} \sum_{E \cap F = A} m_1(E) m_2(F) \quad (A \neq \Phi) \tag{5}$$

Where $N = \sum_{E \cap F \neq \Phi} m_1(E) m_2(F)$ and $N > 0$, still where m is the mass function in Θ, its expression is $m = m_1 \oplus m_2$.

2.3 The Model of BP Network

Typical BP neural network consists of three layers as input layer, hidden layers and output layer. In the paper, also the BP used has three layers. The input layer has l_1 cells. The hidden layer has seven cells and the output layer has l_2 cells. Values in each cell and weights are all saved in a series of double style arrays. The core computations are enclosed in a multi-cycle and the inner cycle has executed a learning mainly including the upload process and the feedback process. In the upload process, the output values of each middle unit are computed as following:

$$\mathrm{M_i} = \frac{1}{1 + e^{-x}} \quad \mathrm{x} = \sum_j MW_{ij} X_{xj} \tag{6}$$

Here "MW_{ij}" is the input weights and "X_{xj}" is the input values. The output values of output units are computed in almost the same way. In computing the output layer error, the output errors, average errors and maximum errors of the output units are computed in terms of the object values and actual values being stored in three arrays respectively. The middle unit errors are computed in terms of the formulation:

$$ME_i = MD_i^{'} \times \sum_j MOW_{ij} \times OE_j \tag{7}$$

Here "$MD_i^{'}$" is the derivative of the middle units output values and $MD_i^{'} = MD_i \times (1 - MD_i)$, "$MOW_{ij}$" is the weights of the middle units to output units and "OE_j" is the output errors of the output units.

The weights adjusting method of the output layer and the middle layer are as following: first, using the "delta" rule:

$$OW_i = K \times OE_i \times P_i \tag{8}$$

Here K=0.09, "OE_i" is the output error and the "P_i" is the input value. Then using the "momentum" rule:

$$DW_i = DW_i + M \times LDW_i \tag{9}$$

Here M=0.09 and "LDW_i" is the value of last DW_i.

This paper uses the errors stored before to check whether the learning is successful when one learning cycle finishes. The learning is successful if the Maximum value of the maximum errors stored before less than a given error and to finish the learning.

3 The Measurement and Compensation System

In the measurement field of the distributed production plan track, the $f(*)$ in section 2.1 has high non-linear characteristic and it is very difficult to set the expression. Due to the character that includes environment changing dynamically, noise, jamming, error, and insecurity and so on, also it is very difficult to set the math model of the measurement using a traditional method. Whereas the BP network has the ability of the non-linear map, according to the map theorem, the three layers BP network with one hidden layer can close in upon any mapping function to complete the mapping role given. Also in order to predigest the complexity and improve the precision of measurement, in the paper, the new system structure model of extension measurement in Fig.1, which has two BP networks for realizing the compensation of the measurable matter-element and matter-element transform respectively and one D-S evidence theory for the matter-element focus. Where the sensor group of measurable

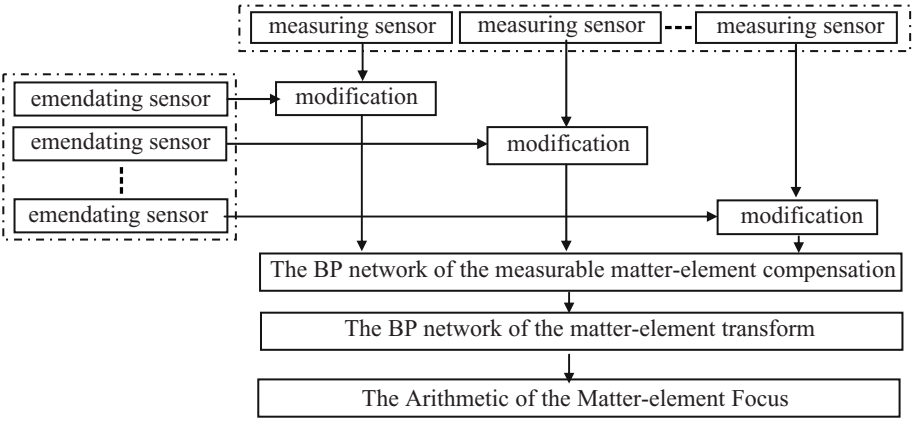


Fig. 1. The measurement and compensation system

matter-element completes the measurement of the measurable matter-element and the sensor group of measurable matter-element error compensation measures the environment factors that affect the measurement precision of the measurable matter-element separately. The BP network of measurable matter-element compensation realizes the map of the modification value of measurable matter-element. The BP network of the matter-element transform completes the fusion of information from revising sensors.

The function of the BP network of matter-element transform is to realize the transformation between the features of the immeasurable matter-element and the feature of the measurable matter-element. The D-S evidence theory is used to make the matter-element focus and precision compensation.

4 The Measurable Matter-Element Compensation

The precision of measurable matter-element has the disperse characteristic owing to both self-nonlinear and environment factor [6]. The affection of the environment factors use emendating sensors to modify, thus the output modification vector of sensor system of measurable matter-element composed of measuring sensors and emendating sensors V_i is

$$V_i = F + Y' \tag{10}$$

Where F is output true value vector of the measuring sensor system, Y' is output true value vector of the emendating sensor system. The affection of the self-nonlinear use the BP network in the system adopts the network in section 2.3, the output modification vector of the sensors system is regarded as the input of the BP network, by the nonlinear map of the BP network, the output of the BP network is the true value of the measurable matter-element. During the correcting course, let the modification vector $X = [X_1, X_2, \cdots, X_n]$ regard as the input vector of the BP network, and the output vector is the true value $Y = [Y_1, Y_2, \cdots, Y_m]$, the expression is

$$Y = f(X) = -Y' \tag{11}$$

Using a group of experiment data to build the training examples, by the learning, the non-linear mapping relation between vector X and Y can be realized. The course of the BP network is in section 2.3 so as to having the high precision and strong anti-jamming ability.

5 The Matter-Element Transform Network

The output of the modification units is regarded as the input vectors of the BP network of matter-element transform, and its output vectors are the feature vectors of immeasurable matter-element. Thus the BP network has $n+1$ cells in input layer and m cells in output layer. The course of the BP network is the same as the BP in section 2.3, where l_1 is $n+1$ and l_2 is m so as to completing the matter-element transforming.

6 The Arithmetic of the Matter-Element Focus

In the paper using the D-S evidence theory in section 2.2 completes the matter-element focus and precision compensation. The information V_i'' coming from the matter-element transform network is regarded as BPA of an evidence and the evidences are combined by one and one orderly, thus the fusion is done between two evidences every time. In other words, the BPA from the matter-element transform network at the same time as evidence is combined with the BPA from the fusion at the next time.

Assuming the BPA of the kth evidence having been combined is $m_i(k), i = 1,2,\cdots,v$, unknown degree is $\theta(k) = 1 - \sum_{i=1}^{v} m_i(k)$, the BPA of the (k+1)th evidence from the fusion in space domain at the k+1-th time is $m_i'(k+1), i = 1,2,\cdots v$, unknown degree is $\theta'(k+1)$, the new BPA can be calculated by (12):

$$m_i(k+1) = \frac{1}{N_{k+1}}(m_i(k)m_i'(k+1) + m_i(k)\theta'(k+1) + \theta(k)m_i'(k+1)) \tag{12}$$

Its unknown degree is:

$$e(k+1) = \frac{e(k)e'(k+1)}{N_{k+1}} \tag{13}$$

where $N_{k+1} = 1 - \sum_{i\neq 1} m_i(k)m_1'(k+1)$, and $\theta'(k+1) = 1 - \sum_{i=1}^{n} m_i'(k+1)$, $m_j(k)m_1'(k+1)(j \neq 1)$ is allotted to the empty set that delegates the conflict.

By the fusion, $m_i(*)$ of the last time of the fusion using the D-S evidence theory above is regarded as the V_i', that is the feature value of the ith immeasurable matter-element.

7 Results

In the paper, the measurement method is applied to measure the qualified rate of a set of work-pieces with 5 machining-procedures during the work-pieces being machined. There are both the measurable information and immeasurable information, for example, a majority of the shape error and position error of the holes can not be measured during the work-pieces being machined directly. In this experience, the measurement sensors are composed of laser sensors, optical fiber sensors, photo-electricity sensors,

laser slit scan sensors, pressure sensors etc. we measure in three phases and every phase has 50 work-pieces. The precision of the qualified rate is list in the table 1. From the table1 it can be seen the measuring precision is over 95%, even Proc. 1-2 and Proc. 4-5 is over 98%. Because the Proc. 3 is the holes, it is about over 95%.

Table 1. The precision of the qualified rate

	Proc. 1(%)	Proc. 2(%)	Proc. 3(%)	Proc. 4(%)	Proc. 5(%)
Phas. 1	99.3	98.3	95.1	98.9	99.2
Phas. 2	99.5	98.7	96.5	99.1	99.6
Phas. 3	99.2	98.1	97.2	98.8	99.5

8 Conclusion

Based on the introduction of conception and principle of the extension measurement, a new extension measuring method is presented to measure the distributed production plan track using a high precision measuring method of immeasurable information based on extension matter-element theory . Following the infrastructure, the BP network is adopted to compensate the measurable matter-element. Then the fusion result is sent to the BP network to realize the matter-element transformation, and D-S evidence theory is used to take the matter-element focus and precision compensation. The information from matter-element focus can be inputted into the Job-Shop production plan, so as to realize the right time to complete the production plan (JIT) and to supply data guarantee for the production plan and control of core enterprise under supply chain. The system is applied to the wu zhong instrument co., ltd, and makes production plan of the company's part products be completed at right time. The result shows that the method is feasible and efficient.

References

1. Comon, P.: Independent Component Analysis-A New Concept. Signal Process. 36(3) (1994) 287-314
2. Hyvarinen, A., Oja, E.: A Fast Fixed-point Algorithm for Independent Component Analysis. Neural Comput. 9(5) (1997) 1483-1492
3. Donoho, D. L., Johnstone, I. M.: Adapting to Unknown Smoothness Via Wavelet Shrinkage, J. Am. Statist Ass. 90(4) (1995) 1200-1224
4. Yang, P.H., Yang, J.: Architecture and Implementation of Knowledge Base in Web-based Agriculture Expert System. Vol. 3 ICEMI'2001, GuiLin China (2001) 1345-1349
5. Su, C., Cai, J.Q.: The Design and Implementation of Classified Expert Consultation System Development Tool. Engineering & Application. 35(5) (2001) 102-104
6. Passerini, K.: A Developmental Model for Distance Learning Using the Internet. J. Computer & Education. 34(1) (2000) 1-15

Modeling and Optimization of High-Technology Manufacturing Productivity

Sheng Xu[1,2], Hui-Fang Zhao[1], Zhao-Hua Sun[2], and Xiao-Hua Bao[3]

[1] School of Management, Hefei University of Technology,
230009 Hefei, China
xsbxh72@163.com, hfang_zhao@yahoo.com
[2] School of Management, Xi'an Jiaotong University,
710049 Xi'an, China
Zhsun@mail.xjtu.edu.cn
[3] School of Electrical Engineering and Automatization,
Hefei University of Technology,
230009 Hefei, China
xiaohua1972@163.com

Abstract. As more and more industries experience the globalization of business activities, measuring productivity performance has become an area of concern for companies and policy makers in Europe, the United States, Japan and so on. A novel way about nonlinear regression modeling of high-technology manufacturing (HTM) productivity with the support vector machines (SVM) is presented in this paper. Optimization of labor productivity (LP) is also presented in this paper, which is based on chaos and uses the SVM regression model as the objective function.

1 Introduction

In recent years, computing high-technology manufacturing (HTM) productivity level and growth rates, as well as analyzing productivity determinants, has gained a renewed interest among both growth economists and trade economists. HTM productivity is perceived as being at the core of observed divergences among countries in the areas of long-run growth and increase in standard of living. In the growing accounting literature, manufacturing productivity growth accounts for up to half of GDP growth rates in some countries. In the empirical literature on HTM productivity determinants, an alternative approach to physical output is to use value-added [1], [2]. Openness to trade, human capital and technology innovation activity are found to have very strong and robust impacts on HTM labor productivity, along with other factors as financial development, infrastructure development and quality [3], [4]. For the evaluation of productivity performance, traditional regression models are being widely used as plural linear model, Cobb-Douglas functional model [5], [6].

The SVM is grounded in the framework of statistical learning theory, which has been developed by Vapnik and co-works [7], [8]. The theory of SVM is based on the idea of structural risk minimization (SRM). Compared with the Artificial Neural Networks (ANN), it provides high generalization ability and overcomes the overfitting problem.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 914 – 920, 2006.

Chaos optimization algorithm is a novel stochastic search algorithm [11], [13]. It is a self-moving form of non-linear dynamics system. The stochastic property shown by chaos is the intrinsic one of the systems. The movement of chaos possesses ergodic property, and within a certain scope, can experience any state according to its own regularity.

In the modeling of HTM, traditional models like plural linear model, Cobb-Douglas functional model and gray models can illustrate the relation between the explaining variables and the explained variables. However, these models are flawed in two aspects. For one thing, they fail to eliminate the self-relation between the explaining variables themselves. For the other, they would not help in finding the optimal ratio relation between the explaining variable and the explained variables. This paper focuses on the reality of the HTM in China and collects data in a period as long as 15 years. It is with SVM and chaos that good results can be achieved with efficiency in solving the relation between the explaining variables and the explained variables and finding out the best ratio relation between them.

2 HTM Productivity

2.1 HTM LP

An alternative approach to physical output is to use value added. Value added measures how much additional value is created in the market-place. Value added per unit of labor is a commonly used measure of HTM productivity at the aggregate level [6]. Labor productivity is given by the following ratio:

$$LP = \frac{VA}{L} \tag{1}$$

Where LP is labor productivity for HTM, VA is value added for HTM, L is the quantity of HTM labor.

2.2 HTM LP Growths

In the empirical literature on HTM productivity determinants, openness to trade, human capital and technology innovation activity are found to have very strong and robust impacts on HTM labor productivity, along with other factors such as financial development and infrastructure development and quality.

With the globalization of the world economy, the export volume of HTM in every country is increasing year by year. The channels through which openness to trade affects HTM productivity are numerous, and are indeed quite well documented. Some studies based on endogenous growth theory focus on the importance of knowledge spillovers due to trade and find that they positively affect income convergence and economic growth during transition period in the long run. As far as China is concerned, in the paper, the variable of export per capita and the export of new products are used to illustrate the degree of trade openness in HTM and test their influences on HTM labor productivity through our model.

The importance of human capital on output and productivity growth has been the main focus of HTM productivity growth accounting regressions. In addition, the new growth theories indeed set the ground for knowledge accumulation and education as giving an impetus to growth. According to the situation in China, the average number of employees, the number of people in engineering and technological activities and the full-time equivalent of R&D activities are used to describe the human capital in HTM, test their influences on HTM productivity and then optimize them.

Since HTM technology innovation decides the long-term increase of economy and the potential in productivity increase. It was considered by the world Economy Forum to be as important as banking factors and the internationalization. Disaggregated data will be used to build technology innovation activity variables at HTM. Patent is a very important index that reflects the technology innovation activity of HTM and many countries regard the output of patents as measurement of the technology innovation activity of HTM. In this paper, patent is taken into consideration. As for China, the following variables are used to explain the technology innovation activity of HTM, that is, the internal expense of R&D funds, the accumulation of funds for scientific and technological activities, the three expenses (the total amount of the expense of technological reformation funds, the expense of technological introduction and the expense of digestion and absorption), the number of patent applied, the number of patents owned, profit and sales income.

3 SVM and SVM Regression

Support vector machines (SVM) and kernel methods (KM) have become, in the last few years, one of the most popular approaches to learning from examples with many potential applications in science and engineering. As a learning method, it is often used to train and design radial basis function (RBF) networks. Given a set of examples $\{(x_i,y_i), x \in R^n, y \in R, i=1,\cdots, N\}$,the SVM learning method in its basic form creates an approximation function $f(x) = b + \sum_{j=1}^{m} y_j \cdot \alpha_j \cdot K(x_j, x)$ with $y \approx f(x)$ for regression. For that purpose, a subset of support vectors $\{x_j, j = 1,...,m\} \subset \{x_i, i = 1,...,N\}$ is determined, the kernel function K is chosen, and the parameters $b,\alpha_j,j=1,\ldots,m$ are estimated.

KM is method that use kernels of the form $K(x_1, x_2) = \phi(x_1) \cdot \phi(x_2)$, is an inner product and Φ is in general a nonlinear mapping from input space *X* onto feature space *Z*. The symmetry of the inner product determines the symmetric of the kernel. The necessary and sufficient conditions for a symmetric function to be a kernel is to be positive definite, thus statistically seen, kernels are covariances. In practice, the kernel function *K* is directly defined. Φ and the feature space Z are implicitly derived from its definition. Kernel substitution of the inner product can be applied for generating SVMs for classification based on margin maximization.

In the ε-SVM regression [9], [10], the goal is to find a function *f(x)* that has at most ε deviation from the actually obtained targets y_i for all the training data, and at the same time, is as flat as possible. The ε-insensitive loss function reads as follows:

$$e(f(x)-y)=\begin{cases}0, \mid f(x)-y \mid \leq \varepsilon \\ \mid f(x)-y \mid -\varepsilon, otherwise\end{cases} \tag{2}$$

To make the SVM regression nonlinear, this could be achieved by simply mapping the training patterns by $\Phi: R^n \rightarrow F$ into some high dimensional feature space F. Suppose $f(x)$ takes the following form:

$$f(x)=w \cdot x+b \tag{3}$$

A best fitting function:

$$f(x)=(w^{'} \cdot \phi(x))+b \tag{4}$$

is estimated in feature space F, where "." denotes the dot product in the feature space F. Flatness in the case of (4) means that one can seek small $w^{'}$. Formally this problem can be written as a convex optimization. By constructing the Lagrangian function, the dual problem can be given as follows:

$$\begin{aligned} \max imize \quad & -\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)(x_i \cdot x_j) \\ & +\sum_{i=1}^{l} y_i(\alpha_i-\alpha_i^*)-\varepsilon\sum_{i=1}^{l}(\alpha_i+\alpha_i^*) \\ subject\ to \quad & \sum_{i=1}^{l}(\alpha_i-\alpha_i^*)=0 \\ & 0 \leq \alpha_i, \alpha_i^* \leq C \end{aligned} \tag{5}$$

Where α_i and α_i^* are the Lagrange multiplier coefficients for the ith training example of regression, and obtained by solving the dual optimization problem in support vector learning [7], [8]. The non-negative coefficients α and α^* are bounded by a user-specified constant C. The training example for which $\alpha \neq \alpha^*$ is corresponded to the support vectors.

At the optimal solution from (5), the regression function takes the following form:

$$f(x)=\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)K(x_i \cdot x_j)+b \tag{6}$$

Where $K(.,.)$ is a kernel function, where b is found by the Karush-Kuhn-Tucker conditions at optimality.

4 Application to HTM Modeling

To study the determinants of productivity growth in china, we should estimate the following equations:

$$LP=f(x_i), i=1,2,...,12 \tag{7}$$

A number of crucial parameters, which mainly determine the *LP*, are considered. Where:

LP is labor HTM productivity□and unit of it is 100 million Yuan per one people;
x_1 is the variable of export per capita, and unit of it is 100 million Yuan per one people;
x_2 is the export of new products, and unit of it is ten thousand Yuan;
x_3 is the average number of employees, and unit of it is one people;
x_4 is the number of people in engineering and technological activities, and unit of it is one people;
x_5 is the full-time equivalent of R&D activities, and unit of it is one people per one year;
x_6 is the internal expense of R&D funds, and unit of it is ten thousand Yuan;
x_7 is the accumulation of funds for scientific and technological activities, and unit of it is ten thousand Yuan;
x_8 is the three expenses, and unit of it is ten thousand Yuan;
x_9 is the number of patent applied, and unit of it is item;
x_{10} is the number of patents owned, and unit of it is item;
x_{11} is the profit, and unit of it is 100 million Yuan;
x_{12} is the sales income, and unit of it is 100 million Yuan.

x_1 and x_2 indicate the influence of openness to trade on *LP*; x_3, x_4 and x_5 indicate the influence of human capital on *LP*; x_6, x_7, x_8.x_9, x_{10}, x_{11} and x_{12} indicate the influence of technology innovation activity on *LP*.

5 Optimization with Chaos

The optimum problem is a high nonlinear, multi-parameter, multi-extreme and non-differentiable one. Some optimization algorithm can be used here, such as genetic algorithm (GA), chaos and so on. But the GA optimum results are a solution set, and a set of appropriate parameters must be selected artificially. In order to reduce the limitation of the artificially selection of GA and get the objective global solution, chaotic search algorithm to optimize is presented.

Chaos variables are usually generated by the well known Logistic map [12], [13]. The Logistic map is a one-dimensional quadratic map defined by the follows:

$$f(x) = ux(1-x) \tag{8}$$

Where u is a control parameter and $x \in [0,1]$.

The solution of the equation exhibits a rich variety of behaviours. For $u>3.75$, system (8) generates chaotic evolutions.

The objective function offered on-line by SVM model during the optimal calculation. By the global search and the local search, the optimum parameter collection is gained. Then, results of the chaotic search algorithm optimum are found, which are shown in Table 1.

Table 1. Compared results

parameters	bad samples	good samples	optimal results
x_1	0.000106	0.000543	0.000557
x_2	2945052	25649389	27658653.17
x_3	70774100	48838300	47695600.55
x_4	2087839	2378297	2379238.33
x_5	227359	430179	430026.76
x_6	781424	6784183	6875265.42
x_7	3878636	14816338	14823567.13
x_8	13787756	21430408	15658792.58
x_9	3343	29810	29821
x_{10}	2189	14654	14623
x_{11}	1225	6165	6063.57
x_{12}	46207	124035	124165.28
LP	0.000173	0.000697	0.000712

6 Conclusions

In the modeling of HTM, traditional models fail to eliminate the self-relation between the explaining variables themselves and they would not help in finding the optimal ratio relation between the explaining variable and the explained variables. This paper focuses on the reality of the HTM in China and collects data in a period as long as 15 years. In this paper, the support vector machine for nonlinear regression is presented. It is shown that the optimization approach based on chaos is effective and feasible. Results of the optimization algorithm are found.

Acknowledgements

The authors would like to thank National Natural Science Foundation of China (70433003), Societal Science Foundation in Anhui, China (AHSK03- 04D37), and Chinese college special fund for doctor subject (20040698027).

References

1. Garcia, P., Peter, F. K., Tilton, J. E.: Labor Productivity and Comparative Advantage in Mining: the Copper Industry in Chile. Resources Policy 27 (2) (2001) 97 - 105
2. Kuby, M., Xie, Z.: The Effect of Restructuring on US Coal Mining Labor Productivity, 1980–1995. Energy 26 (11) (2001) 1015 - 1030
3. Rho, B. H., Yu, Y. K.: A Comparative Study on the Structural Relationships of Manufacturing Practices, Lead Time and Productivity in Japan and Korea. Journal of Operations Management 16 (2) (1998) 257 - 270
4. Fagerberg, J.: Technological Progress, Structural Change and Productivity Growth: a Comparative Study. Structural Change and Economic Dynamics 11 (4) (2000) 393 - 411

5. Mohamed, S., Srinavin, K.: Forecasting Labor Productivity Changes in Construction Using the PMV Index. International Journal of Industrial Ergonomics 35 (4) (2005) 345 - 351
6. Mukhopadhyay, T., Lerch, F. J., Mangal, V.: Assessing the Impact of Information Technology on Labor Productivity: a Field Study. Decision Support Systems 19 (2) (1997) 109 - 122
7. Vapnik, V. N.: The Nature of Statistical Learning Theory, Springer, New York, (1995)
8. Vapnik, V. N.: An Overview of Statistical Learning Theory, IEEE Trans. Neural Networks 10 (5) (1999) 988 - 999
9. Gonzalez-Castano, F. J., Garcia-Palomares, U. M., Meyer, Robert R.: Projection Support Vector Machine Generators. Machine Learning 540 (1) (2004) 33–44
10. Fung, G. M., Mangasarian, O. L.: Multicategory Proximal Support Vector Machine Classifiers. Machine Learning 59 (1) (2005) 77– 97
11. Wang, L., Smith, K.: On Chaotic Simulated Annealing. Neural Networks 9 (4) (1998) 716 –718
12. Varghese, M., Fuchs, A., Mukundan, R.: Chaotic Zero Dynamics in Kinematically Redundant Robots. Aerospace and Electronic Systems 27(5) (1991) 784 – 796
13. Cappetta, L., Feo, M., Fiumara, V., Pierro, V., Pinto, M.: Electromagnetic Chaos in Mode-stirred Reverberation Enclosures. Electromagnetic Compatibility 40 (3) (1998) 185 – 192

Scheduling of Re-entrant Lines with Neuro-Dynamic Programming Based on a New Evaluating Criterion

Ying Wang[1], Huiyu Jin[2], Shunzhi Zhu[1], and Maoqing Li[1]

[1] System and Control Research Center, Xiamen University, Xiamen 361005, China
[2] Department of Automation, University of Science and Technology of China, Hefei 230027, China
wangying@xmu.edu.cn

Abstract. Scheduling of re-entrant lines is very important for manufacturing systems. For some dynamic scheduling methodologies it is necessary to model a production system with finite-state discrete-time Markov process. However, proper state cannot be found as absorbing state of Markov process when general Mean Output Rate is employed as an evaluating criterion. Mean-Output-parts Number Before First Block is presented to be a new evaluating criterion in this paper to evaluate scheduling policies for Closed Re-entrant Lines(CRL). Simulations of four static scheduling policies verify the new criterion. In order to apply a Neuro-Dynamic Programming (NDP) method to scheduling of a CRL, cost-to-go value function and transition cost function are presented as new forms under the new criterion. In addition, the policy obtained in a less-number parts system by the NDP is applied in a more-number parts system directly, whose results are satisfactory.

1 Introduction

Re-entrant lines are introduced in 1993 by Kumar[1], which are a third class of manufacturing systems after the Job shops and Flow shops, which represent semiconductor and thin film manufacturing systems. Its distinguishing feature is that parts visit some machine more than once, thus parts at different stages of their life may be in contention for service at the same machine, which gives rise to problems of machine scheduling.

Research focus on real-time scheduling policies, including Shortest Remaining Processing Time (SRPT), First Buffer First Serve (FBFS), Last Buffer First Serve (LBFS), Workload Balancing (WB), Fluctuation Smoothing Policy for Variance of Cycle-time (FSVCT) [2], Fluctuation Smoothing Mean Cycle-time (FSMCT) [3]. The research focus on Closed Re-entrant Line(CRL) in this paper, where a new part enter the re-entrant line the moment that a part is finished and leaves the lines, so number of parts in lines remain invariant. Usually, Mean Output Rate is used to evaluate scheduling policies in the CRL [1]. However, for some dynamic scheduling methodologies, it is necessary to model the production system with finite-state discrete-time Markov process, and proper state cannot be found as absorbing state when Mean Output Rate is employed as the evaluating criterion. In this paper, Mean-Output-parts Number Before First Block is presented to be a new evaluating criterion to evaluate scheduling policies for the CRL. Simulations of four static scheduling policies verify the new

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 921 – 926, 2006.

criterion. In order to apply a Neuro-Dynamic Programming (NDP) methodology to scheduling of the CRL, cost-to-go value function and transition cost function are presented as new forms under the new criterion. Furthermore the policy obtained in a less-number parts system is applied to a more-number parts system directly, whose results are satisfactory.

2 Description of a Re-entrant Line

As defined by Kumar [1], consider a set $\{1,2,\cdots,s\}$of s "service centers". Each service center $\sigma \in \{1,2,\cdots,s\}$ consists of M_σ identical machines. Parts of a certain type enter at a service center $\sigma(l) \in \{1,2,\cdots,s\}$, where they stored in buffer b_l. Let buffer b_L at service center $\sigma(L)$ be the last buffer visited. The sequence $\sigma(1),\cdots,\sigma(L)$ is the route of parts. We shall allow for the possibility that $\sigma(i)=\sigma(j)$ for some stages $i \neq j$. A new part enters the line as soon as a part is finished and leaves the line. So the number of parts in the system is constant. A two-Center four-Buffer (2-C 4-F) CRL is shown in Fig.1.

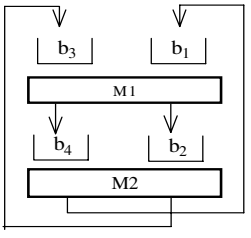


Fig. 1. A two-center four-buffer CRL

Let us assume that: 1) only a type of part is produced in line; 2) route is determined and remains unchanged; 3) $M_\sigma = 1$; 4) one service center can only serve a part at the same time; 5) once a part is processed, service cannot be interrupted; 6) switch time is zero; 7) capacity of all buffers is infinite; 8) service time in stage l are exponentially distributed with mean $\frac{1}{\mu_l} = m_l < \infty$; 9) nonidling, if any buffer at a machine is non-empty, then the machine can not stay idle[4,5].

3 A New Evaluation Criterion

For the CRL, normal evaluation criterion to evaluate scheduling policies is Mean Output Rate:

$\lambda^u(X) = \lim_{T\to\infty} \frac{E(D^X(T))}{T}$, which is output of system when initial state is X and policy is u, where $D^X(T)$ is the number of output parts within time T.

However, for some dynamic scheduling methodologies, it is necessary to model the production system with finite-state discrete-time Markov process, but proper state cannot be found as absorbing state of Markov process when Mean Output Rate is employed as the evaluating criterion. A new evaluation criterion is presented for the CRL. First, let's introduce the concept "block".

Definition: when one service center is idle, the system is in **block**.

A new criterion: Mean-Output-parts Number Before First Block can evaluate scheduling policies effectively.

Because of the nonidling policy, in order to increase the Mean Output Rate, the policy should make service centers in nonidling state as long as possible. In another word, it is expected to increase the mean usage μ of service centers, because blocks result in low mean usage directly. Take a 2-C 4-F CRL as an example, when block occurs, there must be a machine with a lot of parts piling in its one buffer waiting for service, which is the bottleneck of throughput, while another machine is idle because no parts waiting for service, which results in low mean usage directly. Extremely, if no block occurs, the usage of machine should be 100%. Longer no blocks occur, higher the machines usage is. So, Mean-Output-parts Number Before First Block and Mean Output Rate change with same tendency. A 2-C 4-F CRL is simulated to be scheduled by LBFS, FBFS, WB and UNWB static priority policies. Mean-Output-parts Number Before First Block and Mean Output Rate are worked out respectively.

[Simulation 1]. A 2-C 4-F CRL is shown as Fig. 1, Service time in four buffers are exponentially distributed with mean $\left[\frac{21}{76} \quad \frac{6}{76} \quad \frac{7}{76} \quad \frac{42}{76}\right]$. The parts number in closed system is N=20, initial state is that there are 5 parts in each buffer. The four static priority policies for this system are:

$$LBFS \quad O=\begin{bmatrix} b_3 & b_1 \\ b_4 & b_2 \end{bmatrix} \quad FBFS \quad O=\begin{bmatrix} b_1 & b_3 \\ b_2 & b_4 \end{bmatrix} \quad WBAL \quad O=\begin{bmatrix} b_1 & b_3 \\ b_4 & b_2 \end{bmatrix} \quad UNWB \quad O=\begin{bmatrix} b_3 & b_1 \\ b_2 & b_4 \end{bmatrix}$$

The row j in matrix O means the priority of buffers in service center j,for example, $LBFS \quad O=\begin{bmatrix} b_3 & b_1 \\ b_4 & b_2 \end{bmatrix}$ means buffer b_3 has higher priority to b_1 in service center M_1, buffer b_4 has higher priority to b_2 in service center M_2. The simulation results are shown in table 1, the Mean-Output-parts Number Before First Block and the Mean Output Rate have same varying tendency under the four policies.

Table 1. Simulation results of four policies (N=20) (95% confidence interval)

Policy	Mean Output Rate	Mean-Output-parts Number Before First Block
WB	0.2301±0.0012	43.2570±1.5214
FBFS	0.2206±0.0010	14.2872±0.7342
LBFS	0.2188±0.0008	13.6789±0.3975
UNWB	0.2061±0.0008	9.3420±0.4211

4 Policy Optimization with NDP Based on the New Criterion

The static priority scheduling policies are easy to apply and have clear performance bounds but are lack of flexibility. Under static priority policies, parts have the tendency to pile in buffers with the lowest priority and the long mean service time. Different priority scheduling polices just change the buffer parts piling in.

Neuro-Dynamic Programming (NDP) [6] is a kind of optimization methodology, which simplifies description and search of state space and solves the "curse of dimensionality" effectively by function approximation. It is appropriate to solve problems with large state space and high complexity. The CRL just have these features. Furthermore, approximation functions learned in a small-scale system by NDP can be expanded easily to a same-structure large-scale system.

NDP regard the whole production process as a finite-state discrete-time Markov Decision Process (MDP). For a continuous-time CRL, mean service time of each buffer should be normalized, that is, $\sum_{l=1}^{L}\mu_l = 1$. Then sample in time domain, set a clock for each buffer b_l. When the clock returns to zero, a random time exponentially distributed with mean μ_l is created, count down again, once some clock returns to zero, system is sampled again, then a finite-state discrete-time Markov process is obtained, which have steady distribution with original process [7].

The parts number in each buffer constitutes state variable $X = [x_1, \quad x_2, \quad \cdots \quad x_L]^T$, where x_l is the parts number in buffer b_l, decision variable is $U = [u_1, \quad u_2, \quad \cdots \quad u_L]^T$, where $u_l = \begin{cases} 1 & parts\ in\ buffer\ b_l\ in\ service \\ 0 & others \end{cases}$, the transition probability of the Markov process is

$$\begin{cases} P(X,U,X-E_l+E_{l+1}) = \mu_l u_l \quad 1 \le l \le L-1 \\ P(X,U,X-E_L+E_1) = \mu_L u_L \\ P(X,U,X) = 1-\sum_{l=1}^{L}\mu_l u_l \end{cases}$$

where E_l is a column vector, whose row l is 1, other rows are 0.

According to the new evaluation criterion, transition cost function and cost-to-go function can be expressed as corresponding new forms. The CRL can be modeled as a MDP with a absorb state, which is the state that the line is in block.

For a 2-C 4-F CRL, the optimal cost-to-go function and Bellman Equation are:

$$J^*(X) = -\max E\left\{\begin{array}{l}\text{Output parts Number Before First Block} \\ \\ \text{when initial state is X}\end{array}\right\}$$

$$\begin{cases} J^*(X) = \min\limits_{U} E\{g(X,U,Y) + J^*(Y)\} \\ U^*(X) = \arg\min\limits_{U} E\{g(X,U,Y) + J^*(Y)\} \end{cases}$$

"greedy policy" is adopted in order to minimize the cost-to-go function.

Generally, the transition cost function of NDP is selected as G_1, according to the new criterion, transition cost function can be defined as G_2

$$G_1(X,U,Y)=\begin{cases}-1 & if\ y_1-x_1=1\\ 0 & otherwise\end{cases}$$
$$G_2(X,U,Y)=ag(X,U,Y)+bg'(X,U,Y) \tag{1}$$

boundary condition is $\left|x_1+x_3-x_2-x_4\right|=N$, N is the parts number in the whole re-entrant line, where

$$g(X,U,Y)=\begin{cases}-1 & if\ y_1-x_1=1\\ 0 & otherwise\end{cases}\quad g'(X,U,Y)=\begin{cases}1 & if\left|y_1+y_3-y_2-y_4\right|=N\\ 0 & otherwise\end{cases} \tag{2}$$

$g(X,U,Y)$ means once the system outputs a part, the corresponding transition cost decrease 1. The more parts the system output, the fewer the corresponding cost are and the better the policy is. $g'(X,U,Y)$ means a punishment for the block in system. Once the system is in block, the transition cost increases. a,b are the weights of $g(X,U,Y)$ and $g'(X,U,Y)$ in the transition cost function. In a state track, block occurs only once, while system output many parts generally, so the value of b should be greater than that of a. After several trials, $a=1,b=50$ are the proper weights of $g(X,U,Y)$ and $g'(X,U,Y)$. In order to compare the learning effect of G_2 with that of G_1, the learning results are shown in Fig.2. The learning curve of G_2 converges more quickly, which is also more smooth and has less fluctuates. Temporal difference method in NDP are applied to schedule the CRL in [simulation 1], the results are shown in Table 2.

The policy obtained by NDP for $N=20$ is applied directly to the system $N=60$, the simulation results are shown in Table 3.

From Table1, 2 and 3, the scheduling results of NDP are better than the four static policies both in Mean Output Rate and Mean-Output-parts Number Before First-Block, which verify the effective of the new criterion. Further more, the policy for

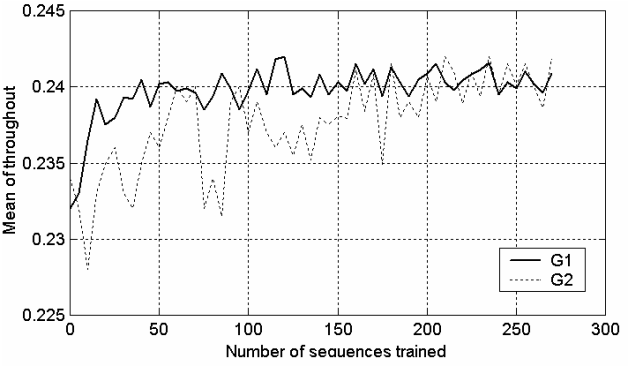


Fig. 2. Learning curves of two transition functions

Table 2. simulation results of NDP (N=20) (95% confidence interval)

Policy	Mean Output Rate	Mean-Output-parts Number Before First Block
NDP	0.2408±0.0007	50.3245±1.0524

Table 3 Simulation results of four policies and NDP (N=60) (95% confidence interval)

Policy	Mean Output Rate	Mean-Output-parts Number Before First Block
NDP	0.2456±0.0009	510.8647±13.3487
WB	0.2448±0.0012	490.0540±15.8608
FBFS	0.2397±0.0008	175.5690±10.1792
LBFS	0.2386±0.0007	52.0150±1.9812
UNWB	0.2289±0.0008	38.7140±1.5779

$N = 20$ still keep the superiority when applied directly to $N = 60$, which make it feasible to learning in the small scale system then extend to the large scale system.

5 Conclusion

Scheduling of the CRL is a strong coupling and complex problem with large state space. Most research is based on the evaluation criterion of Mean Output Rate. Mean-Output-parts Number Before First Block is presented to be a new evaluating criterion in this paper to evaluate scheduling policies for the CRL. This new criterion is more appropriate to model a system as a discrete-time Markov process with abort state. Simulations of four static scheduling policies verify the new criterion. Furthermore, cost-to-go value function and transition cost function of Neuro-Dynamic Programming are presented as new forms under the new criterion to schedule a CRL. The policy obtained in less-number parts system is applied in more-number parts system directly, whose results are satisfactory.

References

1. Kumar, P.R.: Re-entrant Lines. Queuing Systems: Theory and Applications. Special Issue on Queuing Networks 13 (1993) 87-110
2. Lu, S.C.H., Ramaswamy, D., and Kumar, P.R.: Efficient Scheduling Policies to Reduce Mean and Variance of Cycle-time in Semiconductor Manufacturing Plants. IEEE Trans. Semiconductor Manufacturing 7 (1994) 374-385
3. Lu, S.C.H., Kumar, P.R.: Fluctuation Smoothing Scheduling Policies for Queuing Systems. Workshop on Computing and Intelligent Systems, Bangalore, India 12 (1993)
4. Johnson, S.M.: Optimal Two and Three-stage Production Schedules with Set-up Times Included. Nav. Res. Logistics Quart 1 (1954) 61-68
5. Garey, M.R., Johnson, D.S.: Computers and Intertractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, San Francisco, California (1979)
6. Bertsekas, D.P., Tsitsiklis, J.N.: Neuro-Dynamic Programming. Athena Scientific (1996)
7. Lippman, S.: Applying a New Device in the Optimization of Exponential Queuing Systems. Operations Research 23 (1975) 687-710

A Constraint Satisfaction Adaptive Neural Network with Dynamic Model for Job-Shop Scheduling Problem

Li-Ning Xing, Ying-Wu Chen, and Xue-Shi Shen

Department of Management, School of Information System and Management, National University of Defense Technology, Changsha 410073, P.R. China
xln_2002@nudt.edu.cn, phd2008@126.com, fblunwen@126.com

Abstract. It is well known, the Job-Shop Scheduling Problem (JSSP) is the most complicated and typical problem of all kinds of production scheduling problems, the allocation of resources over time to perform a collection of tasks. The current method has several shortcomings in solving the JSSP. In this paper, we correct these deficiencies by introducing a dynamic model that is based on an analysis of the run-time behavior of CSANN algorithm. At the same time, this paper proposes several new heuristics in order to improve the performance of CSANN. The computational simulations have shown that the proposed hybrid approach has good performance with respect to the quality of solution and the speed of computation.

1 Introduction

It is well known, the Job-Shop Scheduling Problem (JSSP) is the most complicated and typical problem of all kinds of production scheduling problems, the allocation of resources over time to perform a collection of tasks [1]. It has been demonstrated that job-shop scheduling is usually an NP-complete (nondeterministic polynomial time complete) problem [2]. Because of the NP-complete characteristics of job-shop scheduling, it is usually very hard to find its optimal solution, and an optimal solution in the mathematical sense is not always necessary in practices [3]. Researchers turned to search its near-optimal solutions with all kind of heuristic algorithms [4]. Fortunately, the searched near-optimal solutions usually meet requirements of practical problems very well.

Reference [5] has proposed an efficient Constraint Satisfaction Adaptive Neural Network (CSANN) and heuristics combined approach for job-shop scheduling problems. In order to improve the performance of CSANN, several new heuristic based on the property of non-delay schedules. But CSANN is not tractable in solving the large size JSSP. In this paper, we introduce a dynamic model that is based on an analysis of the run-time behavior of CSANN algorithm. At the same time, this paper proposes several new heuristics in order to improve the performance of CSANN.

This paper is organized as follows. Section 2 describes the hybrid approach proposed by this paper. Section 3 presents the computer simulation results with 8 examples to show the performance of the proposed new hybrid approach for job-shop scheduling. Finally, the conclusions about the hybrid approach are presented in Section 4.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 927 – 932, 2006.

2 Description of Hybrid Approach

This section describes the hybrid approach proposed by this paper; it introduces the computational flow (Fig.1.) and material implement of this approach detailed.

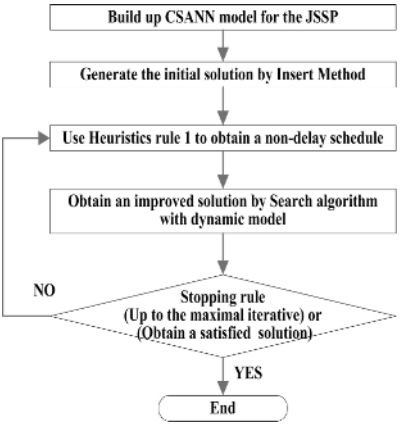


Fig. 1. The computational flow of this hybrid approach (left)

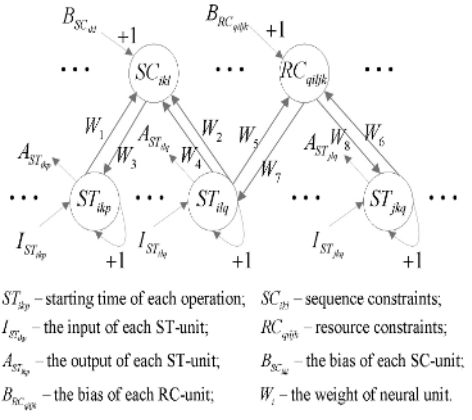


Fig. 2. The CSANN model for the JSSP (right)

2.1 Build Up CSANN Model for the JSSP

The architecture of CSANN model consists of two layers (Fig.2.). The bottom layer consists of only ST-units, corresponding to the starting times of all operations. The top layer contains SC-units and RC-units, which represent sequence and resource constraints respectively and provide feedback information to adjust ST-units in order to satisfy sequence and resource constraints through SC-block and RC-block respectively.

2.2 Generate the Initial Solution by Insert Method

Because CSANN technology is sensitive to the initial solution, so it is a good beginning that generates an excellent solution using other method. In this paper, we apply the Insert Method which belongs to a heuristic algorithm based on time priority to generate an excellent solution. The computational flow of Insert Method is in the following.

STEP 1: Sort the pending job by degressive priority according to the total processing time. Toward two jobs which have the same total processing time, prearrange the job which has more operations.
STEP 2: Arrange the job by the order of the list obtained from STEP1 to each machine.
STEP 3: Arrange the operation by the sequence order to each machine.
STEP 4: Insert the operation obtained from STEP3 forward in compliance with the corresponding sequence constraint or resource constraint.

2.3 Use Heuristics Rule to Obtain a Non-delay Schedule

A schedule is non-delay if no machine lies idle when there is at least one job waiting to be operated on that machine. A non-delay schedule is a local optimal schedule with orders of operations to be operated on each machine already determined.

Heuristics Rule1: Assuming a feasible solution $\{T_{ikp}, i \in N, k \in \{1,2,\cdots,n_i\}, p \in M\}$ has been obtained by previous part. Sort them in non-decreasing order. Then from the minimal to the maximal, each T_{ikp} is adjusted as follows:

$$T_{ikp}^{'} = \begin{cases} T_{ljp} + P_{ljp} & T_{ljp} + P_{ljp} \geq T_{i(k-1)q} + P_{i(k-1)q} \\ T_{i(k-1)q} + P_{i(k-1)q} & T_{ljp} + P_{ljp} < T_{i(k-1)q} + P_{i(k-1)q} \end{cases} \quad (1)$$

Where $T_{ikp}^{'}$ is the starting time of O_{ikp} in the obtained non-delay schedule after the heuristics is run. $O_{i(k-1)q}$ is the precedence operation of O_{ikp} from the same job i, and O_{ljp} is the precedence operation of O_{ikp} sharing the same machine p. Equation (1) means to shorten each starting time T_{ikp} to the completion time of $O_{i(k-1)q}$ or the completion time of O_{ljp}, depending on whichever is smaller.

2.4 Obtain an Improved Solution by Search Algorithm with Dynamic Model

2.4.1 The Dynamic Model of Search Algorithm

Given the objective of makespan minimization, we define a state $S_{OP,DI,EF}$ in our Markov model as the following representing (1) OP denotes a discretionary operation; (2) DI denotes the moving direction of OP, $DI \in \{forward(1), backward(-1)\}$; EF denotes the adjustive effect of the total makespan through moving OP, $EF \in \{bigger(-1), smaller(1)\}$. In our Markov model, the state $S_{OP,DI,EF}$ represents the fluctuant effect of the total search process.

Next, we specify the transition probabilities between pairs of states $S_{OP,DI,EF}$ and $S_{OP,DI',EF'}$ in our model. Let the conditional probability $P(S_{OP,DI,EF} \mid S_{OP,DI',EF'})$ denotes the probability of simultaneously altering the adjustive effect of the total makespan from EF' to EF and moving the operation OP from the direction DI' to DI. The set of transition probabilities is subject to the total-probability constraints:

$$P(S_{OP,1,1} \mid S_{OP,1,1}) + P(S_{OP,1,-1} \mid S_{OP,1,1}) + P(S_{OP,-1,1} \mid S_{OP,1,1}) + P(S_{OP,-1,-1} \mid S_{OP,1,1}) = 1.0;$$

$$P(S_{OP,1,1} \mid S_{OP,1,-1}) + P(S_{OP,1,-1} \mid S_{OP,1,-1}) + P(S_{OP,-1,1} \mid S_{OP,1,-1}) + P(S_{OP,-1,-1} \mid S_{OP,1,-1}) = 1.0;$$

$$P(S_{OP,1,1} \mid S_{OP,-1,1}) + P(S_{OP,1,-1} \mid S_{OP,-1,1}) + P(S_{OP,-1,1} \mid S_{OP,-1,1}) + P(S_{OP,-1,-1} \mid S_{OP,-1,1}) = 1.0;$$

$$P(S_{OP,1,1} \mid S_{OP,-1,-1}) + P(S_{OP,1,-1} \mid S_{OP,-1,-1}) + P(S_{OP,-1,1} \mid S_{OP,-1,-1}) + P(S_{OP,-1,-1} \mid S_{OP,-1,-1}) = 1.0.$$

2.4.2 The Search Algorithm with Dynamic Model

The computational flow of the search algorithm with dynamic model is in the following.

STEP 1: initialize the transition probabilities.

$$P\left(S_{OP,1,EF} \mid S_{OP,DI,EF}\right)=0, P\left(S_{OP,-1,EF} \mid S_{OP,DI,EF}\right)=0.5;\quad OP \in \{first\ oper.of\ each\ mach.\};$$

$$P\left(S_{OP,1,EF} \mid S_{OP,DI,EF}\right)=0.5, P\left(S_{OP,-1,EF} \mid S_{OP,DI,EF}\right)=0;\quad OP \in \{last\ oper.of\ each\ mach.\};$$

$$P\left(S_{OP,1,EF} \mid S_{OP,DI,EF}\right)=0.25, P\left(S_{OP,-1,EF} \mid S_{OP,DI,EF}\right)=0.25;\quad OP \in \{other\ oper.of\ each\ mach.\};$$

STEP 2: select an operation randomly, move it according to the transition probabilities. Moving the giving operation forward or backward is that exchange the orders of two adjacent operations. It will be described in the next part detailed.

STEP 3: update the transition probabilities according to the exchange result.

We denote the total number of occurrences of state $S_{op,DI,EF}$ by $\#\left(S_{op,DI,EF}\right)$ and the total number of occurrences of a successor state $S_{op,DI',EF'}$ given a current state $S_{op,DI,EF}$ by $\#\left(S_{op,DI',EF'} \mid S_{op,DI,EF}\right)$; both quantities are tracked over all iterations of all trials. We update the transition probabilities using the obvious formulas, e.g.,

$$P\left(S_{op,DI',EF'} \mid S_{op,DI,EF}\right)=\frac{\#\left(S_{op,DI',EF'} \mid S_{op,DI,EF}\right)}{\#\left(S_{op,DI,EF}\right)} \tag{2}$$

STEP4: repeat STEP2-STEP3 until the terminative condition is satisfied. The terminative condition can be (1) repeat this cycle for a fixed times and (2) the optimal solution cannot be improved for several cycles.

2.4.3 Exchange the Orders of Two Adjacent Operations

Heuristics 2: Exchange the orders of two adjacent operations. This heuristics has two aspects of function: to accelerate the solving process and to guarantee feasible solution. The former is for two adjacent operations coming from the same job, while the latter is for two adjacent operations sharing the same machine.

On the one hand, assume $\left[O_{ikp}, O_{ilq}\right] \in S_i$. In order to accelerate the solving speed of CSANN, at time t during its processing, if $A_{ST_{ikp}}(t) \geq A_{ST_{ilq}}(t)$, exchange the orders of O_{ikp} *and* O_{ilq} by exchanging their starting times as follows:

$$\begin{cases} A_{ST_{ikp}}(t+1)=T_{ikp}(t+1)=T_{ilq}(t) \\ A_{ST_{ilq}}(t+1)=T_{ilq}(t+1)=T_{ikp}(t) \end{cases} \tag{3}$$

In fact, Eqs. (3) is a more direct method of removing sequence violation than that of the feedback adjustment of CSANN. Thus the adjustment time from removing sequence violations may be shortened and the solving process of CSANN for feasible solution is accelerated.

On the other hand, during the processing of CSANN there may appear the phenomenon of "dead lock" which can result in no feasible solution. In order to remove "dead lock", we use the following heuristic: exchange the orders of two near operations sharing the same machine by exchanging their starting times. Assuming O_{ikq} *and* $O_{ljq} \in R_q$, during the processing of CSANN, if $T_{qiklj}(t) \geq T$, the following equations begin to work:

$$\begin{cases} A_{ST_{ikq}}(t+1) = T_{ikq}(t+1) = T_{jlq}(t) \\ A_{ST_{jlq}}(t+1) = T_{jlq}(t+1) = T_{ikq}(t) \end{cases} \tag{4}$$

Where the parameter T is a prescribed positive integer, variable $T_{qiklj}(t)$ is the summed continuous change times between the starting times of operation pairs O_{ikq} *and* O_{ljq} (sharing machine q) because of their resource violation. That is, at time t, the starting times of O_{ikq} *and* O_{ljq} have already continuously changed $T_{qiklj}(t)$ times because of their resource violation on machine q, and the changing effects are the same (e.g., always putting T_{ikq} forwards and T_{ljq} backwards). When $T_{qiklj}(t)$ reaches T, Eqs. (4) begins to work.

The above heuristic can be used together with CSANN to guarantee the feasible solution. The phenomenon of "dead lock" results from the conflicts of feedback adjustments while removing sequence and resource constraint violations.

3 Simulation Study

In this section, the proposed method was tested using a set of the benchmark JSSP. To avoid any misinterpretation of the optimization results, relating to the choice of a particular initial solution, we performed each test 100 times. The performance of the proposed technology is compared to two other published versions of Tabu Search and CSANN. The experimental results obtained for 8 test problems, using the 3 different methods, are given in table 1. Both the average optimization time and ave-rageoptimization result, the proposed method of this paper is better than the other two

Table 1. The experimental results obtained for 8 test problems using the 3 methods

Test Problem	Average Optimization Time (t)			Average Optimization Result		
	TS	CSANN	This method	TS	CSANN	This method
Abz7	222.41	231.94	161.82	678	684	674
Abz8	248.57	259.22	186.73	691	697	688
La38	91.27	95.177	86.92	1235	1247	1229
La39	100.47	104.78	75.69	1273	1285	1267
La40	108.42	113.07	93.26	1262	1274	1256
Swv12	553.76	577.49	427.39	3069	3098	3055
Swv13	580.92	605.82	453.26	3206	3235	3190
Swv14	566.13	590.39	439.17	3065	3093	3051

methods. Simulations have shown that the proposed hybrid approach for job-shop scheduling has excellent performance with respect to the quality of solutions and the speed of calculation.

4 Conclusions

In view of several shortcomings of current method in solving the JSSP, this paper presents a constraint satisfaction adaptive neural network with a dynamic model. The contribution of this paper is summarized in the following. (1) Introduce the Insert Method (generate the initial solution by Insert Method) to the current CSANN model; it avoids the sensitivity to the initial solution. (2) join a dynamic model that is based on an analysis of the run-time behavior of the total optimization process into the CSANN model; it mines and applies some potential global knowledge to the optimization process. The computational simulations have shown that the proposed hybrid approach has good performance with respect to the quality of solution and the speed of computation.

Acknowledgement

The authors would like to thank our tutor Mr. CHEN. He encourages us to do some innovative working; he presents many new ideas to us in the spare time; he gives many constructive advices in the process of writing this paper. Thanks to him again.

References

1. Baker, K. R.: Introduction to Sequence and Scheduling. Wiley, New York (1974)
2. Garey, M.R., Johnson, D.S., Sethi, R.: The Complexity of Flow-Shop and Job-Shop Scheduling. Mathematics of Operations Research 1(2) (1976) 117-129
3. Erschler, J.F., Roubellat, J.P., Vernhes: Finding Some Essential Characteristics of the Feasible Solutions for a Scheduling Problem. Operations Research 24(4) (1976) 774-783
4. French, S.: Sequencing and Scheduling: an Introduction to the Mathematics of the Job-Shop. Wiley, New York (1982)
5. Yang, S., Wang, D.: Constraint Satisfaction Adaptive Neural Network and Heuristics Combined Approaches for Generalized Job-Shop Scheduling. IEEE Transactions on Neural Networks 11(2) (2000) 474-486

Neural Network Based Industrial Processes Monitoring

Luis P. Sánchez-Fernández, Cornelio Yáñez-Márquez, and Oleksiy Pogrebnyak

Centro de Investigación en Computación, Instituto Politécnico Nacional,
Av. Juan de Dios Bátiz s/n casi esq. Miguel Othón de Mendizábal,
Col. Nueva Industrial Vallejo. CP 07738, Mexico City, Mexico
{lsanchez, cyanez, olek}@cic.ipn.mx

Abstract. This industrial processes monitoring based on a neural network presents low run-time, and it useful for critical time tasks with periodic processing. This method allows the time prediction in which a variable will arrive to abnormal or important values. The data of each variable are used to estimate the parameters of a continuous mathematical model. At this moment, four models are used: first-order or second-order in three types (critically damped, overdamped or underdamped). An optimization algorithm is used for estimating the model parameters for a dynamic response to step input function, because this is the most frequent disturbance. Before performing the estimation, the most appropriate model is determined by means of a feed-forward neural network.

1 Introduction

In many studies it has been estimated that the petrochemical industry of United States loses $10 billion annually due to abnormal situations [1], a generic term describing departures from acceptable operating conditions. Consequently, there is considerable incentive for the development of process monitoring and fault detection techniques that can detect an abnormal situation quickly and accurately [2]. If an abnormal operation can be predicted with a sufficient warning period, then an effective corrective action may be able to prevent the predicted operation from actually occurring [3]. In several papers, some methods are proposed that signals impending emergency limit violations by using predictions from a process model [3].

This paper proposes an efficient predictive supervisory algorithm for the trending of variables of industrial processes, with low run-time, for periodic analysis of periods smaller than a second. The method allows to predict the time in which the variables of a process will arrive to a critical or abnormal value. Neural networks (NN) are very powerful mathematical tools for systems identification. In our case, a NN is used to recognize signal patterns of first and second order dynamic systems [4] in which the dynamics of a considerable amount of industrial processes can be represented [5], [6]. The methodology consists of to recognize the most appropriate model by means of a NN and later, the parameters are estimated through an optimization algorithm [7] thus reducing the total processing time.

This work bears relationship to a broad range of mathematical techniques, ranging from statistics to fuzzy logic, which have been used successfully in intelligent data analysis [8]. The transfer functions of the used models are the following:

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 933 – 938, 2006.

First order model:

$$\frac{Y(S)}{X(S)} = \frac{Ke^{-\theta S}}{T_1 s+1} \quad T_1\text{: time constant; K: gain;} \tag{1}$$

Second order model:

$$\frac{Y(S)}{X(S)} = \frac{Ke^{-\theta S}}{s^2/\omega n^2 + 2\xi s/\omega n + 1} \quad \omega_n\text{:natural oscillation freq.; } \theta\text{ :time delay; } \zeta\text{:coeff. damping.} \tag{2}$$

2 Block Chart of the Method

A circular buffer of changeable dimensions is used. This cycle begins with the permanent storing of N last data of the variable of the technological process or device under supervision.

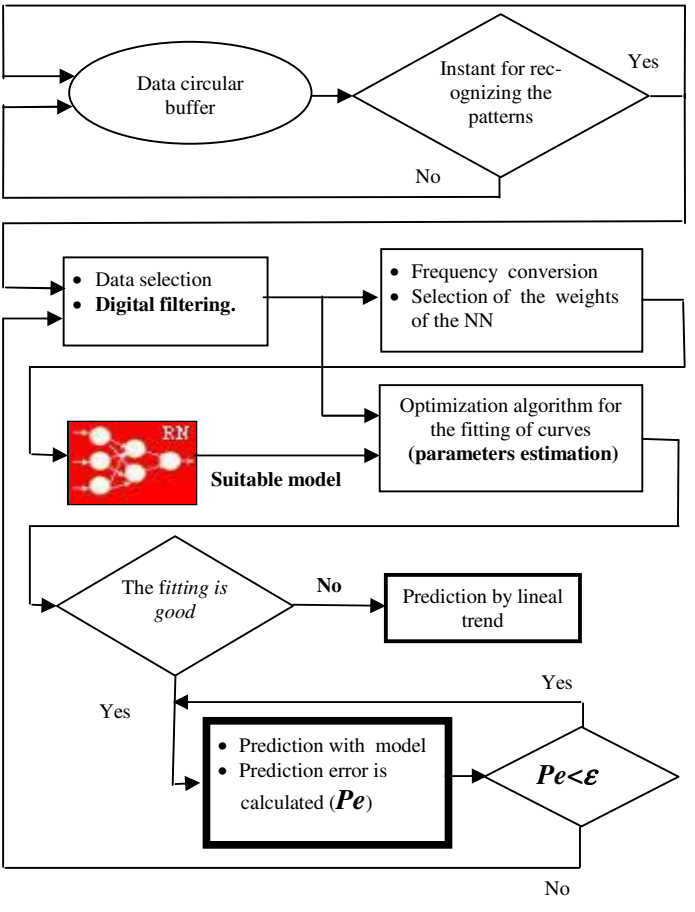


Fig. 1. Flow diagram of the predictive alarm algorithm cycle

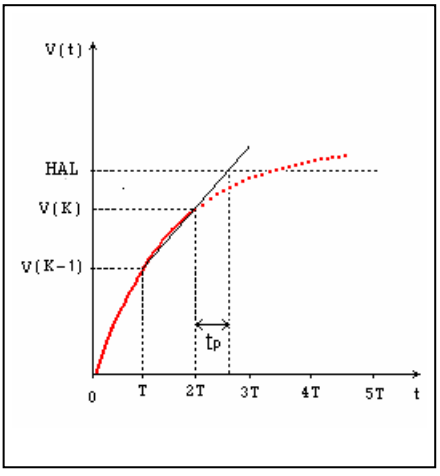


Fig. 2. Prediction based on the linear-trend of the variable

Fig. 2 shows an example of a variable versus time plot V(t) with the following parameters. HAL: high alarm limit. v(k): current value. v(k-1): previous value. T: sampling period. t_p: prediction time.

The current sampling instant in this example is 2T. As shown in Fig. 2, the ***instant for recognizing the pattern of transitory response***, of the model representing by the points stored in the circular buffer is determined by an algorithm of lineal trend prediction. On predicting by lineal trend, the behavior of the variable is considered to be that of a straight line from the current sampling instant.

Regarding Fig. 2, it can be stated that:

$$\frac{V(k) - V(k-1)}{2T - T} = \frac{HAL - V(k)}{tp} \tag{3}$$

obtaining t_p as:

$$tp = \left(\frac{T}{V(k) - V(k-1)} \right) [HAL - V(k)] \tag{4}$$

A minimum prediction time T_{mp} must be set, such that if $t_p < T_{mp}$, then the recognition process of the signal pattern represented by the samples stored in the circular buffer begins. ***The latest data are selected*** if it is the ***instant for recognizing the pattern of transitory response***. The oldest data are discarded until significant points indicating the beginning of a new temporary response (thick line) are found. Then, a ***sampling frequency conversion*** with a non-integer factor combining interpolation and decimation is performed [9], which makes it possible to obtain 30 points. This is the number of input neurons of the NN. Later, the ***selection of the weights of the NN*** is accomplished in accordance with the sign of the 30-point-curve slope, since it was trained for the patterns with a positive and negative slope. As an output, the NN will

produce the most *suitable model* with its parameters estimated through an ***optimization*** algorithm ***for the fitting of curves*** using all the selected points**.** Hooke and Jeeves´s [10] optimization method of direct search is used.

3 First and Second Order Systems Patterns for Training the NN

The selection of the NN training patterns was based on the behavior of the dynamic responses of first- and second-order systems to step input function, because this is the most frequent type of disturbance. In other cases, even though it might not strictly be an ideal step, it can be considered as such, provided that for instance, the time constants of the process are relatively larger than the time constants of an exponential signal. For a ramp input function, the lineal trend algorithm is sufficient for a prediction with high accuracy.

If we have the responses of critically damped second order systems, with natural oscillation frequencies w_n equal to 1, 0.5 and 0.25, respectively, and the same sampling frequency, we can to obtain 30 points for every curve. If these points were graphically represented using the same time interval for axis X, they would be superimposed, as shown in Fig. 3.

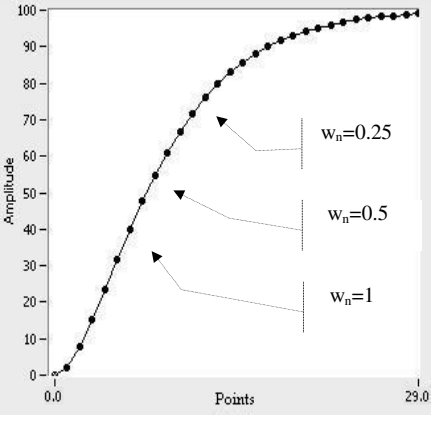


Fig. 3. The three curves superimposed

A similar behavior will occur in first order systems with respect to the time constant, as well as in overdamped and underdamped second order systems, in which only its coefficient of damping will show any difference. It can be seen that w_n does not exert any influence on the signal pattern when each mark on axis X indicates the order in which the points were taken up rather than the time. This is the procedure to follow, since it is not of any interest to estimate the parameters of the model (time constant and gain for first order systems; gain, ζ and w_n for second order systems). It only matters to recognize, through the neural network, the type of model that best represents the register of stored data. Therefore, for input patterns of critically

damped first and second order systems, only the amplitude that varies; for over-damped and underdamped second order systems only the amplitude and the ζ vary.

For NN training patterns, the maximum value of signal amplitude are taken up in percents (%), standardized, from 40% to 90%.

After numerous tests, training was carried out with 858 input patterns, distributed in the following way:

- For overdamped second order systems (OSOS):
  For every ζ value, 11 patterns are obtained corresponding to the variations of the amplitude from 40 to 90, with an increase of 5 (40, 45, 50, …, 90). The ζ varies from 1.2 to 3, with an increase of 0.09 (**220 patterns).** For ζ greater than 3, the behavior of the system is similar to a first order system.
- For underdamped second-order systems (USOS):
  As for every ζ value, 11 patterns for amplitude variations are obtained. The ζ varies from 0.1 to 0.7, with an increase of 0.0667 (**99 patterns**)
- For first- (FOS) and critically damped second-order (CSOS) systems, **11 patterns** are created, respectively, corresponding to amplitude variations from 40 to 90.

In order to have a similar number of patterns for each model and achieve a better training of the NN, the F and CSOS patterns are repeated 20 times, respectively, for a total of 440 patterns. For the USOS pattern they are repeated twice for 198 patterns. 858 PATTERNS IN ALL. Once the patterns were chosen, varied topologies were used until the simplest with the most suitable response was obtained. Eventually, a 30-input neural network was used, 11 neurons in the hidden layer and four-output neurons. Very good results were obtained in the training and generalization of the NN. The training error was 0.15%. Over 1000 test patterns were used, obtaining a correct response, with an error of 0.9% of failures.

4 Results Assessment of the Patterns Recognition

The following Table presents the results obtained from five test patterns which were well recognized by neural network. The used legends are the following: **osos**: Over-damped second-order systems; **usos**: Underdamped second-order systems; **csos**: Critically damped second-order systems, and **fos**: First-order systems.

Test Pattern	**N1 (fos)**	**N2 (csos)**	**N3 (osos)**	**N4 (usos)**
fos	**0.998687**	0	0.002310	0
csos 1	0.000027	**0.996556**	0.003431	0.000000
csos 2	0	**0.998668**	0.002008	0.000015
usos	0.000008	0	0	**0.999989**
osos	0	0	**1**	0

The following Table presents the results obtained from three noisy test patterns for first-order systems which were well recognized by neural network.

Test Pattern	**N1 (fos)**	**N2 (csos)**	**N3 (osos)**	**N4 (usos)**
fos 1	**0.999999**	0	0	0
fos 2	**0.999979**	0	0	0
fos 3	**0.962691**	0	0	0

5 Conclusions and Future Work

Satisfactory results were obtained on training the neural network, having a high level of generalization. During the operation, the neural network recognized the signals used, even those affected by noise. The research and the technological advances presented are a satisfactory step forward in facilitating the use of advanced and efficient algorithms of predictive alarm by trend, with minimum processing time. The method has been applied by using LabVIEW and DLL's written in other programming languages such as C and Delphi. Signals affected by higher levels of noise will be used for future studies. This optimization algorithm would be extremely efficient as its initial operation conditions would be the values pre-estimated by the neural network. This algorithm has been successfully applied in forecasts of temperature, level, hydraulic channels and simulation.

Acknowledgements

The authors would like to thank the Instituto Politécnico Nacional (Secretaría Académica, COFAA, CGPI, and CIC), the CONACyT, and SNI for their economical support to develop this work.

References

1. Nimmo, I.: Adequately Address Abnormal Operations. Chem. Eng. Progr. 91(1995) 36-45
2. Juricek, B., Larimore, W., Seborg, D.: Early Detection of Alarm Situations Using Model Predictions. In: Proc. IFAC Workshop on On-Line Fault Detection, Lyon, France (1998)
3. Juricek, B., Dale E., Seborg, D., Larimore, W.: Predictive Monitoring for Abnormal Situation Management. Journal of Process Control 11 (2001) 111-128
4. Ogata, K.: Modern Control Engineering. 4th. Edition. Prentice Hall, NY (2001).
5. Boyer, S.A.: SCADA: Supervisory Control and Data Acquisition. 3rd edn. Book News, Inc., Portland, OR (2004)
6. Södertröm, T., Stoica, P.: System Identification. Englewood Cliffs, NJ, Prentice-Hall (1989)
7. Edgar, T., Himmelblau, D.: Optimization of Chemical Processes. NY, MacGraw-Hill (1988)
8. Robins, V. et al.: Topology and Intelligent Data. Advances in Intelligent Data Analysis V, Lecture Notes in Computer Science, Vol. 2810. Springer-Verlag GmbH (2003) 275–285
9. Oppenheim, A., Schafer, R., Buck, J.: Discrete-Time Signal Processing. 2nd edn. Prentice-Hall Int. Editions (1999)
10. Hooke, R., Jeeves, T.: Direct Search Solution for Numerical and Statistical Problems. Journal ACM 8 (1961) 212-221

A New Method for Process Monitoring Based on Mixture Probabilistic Principal Component Analysis Models

Zhong-Gai Zhao and Fei Liu

Institute of Automation, Southern Yangtze University,
Wuxi, 214122, P.R. China
gaizihao@yahoo.com.cn, fliu@thmz.com

Abstract. Conventional PCA-based monitoring method relies on the assumption that process data is normally distributed, which the actual industrial processes often don't satisfy. Instead, mixture probabilistic principal component analysis (MPPCA) models are suitable to process with any probability density function. But, it suffers a drawback that the needed charts are too many to be watched in practice while the number of sub-models in MPPCA is large. Different from existing MPPCA, this paper proposes a novel method, which integrates every monitoring chart of MPPCA models into only one chart via probability and field process monitoring can rely on just one chart. The application in real chemical separation process shows validity of the proposed method.

1 Introduction

Conventional PCA-based monitoring method relies on the assumption that process data is normally distributed. However in the real industrial process, because there exist nonlinearities, dynamics, and multiple steady or multiple operating states, the data obtained from steady state process seldom satisfies the assumption [1]. In order to break the limitation, several extended approaches for PCA have been developed. Chen and Liu propose a mixture principal component analysis (MixPCA) models detector [2]. Zhang [3] applies mixture probabilistic PCA (MPPCA) models developed by Tipping and Bishop [4] to process monitoring. Unfortunately, while process data is clustered into too many MPPCA sub-models, the monitoring charts relating to all sub-models are too many to be kept watching. To overcome this disadvantage, Choi, *et al.* [5] integrate all SPE and T^2 charts of sub-models into two overall SPE and T^2 charts. But if the numbers of principal components (PCs) in each sub-model are different, the control limit of overall chart cannot be calculated and so overall chart cannot be obtained.

Aiming to improve MPPCA-based monitoring, this paper proposes a new method, which monitors sub-models with probabilistic principal component analysis (PPCA) rather than PCA used in [5]. Because the confidence limits of monitoring indices in PPCA among all sub-models (i.e. components) are identical even if the numbers of selected PCs are different, an overall chart is constructed

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 939–944, 2006.

by probability instead of all the charts in components. During monitoring, field operators can detect fault just only by the overall chart: if the index value of current sample fall into the control limit, the process operates normally; or else the process may have something wrong. In the end, the proposed methods is applied into a real chemical separation process, and the result shows its validity.

2 MPPCA Models

Assume that the independent sample data $X = [x_1, x_2, \cdots, x_N] \in R^{q \times N}$, meets an arbitrary probability density, $p(X \mid \theta)$, where q is the number of variables and N is the number of samples. $p(X \mid \theta)$ can be approximated by a mixture of K Gaussian density functions, while Gaussian density function can be replaced with PPCA model according to generative model $X = W \cdot T + \mu + E$, where W is loading matrix, T is PCs matrix, μ is the mean vector and E is noise matrix. So $p(X \mid \theta)$ can be denoted by local PPCA models as following:

$$p(X \mid \theta) = \sum_{i=1}^{K} \beta_i p(X \mid i) \tag{1}$$

where β_i is the prior probability of the data point having been generated from component i of MPPCA models, satisfying $\sum_{i=1}^{K} \beta_i = 1$ and $\beta_i \geq 0$, $p(X \mid i)$ represents the ith PPCA model, $\theta = \{\beta_1, \beta_2, \cdots, \beta_K, \theta_1, \theta_2, \cdots, \theta_K\}$ and θ_i is a parameter vector of the ith PPCA model. According to Bayes formula, the posterior probability $p(i \mid X) = p(X \mid i)\beta_i / \sum_{j=1}^{K} [p(X \mid j)\beta_j]$, then the posterior probability of the nth sample $R_{ni} = p(i \mid x_n) = p(x_n \mid i)\beta_i / \sum_{j=1}^{K} [p(x_n \mid j)\beta_j]$. Using the EM algorithm, the estimated parameters can be determined by:

$$\widetilde{\beta}_i = N^{-1} \sum_{n=1}^{N} R_{ni} \tag{2}$$

$$\widetilde{\mu}_i = \sum_{n=1}^{N} R_{ni}(x_{ni} - \widetilde{W}_i \langle T_{ni} \rangle) / \sum_{n=1}^{N} R_{ni} \tag{3}$$

$$\widetilde{S}_i = 1/(\widetilde{\beta}_i N) \sum_{n=1}^{N} R_{ni}(x_n - \widetilde{\mu}_i)(x_n - \widetilde{\mu}_i)^T \tag{4}$$

$$\widetilde{W}_i = \widetilde{S}_i W_i (\sigma_i^2 I + M_i^{-1} W_i^T \widetilde{S}_i W_i)^{-1} \tag{5}$$

$$\widetilde{\sigma}_i^2 = (1/q) tr(\widetilde{S}_i - \widetilde{S}_i W_i M_i^{-1} W_i^T) \tag{6}$$

where μ_i, S_i, W_i, σ_i respectively represent the mean vector, covariance matrix, loading matrix, variance parameter of the ith component, while $M_i^{-1} = (\sigma_i^2 I + W_i^T W_i)^{-1}$, $\langle T_{ni} \rangle = M_i^{-1} W_i^T (x_n - \mu_i)$, $\widetilde{\cdot}$ represents the updated value of $\cdot$ by iteration of the above equations. Repeatedly iterate and the parameters are guaranteed to converge to a local optimal point by EM algorithm.

3 New Process Monitoring Method Based on MPPCA Models

3.1 Analysis on Traditional Monitoring Based on MPPCA Models

In traditional MPPCA-based monitoring method, SPE and T^2 statistics are used as monitoring indices in every sub-model. For current sample, field operators must detect the monitoring charts of every sub-models, without loss of generality, assume the number of sub-models is l, the total number of charts wanted to monitor is $2l$, which is tedious for the field operators and makes it easy to give false alarms or missing alarms.

The literature [5] integrates all SPE charts and T^2 charts of every sub-model into an overall SPE chart and an overall T^2 chart. In MPPCA models, R_{ni} represents the relative probability of the nth sample belonging to the ith component and satisfies $\sum_{i=1}^{K} R_{ni} = 1$, if $R_{ni} > R_{nj}, j = 1, 2, \cdots i-1, i+1, \cdots K$, field monitoring can rely on just the charts in the ith component. Therefore, all of the local charts may be unified into a single chart according to R_{ni}. But in order to achieve this, the important problem is that the control limits of all component charts must be identical. While the control limits of SPE chart and T^2 chart, SPE_{lim} and T^2_{lim}, are given from the following equations:

$$SPE_{lim} = \theta_1 [\frac{C_\alpha \sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1}]^{1/h_0} \tag{7}$$

$$T^2_{lim} = \frac{d_i (N-1)}{N - d_i} F_{d_i; N-1; \alpha} \tag{8}$$

where $\theta_i = \sum_{d_i+1}^{q} (S_i)^j$, $j = 1, 2, 3$, $h_0 = 1 - 2\theta_1\theta_3/(3\theta_2^2)$, α is level of significance (Los), C_α is the critical value of normal distribution with α Los, d_i is the number of PCs in the ith component, $F_{d_i; N-1; \alpha}$ is an F-distribution with degrees of freedom d_i and $N-1$, and with α Los. According to formula (7) and (8), the control limits of SPE and T^2 chart in all components are not the same as each other if the covariance matrix and the number of PCs are different. So it is difficult to convert the charts based on SPE and T^2 statistics into two overall charts.

3.2 New Process Monitoring with Overall Chart

The previous work mostly focuses on the PCA-based method in sub-models and uses the SPE and the T^2 charts respectively monitor the residual value and the score value. As the mentioned above, it is difficult to synthesize the charts in all sub-models into two overall charts. The paper introduces PPCA-based method [6] into the monitoring in sub-model and overcomes the drawback.

Given the nth sample x_n, the corresponding probability density in the ith component is as following:

$$p(x) = (2\pi)^{-0.5q} det[(W_i W_i^T + \sigma_i^2 I)^{-1}]^{-0.5} \cdot exp(-0.5 x^T (W_i W_i^T + \sigma_i^2 I)^{-1} x) \tag{9}$$

As PCA-based method, process monitoring with PPCA also monitors the scores value and the residual value. The monitoring charts in the ith component are given by the following inequations.

$$\| M_i W_i^T x_n \|^2 \leq \chi^2_{1-\alpha;d_i} \tag{10}$$

$$\| \sigma_i^{-1}(I - W_i M_i W_i^T) x_n \|^2 \leq \chi^2_{1-\alpha;q} \tag{11}$$

The prior of the above two inequations are the monitoring index like T^2 statistic and SPE statistic in PCA-based monitoring, respectively; the back of sign of inequality respectively represents the control limit of the two statistics. If the ith component in which the sample satisfies the formula (10) and (11) exits, it can be concluded that the process operates well. The above two charts use the identical measurement unit, i.e. Mahalanobis norm, they can be combined into one chart as following:

$$\| (W_i W_i^T + \sigma_i^2 I)^{-0.5} x_n \|^2 \leq \chi^2_{1-\alpha;q} \tag{12}$$

The formula (12) is the so-called measurement variable monitoring index. Because the monitoring chart of measurement variable can take place of the monitoring charts of PCs and noises, so it is enough to monitor it to estimate the state of process. From formula (12), in the monitoring charts with PPCA, the control limits in all sub-models are identical and equal to $\chi^2_{1-\alpha;q}$ whatever the covariance matrix and the number of PCs are. Using the R_{ni}, it is easy to implement overall chart monitoring. For the nth sample, the detailed monitoring steps with overall chart after modeling are as follows: (i) Calculate the R_{ni}, $i = 1, 2, \cdots, K$; (ii) Get the component number the sample belongs to, assume i is the number; (iii) Calculate the monitoring index using formula (12); (iv) Determine whether the monitoring index violates the confidence limit $\chi^2_{1-\alpha;q}$ or not, if it does, the process may have something wrong.

One important issue in MPPCA models is to determine the number of components the MPPCA models include. Deterministic method is applied in the paper to determine the component size[7].

4 Application Study

Apply the novelty process monitoring method to an actual chemical separation process, the detailed description consults literature [8]. In industrial practice, a series of real data under normal operating condition is collected by on-line industrial computers, in which there are fifteen process variables including temperature, pressure, liquid level and flow rate.

Due to the complexity of real industrial process, it is possible that the process variables are not normally distributed. So the Q-Q plot is used to test the normality of variables and we find that almost all variables don't satisfy the normal distribution. Under this case, the MPPCA models are applied to model process data and detect fault. 400 samples out of real raw data under normal operating condition are selected to build MPPCA models.

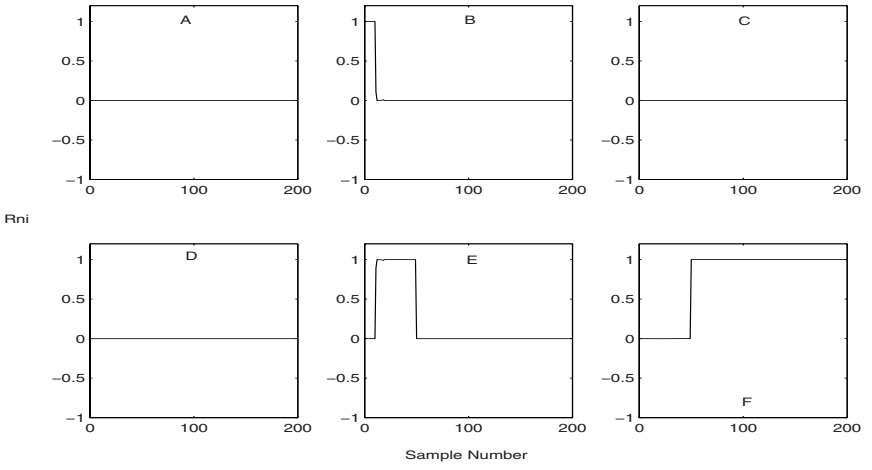


Fig. 1. Posterior probability

By deterministic method, the number of components is determined to be 6. Using EM algorithm, the parameters are estimated by means of Eqs. (2), (3), (4), (5), (6). After this, process monitoring is implemented with the MPPCA models. In conventional monitoring method based on MPPCA models, 12 charts including 6 SPE charts and 6 T^2 charts are needed to monitor at a sample, which is tedious for field operators. To facilitate monitoring, further research is to integrate all charts into one overall chart based on R_{ni}, the R_{ni} is shown in Fig 1.

Fig 1 tells the responsibility of which component produces the sample. From the Fig 1, sample 1 to sample 10 almost all belong to the 2th sub-model and sample 11 to about 50 almost all belong to the 5th component, and the rest samples all fall into the 6th component. Therefore, we can determine the process operation status during sample 1 to 10 by testing the 2th component, sample 11 to 50 by testing the 2th component and sample 51 to 200 by the 6th component. Based on this, PPCA-based monitoring method is introduced in every sub-model and the overall chart is given as Fig 2.

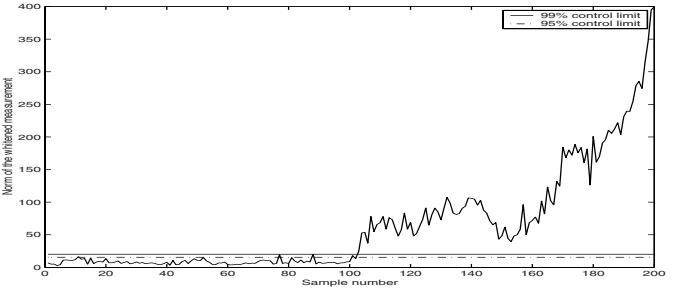


Fig. 2. Overall chart

From Fig 2, operators can easily estimate the state of process running by one chart. The monitoring result based on the novelty method is that the process is normal from sample 1 to sample 100 and faulty from sample 101 to sample 200. Compared with the traditional monitoring method based on MPPCA models, the workload of workers using the novelty method is very small and just the twelfth of that of traditional monitoring method.

5 Conclusion

In traditional monitoring method based on MPPCA models, field operators have large monitoring load. The literature introduces a new process monitoring method based on MPPCA models, in which operators just needs one monitoring chart and the workload of monitoring is largely lightened. In addition, the paper applies the method to an actual chemical separating process whose process variables are not normally distributed and compares the new method with the conventional MPPCA-based method in the application study, the result shows that the proposed method is better.

Acknowledgement

This work is supported by Key Project of Chinese Ministry of Education (NO 105088) and by Program for New Century Excellent Talents in University (NCET).

References

1. Martin, E.B., Morris, A.J.: Non-parametric Confidence Bounds for Process Performance Monitoring Charts. Journal of Process Control 6(6) (1996) 349-358
2. Chen, J.H., Liu, J.L.: Mixture Principal Component Analysis Models for Process Monitoring. Ind. Eng. Chem. Res. 38 (1999) 1478-1488
3. Zhang, F.: A Mixture Probabilistic PCA Model for Multivariate Processes Monitoring. Proceeding of the 2004 American Control Conference, Boston Massachusetts (2004) 3111-3115
4. Michael, E.T., Christopher, M.B.: Mixtures of Principal Component Analyzers. Neural Computation 11(2) (1999) 443-482
5. Choi, S.W., Park, J.H., Lee, I.B.: Process Monitoring Using a Gaussian Mixture Model via Principal Component Analysis and Discriminant Analysis. Computers and Chemical Engineering 28(8) (2004) 1377-1387
6. Kim, D.S., Lee, I.B.: Process Monitoring Based on Probabilistic PCA. Chemometrics and Intelligent Laboratory Systems 67(2) (2003) 109-123
7. Mario, A.T.F., Anil, K.J.: Unsupervised Learning of Finite Mixture Models. IEEE Trans. on P.A.M.I. 24(3) (2002) 381-396
8. Liu, F., Zhao, Z.G.: Chemical Separation Process Monitoring Based on Nonlinear Principal Component Analysis. Lecture Notes in Computer Science, Vol. 3173. Springer-Verlag, Berlin Heidelberg New York (2004) 798-803

On-Line Nonlinear Process Monitoring Using Kernel Principal Component Analysis and Neural Network

Zhong-Gai Zhao[1] and Fei Liu[2]

Institute of Automation, Southern Yangtze University,
Wuxi, 214122, P.R. China
gaizihao@yahoo.com.cn, fliu@thmz.com

Abstract. As a valid statistical tool, principal component analysis (PCA) has been widely used in industrial process monitoring. But due to its intrinsic linear character, it performs badly in nonlinear process monitoring. Kernel PCA (KPCA) can extract useful information in nonlinear data. However KPCA-based monitoring is not suitable for on-line monitoring because of large calculation and much memory occupation. The paper introduces an on-line monitoring method based on KPCA and neural network (NN), where KPCA is used to extract nonlinear principal components (PCs) and then NN approximates the relationship between process data and nonlinear PCs. We can obtain nonlinear PCs by NN to compute the monitoring indices and then achieve the on-line monitoring. The case study shows the validity of the method.

1 Introduction

Principal component analysis (PCA) is an optimal dimensionality reduction technique in terms of capturing the information of the data, it can improve the proficiency of process monitoring and so has been widely used in industrial process monitoring [1]. But due to intrinsic linear character, it shows a poor monitoring performance in nonlinear process. Some extensions of PCA have been developed to overcome such a drawback. Kernel PCA is a rising technique to tackle nonlinear problem [2], it firstly maps raw process data into high-dimensional feature space to get linear mapped data, then PCA is carried out on the mapped data using kernel trick. Because of its powerful ability of handling nonlinear character, KPCA has been used into process monitoring [3-5]. But when process is complex and process variables are too many, KPCA-based monitoring method bears large calculation, which results in that KPCA performs badly in on-line monitoring and is limited in off-line evaluation of process operation.

The paper extends KPCA to on-line process monitoring based on NN. In the method, KPCA is only used to extract PCs off-line in high-dimensional feature space, while NN is used to approximate the relationship between raw process data and PCs. During monitoring, PCs of the current sample in feature space are firstly obtained using NN, then the monitoring indices are calculated to achieve the on-line monitoring. By means of NN, the method avoids large calculation and

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 945–950, 2006.

memory occupation. In the end, a benchmark simulation of Tennessee-Eastman process (TEP) is researched, the result shows the validity of the method.

2 KPCA

Given the process sample data $X = [x_1^T, x_2^T, \cdots, x_N^T]^T$, where $x_i \in R^{1\times M}$ is the ith sample, N is the number of samples and M is the number of process variables. The high-dimensional nonlinear mapping function is $\{\phi\} : x_i \longrightarrow \phi(x_i)$, then the mapped data in high-dimensional space is $\phi(x_1), \phi(x_2), \cdots, \phi(x_N)$. This higher-dimensional linear space is referred to as the feature space. Assume $C^F = N^{-1}\sum_{j=1}^{N}\phi(x_j)^T\phi(x_j)$ is the covariance matrix of the mapped data, λ and v are respectively the eigenvalue and the associated eigenvector of C^F, so

$$C^F v = [N^{-1}\sum_{j=1}^{N}\phi(x_j)^T\phi(x_j)]v = N^{-1}\sum_{j=1}^{N}\langle\phi(x_j), v\rangle\phi(x_j) \tag{1}$$

Since formula (1) implies $v = \sum_{j=1}^{N}\alpha_j\phi(x_j)$, where $\alpha_j (j = 1, 2, \cdots, N)$ is the coefficients, formula (1) can be expressed as

$$\lambda\langle\phi(x_k), v\rangle = \langle\phi(x_k), C^F v\rangle, k = 1, 2, \cdots, N \tag{2}$$

Denote a kernel matrix $K \in R^{N\times N}$, where $[K]_{ij} = K_{ij} = \langle\phi(x_i), \phi(x_j)\rangle$, and combine Eqs. (1) and (2), we can obtain

$$N\lambda\alpha = K\alpha, \alpha = [\alpha_1, \alpha_2, \cdots, \alpha_N] \tag{3}$$

Eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$ and the corresponding eigenvectors α_1, α_2, $\cdots, \alpha_N$ are obtained by resolving the eigen-problem of Eqs. (3). and the dimensionality reduction is achieved by retaining only the first p eigenvectors. In order to guarantee $\langle v_k, v_k\rangle = 1$, $k = 1, 2, \cdots, q$, $\alpha_1, \alpha_2, \cdots, \alpha_q$ is normalize by $N\lambda_k\langle v_k, v_k\rangle = 1$. In the feature space, for a sample vector x, the value of principal component, t_k, by projecting the mapped data $\phi(x)$ onto loading vector v_k like

$$t_k = \langle v_k, \phi(x)\rangle = \sum_{i=1}^{N}\alpha_{ik}\langle\phi(x_i), \phi(x)\rangle \tag{4}$$

where α_{ik} is the ith element of vector α_k. Let $k = 1, 2, \cdots, q$, PCs $(t_1, t_2, \cdots, t_q)$ are computed one by one, denote PCs matrix $T = [t_1, t_2, \cdots, t_q]$, where q is the number of PCs. The number of PCs can be determined according to [3].

It must be clarified that mean centering in feature space should be performed before carrying out KPCA [6] and the paper does not include it for simplicity.

3 On-Line Process Monitoring Using KPCA and NN

3.1 Process Monitoring Based on KPCA

The key idea of the method is that the linear mapped data is monitored in feature space using PCA-based method, it can be concluded that process is under

wrong operation if the monitoring indices of mapped data violate the confidence limits. The monitoring indices in feature space are SPE and T^2 statistics, the T^2 statistic is defined as follows

$$T_i^2 = [t_1, t_2, \cdots, t_q]\Lambda^{-1}[t_1, t_2, \cdots, t_q]^T \tag{5}$$

where Λ is the diagonal matrix of the eigenvalues associated with the retained PCs, T_i^2 is the T^2 statistic value of the ith sample, the SPE statistic of the ith sample is defined as follows

$$SPE_i = \| \phi(x_i) - \widehat{\phi}(x_i) \|^2 = \| \phi(x_i) - \phi(x_i)VV^T \|^2 = k(x_i, x_i) - TT^T \tag{6}$$

where $V = [v_1, v_2, \cdots, v_q]$, which satisfies $V^TV = I$, I is identity matrix. Because KPCA assumes that the mapped data meets Gaussian distribution, SPE and T^2 statistics respectively satisfy χ^2 distribution and F-distribution and their confidence limits can be calculated using their distributions. After modeling by KPCA and calculating the confidence limits, process monitoring is carried out by calculating SPE and T^2 statistics of each sample and comparing them with their confidence limits.

3.2 On-Line Process Monitoring Using KPCA and NN

From Eqs. (5) and (6), we can see that if KPCA-based method is used to monitor process on-line, it can be results in two problems. On the one hand, the load of calculation is largely increased. To statistically populate the covariance matrix, the required number of observations is at least approximately 10 times the dimensionality of the observation space [7]. Thus the more process variables there are, the larger number of samples, N, is required. While the kernel function must be calculated for N times, and the calculation of addition and multiplication is performed for about qN times to obtain all PCs according to Eqs. (4). In many processes especially in complex industrial processes, there are many process variables which have influence on products, N is very large and the load of calculation of monitoring indices is too large to finish on-line. On the other hand, the N observations for modeling must be stored as the loading vectors according to Eqs. (4), it occupies much memory to store modeling samples.

Through the analysis above, the main disadvantage of KPCA is that the nonlinear loading vectors need to be expressed by the N observations for modeling and largely increase the calculation. The paper introduces NN to capture the nonlinear loading vectors. Assume the nonlinear function represented by NN is $f(x)$, then the PCs of the ith sample are $T = f(x_i)$. The conceptual diagram of KPCA and NN is show in Fig.1.

From Fig.1, NN is used to build the relationship between raw space and KPCA space. After NN training, NN replaces conventional loading vectors of KPCA and the PCs are computed on-line by NN rather than by the loading vectors, which greatly reduces computation comparing with Eqs. (4). The SPE and the T^2 statistics can be expressed as

$$T_i^2 = [t_1, t_2, \cdots, t_q]\Lambda^{-1}[t_1, t_2, \cdots, t_q]^T = T\Lambda^{-1}T^T = f(x_i)\Lambda^{-1}f(x_i)^T \tag{7}$$

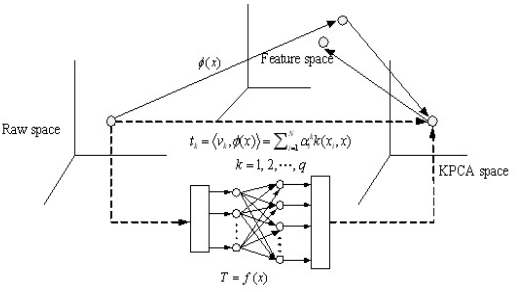


Fig. 1. Conceptual diagram of KPCA and NN

$$SPE_i = k(x_i, x_i) - TT^T = k(x_i, x_i) - f(x_i)f(x_i)^T \tag{8}$$

In the conventional KPCA-based monitoring, the SPE statistic is always positive according to formula (7). While in the monitoring method based on KPCA and NN, since the SPE statistic is associated with NN model, the $f(x_i)$ may be so large that $f(x_i)f(x_i)^T > k(x_i, x_i)$ when special events entering the system, that is to say the SPE statistic is sometimes negative. The SPE representing normal operation varies from 0 to the confidence limit. In order to accord with the monitoring custom of field operators, if the SPE is negative, the paper adds its absolute value on the confidence limit to show the status of violation.

The whole monitoring procedure is divided into two steps, off-line modeling and on-line monitoring. In the off-line modeling step, the raw process data under normal operation is firstly mapped into feature space, then PCs are extracted in feature space using PCA, lastly NN is used to set up the relationship between raw process data and the PCs. After off-line modeling, the NN acts as model in on-line monitoring, raw sample data is input into NN to get PCs and then the monitoring indices are easily calculated to evaluate the process operation.

4 Cases Study

TEP is created by the Eastman Chemical Company to provide a realistic industrial process for evaluating process control and monitoring method, it has been widely used as a benchmark simulation to compare various monitoring methods. The process consists of five major units: a reactor, condense, compressor, separator, stripper. TEP has 52 process variables and is set 21 fault modes. Its detailed introduction and process flow sheet consult [7].

Collect 960 samples under normal operation to model off-line. Firstly, choose a kernel function to capture the nonlinear characteristic of the system, through trial and error, the Gaussian kernel is adopted. Carry out KPCA and get PCs, 40 PCs are selected. Then a 52-60-40 three-layer NN is built to approximate the relationship between PCs and raw process data. There are many ways to increase the training rate, the paper adopts the method of decreasing learning rate and adding a momentum disturbance term. After the NN training, the

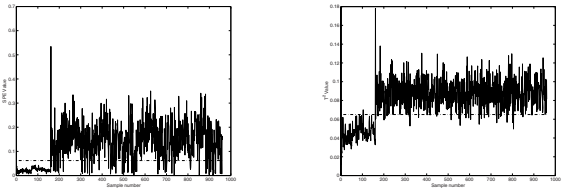


Fig. 2. KPCA and NN monitoring results for fault 4

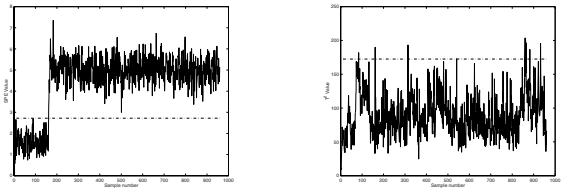


Fig. 3. PCA monitoring results for fault 4

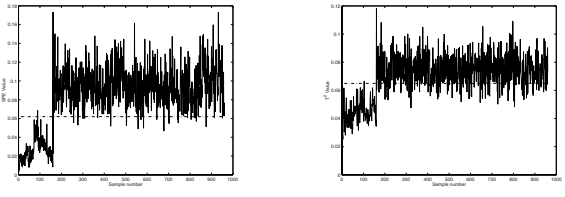


Fig. 4. KPCA monitoring results for fault 4

on-line monitoring is implemented. Taking the 4th fault case for example, the performances of the method comparing with that of PCA and KPCA are shown in Fig.2, Fig.3 and Fig.4.

Fault 4 involves a step change in the reactor cooling water inlet temperature, then the temperature of the reactor will suddenly rise later, while the other variables remain steady, therefore it is difficult to detect. From the monitoring figures shown, the SPE statistic of PCA-based method performs best, the normal condition and the fault condition are separated distinctly and there are no fault alarm and no missing detection, however, its T^2 statistic can hardly detect the fault, that is to say there is one monitoring index worth relying on. Though the SPE and the T^2 charts of KPCA both show a good result, there are 960 samples totaling 960×52 floating point numbers wanted to store in computer, furthermore field computers must calculate the kernel function for 960 times and addition and multiplication for about 960×40 times to obtain monitoring indices of one sample, which may result in large delay and makes KPCA incompetent at the on-line monitoring. The monitoring based on KPCA and NN performs best in T^2 chart and the SPE statistic also can reflect the fault on time, in addition compared with KPCA, it only needs storing $60 \times 52 + 60 \times 40 + 100$ floating point numbers and computing transfer function of NN for 100 times.

5 Conclusion

Though KPCA can overcome the disadvantage of PCA that process data must be linear, it can not be used into on-line monitoring for complex industrial process because of large calculation and much memory occupation. The paper introduces NN into the KPCA-based monitoring, where NN is used to model the relationship between KPCA space and raw space and get PCs on-line. The method greatly reduces calculation and storage and achieves on-line monitoring for complex process, the benchmark monitoring simulation shows its validity.

Acknowledgement

This work is supported by the Key Project of Chinese Ministry of Education (NO 105088).

References

1. Venkatasubramanian, V., *et al.*: A Review of Process Fault Detection and Diagnosis. Computers and Chemical Engineering 27(3) (2003) 327-346
2. Schölkopf, B., *et al.*: Nonlinear Component Analysis as a Kernel Eigenvalue Problem. Neural Computation 10(5) (1998) 1000-1016
3. Lee, J.M., *et al.*: Nonlinear Process Monitoring Using Kernel Principal Component Analysis. Chemical Engineering Science 59(1) (2004) 223-234
4. Cho, J.H., *et al.*: Fault Identification for Process Monitoring Using Kernel Principal Component Analysis. Chemical Engineering Science 60(1) (2005) 279-288
5. Choi, S.W., *et al.*: Fault Detection and Identification of Nonlinear Processes Based on Kernel PCA. Chemometrics and Intelligent Laboratory Systems 75(1) (2005) 55-67
6. Jade, A.M.: Feature Extraction and Denoising Using Kernel PCA. Chemical Engineering Science 58(19) (2003) 4441-4448
7. Chiang, L.H., Russell, E.L., Braatz, R.D.: Fault Detection and Diagnosis in Industrial Systems. Springer-Verlag, London (2001)

On-Line Batch Process Monitoring Using Multiway Kernel Independent Component Analysis

Fei Liu and Zhong-Gai Zhao

Institute of Automation, Southern Yangtze University,
Wuxi, 214122, P.R. China
fliu@thmz.com

Abstract. For on-line batch process monitoring, multiway principal component analysis (MPCA) is a useful tool. But the MPCA-based methods suffer two disadvantages: (i) it restricts itself to a linear setting, where high-order statistical information is discarded; (ii) all the measurement variables must follow Gaussian distribution and the objective of MPCA is only to decorrelate variables, but not to make them independent. To improve the ability of batch process monitoring, this paper proposes a monitoring method named multiway kernel independent component analysis (MKICA). By using kernel trick, the new monitoring indices are investigated, which have been mapped into high-dimensional feature space. On the benchmark simulator of fed-batch penicillin production, the presented method has been validated.

1 Introduction

Batch and semi-batch processes play a more and more important role in modern industries. To achieve consistent and reproducible quality, produced quality variables in batch processes are often examined off-line in a laboratory, which makes it difficult to detect when the processes become abnormal. As on-line monitoring tool, multiway principal component analysis (MPCA) has attracted research interesting [1], [2]. However, the PCA-based methods have two qualifications [3]: (i) it restricts itself to a linear setting, where high-order statistical information is discarded; (ii) all measurement variables must follow Gaussian distribution and the objective of PCA is to make variables de-correlation but not independent. Since batch processes are highly nonlinear, time- varying and seriously interconnected with uncertainties, most batch processes don't follow these terms of PCA.

Some extended approaches have been proposed to overcome the drawbacks of PCA. Nonlinear PCA based on neural networks and principal curves are used in batch processes monitoring to overcome the restriction of linear setting [4]. Though nonlinear PCA can get nonlinear batch trajectory and nonlinear principal components (PCs), it usually requires more computation and inevitably leads to convergence to the local minimum during the network training. In addition, the number of nonlinear PCs is difficult to determine. Kernel PCA is a rising

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 951–956, 2006.

technique to deal with the nonlinear relation among variables [5]. Recently, it has been generalized to multiway kernel PCA (MKPCA) in batch process monitoring [6]. On the other hand, to break through the qualification of Gaussian distribution, multiway independent component analysis (MICA) is introduced to batch process monitoring [7]. ICA-based methods look for components that are both statistically independent and non-Gaussian rather than just uncorrelated and Gaussian distribution. Also, kernel ICA has the same attractive aspect of kernel PCA that it needs no optimizing calculation [8].

This paper introduces a novel on-line batch process monitoring method based on multiway kernel ICA (MKICA), which integrates the ability of kernel function to deal with nonlinear relation and that of ICA to separate non-Gaussian sources. The statistical indices of ICA-based monitoring are extended to high-dimensional feature space, a series of new monitoring indices are proposed. As an example, the presented monitoring method is applied for a fed-batch penicillin production.

2 Statistical Monitoring

2.1 Multiway KICA (MKICA)

In several ($i = 1, 2, \cdots, I$) similar batch process runs, variables $j = 1, 2, \cdots, J$, are measured at time intervals $k = 1, 2, \cdots, K$, throughout every batch. The vast data will be arranged into a three-dimensional matrix $\underline{X} \in R^{I \times J \times K}$. There are two approaches to construct the two-dimensional matrix X by unfolding the three-way data, i.e. time-wise unfolding method and batch-wise unfolding method[1], [9]. The first approach is adopted for MKICA, because the value of JK is very large in batch processes and the load of calculation is great if the second approach is used. Assume the two-way data $X \in R^{I \times JK}$ and the nonlinear mapping function is $\{\phi\} : x_i \longrightarrow \phi(x_i)$. Thus the mapped data is $\phi(x_1), \phi(x_2), \cdots, \phi(x_I)$, denote $P = [\phi(x_1), \phi(x_2), \cdots, \phi(x_I)]$, for simplicity, assume P have been centered in feature space and the paper doesn't deal with the centering problem. The covariance matrix of P is $S^\phi = (1/I)PP^T = (1/I)\sum_{i=1}^{I} \phi(x_i)\phi(x_i)^T$ and the Gram matrix of P is $R = P^T P$, where $R_{ij} = \phi(x_i)^T \phi(x_j) = k(x_i, x_j)$. Carry out eigen-decomposition on R, the first m eigenvectors $\gamma_1, \gamma_2, \cdots, \gamma_m$ and the associated eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_m$ are obtained, where $\lambda_1 > \lambda_2 > \cdots > \lambda_m$. The eigenvectors and the corresponding eigenvalues satisfy $R\gamma_i = P^T P\gamma_i = \gamma_i\lambda_i$, therefore $PP^T(P\gamma_i) = (P\gamma_i)\lambda_i$, the eigenvalues of S^ϕ are λ_i/I and the eigenvectors of S^ϕ are $P\gamma_i$. Because $(P\gamma_i)^T P\gamma_i = \lambda_i\gamma_i^T\gamma_i = \lambda_i$, the normalized eigenvectors are $\beta_i = (1/\sqrt{\lambda_i})P\gamma_i$. Given $\Lambda = diag(\lambda_1, \lambda_2, \cdots, \lambda_m)$ and $V = [\gamma_1, \gamma_2, \cdots, \gamma_m]$. The whitened mapped data can be expressed as follows:

$$\begin{aligned} z &= Q^\phi\phi(x) = \sqrt{I}\Lambda^{-\frac{1}{2}}A^T\phi(x) \\ &= \sqrt{I}\Lambda^{-\frac{1}{2}}\Lambda^{-\frac{1}{2}}V^T P^T\phi(x) = \sqrt{I}\Lambda^{-1}V^T P^T\phi(x) \\ &= \sqrt{I}\Lambda^{-1}V^T[k(x_1, x), k(x_2, x), \cdots, k(x_I, x)]^T = \sqrt{I}\Lambda^{-1}V^T R_x \end{aligned} \tag{1}$$

where $R_x = P^T\phi(x) = [k(x_1,x), k(x_2,x), \cdots, k(x_I,x)]^T$ and Q^ϕ is the whitening matrix. According to [7], the Fast ICA algorithm to find the mixing matrix B is as follows:

Step 1. Take a random initial vector B_i of unit norm.
Step 2. Let $B_i = E\{zg(B_i^T z)\} - E\{g^{'}(B_i^T z)\}B_i$, where g is the first derivative and g' is the second derivative of G function, where G takes the form of $G(u) = (1/a_1)logcosh(a_1 u)$ or $G(u) = exp(-a_2 u^2/2)$ or $G(u) = u^4$, and the a_1, a_2 are coefficients set as $1 \le a_1 \le 2$, $a_1 \approx 1$.
Step 3. Orthogonalize B_i: $B_i = B_i - \sum_{j=1}^{i-1}(B_i^T B_j)B_j$.
Step 4. Normalize: $B_i = B_i / \parallel B_i \parallel$.
Step 5. If B_i has converged, set $i = i + 1$ and output the vector B_i, then go back to Step 1, or else go back to Step 2.

In the above algorithm, B_i is the ith vector of B and the algorithm is iterative until the whole B is obtained, then the demixing matrix W and the independent components matrix S can be calculated by the relations among them mentioned above. The separation matrix is $W = B^T Q^\phi = B^T\sqrt{I}\Lambda^{-1}V^T P^T$ and the ICs can be obtained by:

$$s = W\phi(x) = B^T\sqrt{I}\Lambda^{-1}V^T P^T\phi(x) = B^T\sqrt{I}\Lambda^{-1}V^T R_x. \tag{2}$$

2.2 On-Line Batch Monitoring Using Multiway KICA

Because the Euclidean norm of the ith row is $(L_2)_i = \parallel W_i \parallel = \parallel B_i \parallel \cdot \parallel \sqrt{I}\Lambda^{-1}V^T P^T \parallel$ and the P is unknown [10], it is difficult to order the rows of W in MKICA. Considering the right of the above equation, the paper orders the rows of B^T instead of ordering the rows of W, and decides the number of ICs according to the explanation ratio of the selected rows of B^T to the sum of B^T.

Denote the matrix W_d, W_e constituted by the selected rows of W and the remaining rows of W, respectively, a reduced matrix is constructed by $B_d = (W_d Q^{-1})^T$ which also can be obtained by selecting the columns from B whose indices correspond to the indices of the rows selected form W, and the remaining columns of B constitute the matrix B_e.

In the high-dimensional feature space, the monitoring indices cannot be directly calculated. The paper extends the monitoring indices in ICA-based monitoring to feature space using kernel trick. For sample data $x_{new}(k)$ in time instant k, new independent data vector, $s_{newd}(k)$ and $s_{newe}(k)$, can be obtained by:

$$\widehat{s}_d(k) = W_d\phi(x_k) = B_d^T\sqrt{I}\Lambda^{-1}V^T P^T\phi(x_k) = \sqrt{I}B_d^T V^T R_{x_k} \tag{3}$$

$$\widehat{s}_e(k) = W_e\phi(x_k) = B_e^T\sqrt{I}\Lambda^{-1}V^T P^T\phi(x_k) = \sqrt{I}B_e^T V^T R_{x_k} \tag{4}$$

where $R_{x_k} = P^T\phi(x_k) = [k(x_1,x_k), k(x_2,x_k), \cdots, k(x_I,x_k)]$, then the I^2 and I_e^2 statistics can be defined as follows:

$$I^2(k) = \widehat{s}_d(k)^T\widehat{s}_d(k) = IR_{x_k}^T V\Lambda^{-1}B_d B_d^T\Lambda^{-1}V^T R_{x_k} \tag{5}$$

$$I_e^2(k) = \widehat{s}_e(k)^T \widehat{s}_e(k) = I R_{x_k}^T V \Lambda^{-1} B_e B_e^T \Lambda^{-1} V^T R_{x_k} \tag{6}$$

The mapped data can be estimated by:

$$\begin{aligned}\widehat{\phi}(x_k) &= (Q^{\phi})^{-1} B_d W_d \phi(x_k) = (\sqrt{I}\Lambda^{-1} V^T P^T)^{-1} B_d B_d^T \sqrt{I} \Lambda^{-1} V^T P^T \phi(x_k) \\ &= P^{-T} V^{-T} \Lambda B_d B_d^T \Lambda^{-1} V^T R_{x_k}\end{aligned} \tag{7}$$

Then the SPE statistic can be defined as follow:

$$\begin{aligned}SPE_k &= (\phi(x_k) - \widehat{\phi}(x_k))^T (\phi(x_k) - \widehat{\phi}(x_k)) \\ &= \phi(x_k)^T \phi(x_k) - \phi(x_k)^T \widehat{\phi}(x_k) - \widehat{\phi}(x_k)^T \phi(x_k) + \widehat{\phi}(x_k)^T \widehat{\phi}(x_k) \\ &= k(x_k, x_k) - R_{x_k}^T R^{-1} V^{-T} \Lambda B_d B_d^T \Lambda^{-1} V^T R_{x_k} - R_{x_k}^T V \Lambda^{-1} B_d B_d^T \Lambda \ldots \\ &\quad V^{-1} R^{-1} R_{x_k} + R_{x_k}^T V \Lambda^{-1} B_d B_d^T \Lambda V^{-1} R^{-1} V^{-T} \Lambda B_d B_d^T \Lambda^{-1} V^T R_{x_k}\end{aligned} \tag{8}$$

Because the observations don't satisfy some specific distribution, the monitoring indices also don't meet some fixed distribution, and the confidence limits of them at each sample time can not be determined directly from a particular approximate distribution as shown in [3]. Instead, a method is given, which firstly estimates the probability density function of monitoring indices in normal operation using Parzen windows and then calculates the confidence limits according to the distribution function [3]. After the model and confidence limits of monitoring indices are obtained, the monitoring of the process is realized by using formula (5), (6) and (8).

3 Cases Study

3.1 Process Description

A modular simulator (PenSim v2.0) for fed-batch fermentation has been developed by Birol G., *et al.* [11] that can simulate concentration of biomass, CO2, hydrogen ion, penicillin, carbon source, oxygen and heat generation under various operation conditions. The process can be seen in Fig. 1, which consists of one main fermenter, where a small amount of biomass and substrate are added at first and the substrate is going to be fed in unceasingly when most of the initially added substrate has been consumed by the microorganisms.

3.2 On-Line Monitoring

The initial conditions of the penicillin process and the 10 selected variables may consult [6]. By adding Gaussian variation to the simulation input data to mimic the process variations in the normal operating conditions and adding the measurement noise to each variable, the simulation data of 50 batches in normal operation condition is available.

The duration of each batch is 400h and the sample interval time is 1h, the three-way sample data is $\underline{X} \in R^{50\times10\times400}$, unfold it to $X \in R^{50\times4000}$ and normalize X with its means and the standard deviation of every column. Through

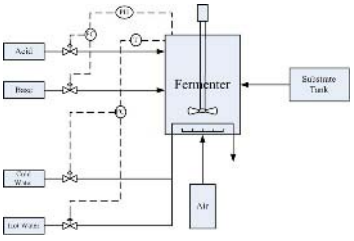


Fig. 1. Penicillin fermentation process

trial and error, a two-order polynomial kernel is found appropriate to capture the nonlinearity of the system and used as the mapping function. Select ten independent components and then calculate the matrices, W_d, W_e, B_d and B_e of the MKICA. Normal processes are monitored by the model to calculate the confidence limits of the statistics at each sample.

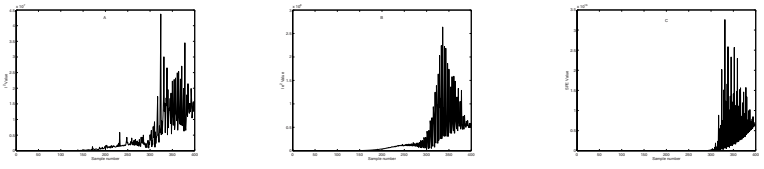


Fig. 2. On-line monitoring charts for (A) I^2 Value (B) I_e^2 Value (C) SPE Value

After the off-line modeling, a fault is imposed to a batch to test the monitoring performance of MKICA in penicillin. As done in [6], a 15% step-decrease in the substrate feed rate is introduced at 55h and retained until the end of fermentation. The monitoring results are shown in Fig 2 and Fig 3.

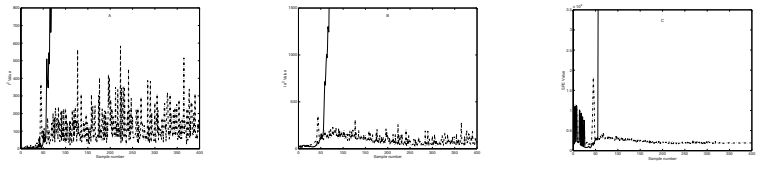


Fig. 3. Zooming in the data of monitoring charts around the control limits

From Fig 3, the I^2 statistic breaks the control limit at sample 57, I_e^2 statistic breaks the limits at sample 56 and SPE does at sample 56. The monitoring indices rise sharply at the fault time and the broken time accords with the sample time the fault introduced. By means of the monitoring charts, it is easy to detect the fault in time. While the MICA-based method doesn't detect the fault until 70h and the indices change slowly in fault time according to [7], the MKICA-based method performs better in fault detection.

4 Conclusion

In order to overcome the two disadvantages of MPCA-based monitoring, the paper introduces an MKICA-based monitoring method . The off-line modeling and on-line monitoring is discussed in detail, and the simulation results reflect that the MKICA is able to detect significant deviation in the batch process. Because the character of batch, the MKICA-based method can provide an effective and well-suited tool for batch monitoring. Though the method developed performs well, the further research is essential to reduce the load of calculation and carry out fault isolation when fault has been detected.

Acknowledgement

This work is supported by Key Project of Chinese Ministry of Education (NO 105088) and by Program for New Century Excellent Talents in University (NCET).

References

1. Nomikos, P., MacGregor, J.: Multivariate SPC Chart for Monitoring Batch Processes. Technometrics 37(1) (1995) 41-59
2. Nomikos, P., MacGregor, J.: Monitoring Batch Processes Using Multi-way Principal Component Analysis. AIChE J. 40(8) (1994) 1361-1375
3. Martin, E. B., Morris, A. J.: Non-parametric Confidence Bounds for Process Performance Monitoring Charts. Journal of Process Control 6(6) (1996) 349-358
4. Dong, D., McAvoy, T.J.: Nonlinear Principal Component Analysis Based on Principal Curves and Neural Networks. Computers and Chemical Engineering 20(1) (1996) 65-78
5. Schölkopf, B., Smola, A., Müller, K.R.: Nonlinear Component Analysis as a Kernel Eigenvalue Problem. Neural Computation 10(5) (1998) 1000-1016
6. Lee, J.M., Yoo, C.K., Lee, I.B.: Fault Detection of Batch Processes Using Multiway Kernel Principal Component Analysis. Computer and Chemical Engineering 28(9) (2004) 1837-1847
7. Yoo, C.K., Lee, J.M., *et al.*: On-line Monitoring of Batch Processes Using Multiway Independent Component Analysis. Chemometrics and Intelligent Laboratory Systems 71(2) (2004) 151-163
8. Yang, J., Gao, X., *et al.*: Kernel ICA: An Alternative Formulation and Its Application to Face Recognition. Pattern Recognition 38(10) (2005) 1784-1787
9. Wold, S., Kettanhe, N., Friden, H., Holmberg, A.: Modelling and Diagnostics of Batch Processes and Analogous Kinetic Experiments. Chemometrics and Intelligent Laboratory Systems 44(2) (1998) 331-340
10. Cardosom, J.F., Soulomica, A.: Blind Beamforming for Non-Gaussian Signals. IEEE Proc. F. 140(6) (1993) 362-370
11. Birol, G., Undey, C., Cinar, A.: A Modular Simulation Package for Fed-batch Fermentation: Penicillin Production. Computers and Chemical Engineering 26(11) (2002) 1553-1565

Tool Wear Monitoring Using FNN with Compact Support Gaussian Function

Hongli Gao[1], Mingheng Xu[1], Jun Li [2], and Chunjun Chen[1]

[1] School of Mechanical Engineering, Southwest Jiaotong University, Chengdu, Sichuan 610031 China
swghl@163.com
[2] School of Economics & Management, Southwest Jiaotong University, Chengdu, Sichuan 610031 China

Abstract. A novel approach of tool wear monitoring based on localized fuzzy neural networks with compact support Gaussian basis function (CSGFFNN) was proposed to improve classification accuracy of tool states and solve the problems of slow computing speed of BP neural networks. By analyzing cutting forces signals, acoustic emission signals and vibration signals in time domain, frequency domain, and time-frequency domain, a series of features that sensitive to tool states were selected as inputs of neural networks according to synthesis coefficient. The nonlinear relations between tool wear and features were modeled by using CSGFFNN that constructed and optimized through fuzzy clustering and an adaptive learning algorithm. The experimental results show that the monitoring system based on CSGFFNN is provided with high precision, rapid computing speed and good multiplication.

1 Introduction

Tool wear is an inevitable result of the metal cutting process, the change of tool states can affect the process significantly, so it is essential to develop tool condition monitoring system that can improve the dimensional accuracy of workpiece and guarantee a reliable and automatic cutting process.

Tool condition monitoring (TCM) has been investigated since 1980s, lots of scholars utilize various techniques to predict tool states including simple decision logic, pattern recognition, fuzzy logic[1], GDMH [2], and neural networks [3][4]etc. Compare with other techniques, neural networks is the most popular and successful method. At present, above 70% scholars use BP neural networks (BPNN) to model the relations between tool states and signal features extracted from acoustic emission sensor, vibration sensor, dynamometer, power (motor current) etc[5][6][7], and make great progress. However, the industrial tool condition monitoring system based on BPNN has not been designed owing to major disadvantages as follows: slow computing speed that not adapt to on-line modeling, and bad multiplication that not adapt to diversity of machining condition [8].

Aiming at above disadvantages of BPNN, this paper use a new class of localized fuzzy neural networks with compact support Gaussian basis function to predict tool

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 957 – 963, 2006.

wear and fuse multisensor information from vibration signal, acoustic emission signal and cutting forces signal. A series of features were extracted by different signal processing method, and only 10 features that most sensitive to tool wear were selected as inputs of network according to sort order of synthesis coefficients. The highly nonlinear functional relations between tool states and features were modeled by using CSGFFNN which was constructed and optimized through fuzzy clustering and an adaptive learning rule. The experiment results indicate that the approach proposed in the paper can reduce the cost and complexity of the monitoring system, improve the on-line computing speed and classify accuracy of tool states.

2 FNN with Compact Support Gaussian Basis Function

2.1 Compact Support Gaussian Basis Function

A univariate compact support Gaussian basis function is defined as:

$$N(u,v,x)=\begin{cases}\exp(1-\dfrac{v^2}{v^2-(u-x)^2}) & x\in(u-v,u+v)\\ 0 & x\notin(u-v,u+v)\end{cases} \tag{1}$$

Where u is the center of univariate Gaussian basis function, v is radius of univariate Gaussian basis function.

Multivariate compact support Gaussian basis functions are formed by taking the tensor product of n univariate compact support Gaussian basis function:

$$N(U,V,X)=\begin{cases}\prod_{i=1}^{n}\exp(1-\dfrac{v_i^2}{v_i^2-(u_i^2-x_i^2)}) & X\in(U-V,U+V)\\ 0 & X\notin(U-V,U+V)\end{cases} \tag{2}$$

Where U is the n-dimensional vector denoting the centre of N (U, V, X), V is the n-dimensional vector denoting the radius of N (U, V, X), and X is the n-dimensional input vector.

2.2 Structure of CSGFFNN

The structure of CSGFFNN is shown in Fig. 1. When the neurons in hidden layer use compact support Gaussian basis function, each input only activates a small number of basis functions, and it is the knowledge stored in this region of the network which contributes to its response.

It is the localized Gaussian basis functions which endow the neural networks with desirable properties, the network's output is linearly dependent on these basis functions and so their contribution can easily be understood, because each basis function can be interpreted as a set of fuzzy linguistic rules. The n-dimensional input

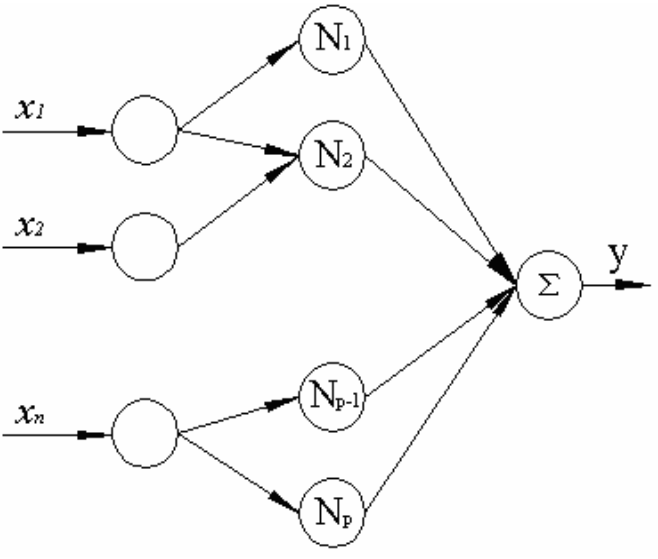


Fig. 1. The structure of CSGFFNN

to the network is denoted by X, the output of the j^{th} Gaussian basis function by N_j, and the network's output by y. Hence the output of the CSGFFNN is given by:

$$y = \sum_{j=1}^{p} w_j N_j(U_j, V_j, X) \tag{3}$$

where w_j is the weight corresponding to the j^{th} basis function, p is the number of neuron in the hidden layer, U_j is the centre of N_j (U_j, V_j, X), V_j is the radius of N_j (U_j, V_j, X). The j^{th} Gaussian basis function N_j is generated by multiplying n univariate basis function N_{ij} (u_i, v_i, x_i)

$$N_j(U_j, V_j, X) = \begin{cases} \exp(\sum_{i=1}^{n}(1 - \frac{v_{ij}^2}{v_{ij}^2 - (u_{ij} - x_i)^2}) & x_i \in (u_{ij} - v_{ij}, u_{ij} + v_{ij}) \\ 0 & x_i \notin (u_{ij} - v_{ij}, u_{ij} + v_{ij}) \end{cases} \tag{4}$$

Note that each multivariate compact support Gaussian basis function is calculated using a different set of univariate compact support Gaussian basis function.

2.3 Adaptive Learning Algorithm of CSGFFNN

This paper adopts an adaptive learning algorithm to train CSGFFNNs. The number of basis function in hidden layer is firstly determined by fuzzy clustering. The parameters U_j and V_j are updated together with weight vector ω_j to minimize the mean square output error given by:

$$E = \frac{1}{2}(y_i - \hat{y}_i)^2 \tag{5}$$

Where y_i is the practical output of CSGFFNN, $\hat{y}_i$ is the desired output of CSGFFNN.

The weight update Δw_j, the adjustment of the centre ΔU_j and the radius ΔV_j of j^{th} Gaussian basis function in hidden layer can be obtained by employing instantaneous gradient descent rules as follows:

$$\Delta w_j = -\alpha \frac{\partial E}{\partial w_j} = -\alpha (y_i - \hat{y}_i) N(U_j, V_j, X_i) \tag{6}$$

$$\Delta U_j = -\beta \frac{\partial E}{\partial U_j} = -\beta w_j (y_i - \hat{y}_i) \frac{\partial N(U_j, V_j, X_i)}{\partial U_j} \tag{7}$$

$$\Delta V_j = -\delta \frac{\partial E}{\partial V_j} = -\delta w_j (y_i - \hat{y}_i) \frac{\partial N(U_j, V_j, X_i)}{\partial V_j} \tag{8}$$

where α, β and δ are learning rates given by small positive constants.

3 Experimental Design

3.1 Experimental Setup

The schematic diagram of the tool wear monitoring system is shown in Fig. 2. The experiments were carried out on a Cincinnati Milacron Sabre 500 Vertical Machining Centre, a Kistler 9257B dynamometer was installed on the table of the machine, and a vibration sensor (type B&K 4370) and a piezoelectric acoustic sensor transducer (type RNT-200) were mounted directly on the side face of workpiece.

When a milling cutter (ö80mm) with six YT15 inserts (type VX-81032)was used to machine a 45# middle-carbide steel workpiece(80mm×50mm×25mm) that meet the measuring need of dynamometer in the milling operation, three orthogonal cutting forces signals *Fx*, *Fy*, *Fz* were measured and amplified by a multichannel charge amplifier (type 5019B), and through a A/D converter board (type Amplcon PC266) to computer. Sampling rate is 16 kHz and sampling time is 2s. In the meantime, the AE signals and vibration signals were amplified and filtered with corresponding transducer, and then sent to computer via an A/D converter board type Jv52014, sampling rate is 2 MHz, and sampling length is 256k.

3.2 Experimental Method

In order to acquire enough datasets that CSGFFNN need for modeling the relations between tool states and features extracted from three sensors, this paper adopts full factors combination experimental method. The cutting parameters are shown in Table.1, and tool inserts were artificially worn in a tool grinder beforehand according to the classification of tool flank wear shown in Table 2. The lengths of each insert wear land were measured at intervals throughout the tests by means of a tool microscope.

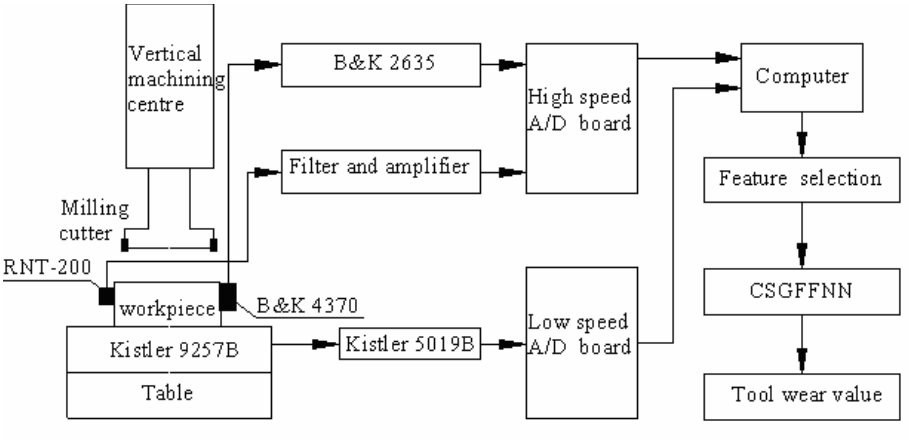


Fig. 2. The schematic diagram of the tool wear monitoring system

Table 1. Experimental conditions

Cutting speed (m/min)	600	900	1200
Feed rate (mm/r)	0.2	0.4	0.8
Depth of cut (mm)	0.5	1	1.5

Table 2. Classification of tool flank wear

Classification	M1	M1	-	-	-	M10	M11
Tool wear value (mm)	0	0.05	-	-	-	0.45	0.5

4 Feature Selection and Experiment Results

4.1 Feature Selection

It can be seen from the measured signals in the metal cutting process that cutting forces change clearly with progress of tool wear, however, AE signals and vibration signals in time domain change slightly. In order to extract more features that sensitive to tool wear, different signal processing methods were adopted such as time domain analysis, frequency analysis, and wavelet transformation. Up to now, how to select appropriate features is always the most difficult problem for prediction of tool wear, especially for automatic selection, so a new approach using synthesis coefficient was introduced to solve the problem. The synthesis coefficient is defined as:

$$C_k^i = \frac{\frac{x_{\max}}{i}\sum_i (x_i - \bar{x})(y_i - \bar{y})}{x_{\min}\sqrt{\frac{1}{i}\sum_i (x_i - \bar{x})^2}\sqrt{\frac{1}{i}\sum_i (y_i - \bar{y})^2}} \tag{9}$$

Where $\bar{x}$ and $\bar{y}$ are mean values of x and y, respectively; x_i are i^{th} features extracted from signals; y_i are i^{th} tool wear value; $x_{\max}$ is maximum of x_i; $x_{\min}$ is minimum of x_i; C_k^i is the synthesis coefficient. The synthesis coefficient C_k^i represents a change trend between tool wear and features, so we choose two features for each signal according to sort ascending of the coefficient C_k^i. Finally, 10 features are selected as inputs of CSGFFNN, the results are shown in Table 3.

Table 3. Selected features through synthesis coefficient

signal	F_x		F_y		F_z		AE		vibration	
feature	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
C_k^i	9.09	2.74	13.26	2.66	2.12	1.83	5.47	5.08	4.32	3.08

Where, x_1, x_3, x_7, x_8, x_9, x_{10} are wavelet coefficients of signals in some frequency bands, x_2 and x_4 are RMS values, x_5 is kurtosis value, and x_6 is variance.

4.2 Experiment Results

In the section, the performance of the proposed TCMS by using CSGFFNN is given and compared with that of traditional BPNN. A total of 891 datasets acquired in the experiment were studied. 164 samples were cautiously selected as the training samples, and 128 samples were picked as testing samples. For CSGFFNN, the number of neurons in input layer and in output layer is 10 and 1, respectively. The number of neurons in hidden layer determined by fuzzy clustering is 13. The learning rates α, β and δ is initially 0.1, 0.05 and 0.1. The ideal MSE of NN is 0.002, and the parameters U, V is obtained by proposed adaptive learning rule. Fig.3 show the recognized results of tool wear monitoring system founded on BPNN and CSGFFNN. As we can see, the predicted results based on CSGFFNN has a better

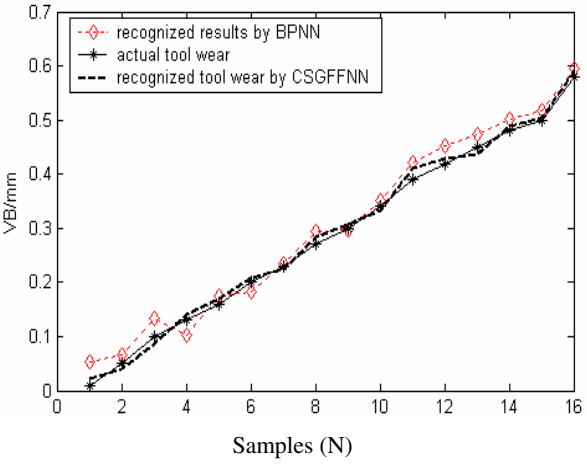


Fig. 3. Predicted tool wear of TCMS by BPNN and by CSGFFNN

performance than that based on BPNN, the mean classifying error of CSGFFNN is 0.023, and mean error of BPNN is 0.058. Further, an arbitrary input vector X of CSGFFNN only activates a few multivariate basis functions compared with BPNN, so computational cost is largely reduced. This characteristic is crucial to on-line classification of tool wear.

5 Conclusions

In the paper, AE signals, vibration signals and cutting forces signals have been analyzed by different signal processing method, a series of features are automatically and successfully selected according to synthesis coefficient, and a new approach of tool wear monitoring based on CSGFFNN has been investigated in this work, which can realize prediction of tool states. Furthermore, we have demonstrated that fuzzy clustering and an adaptive learning algorithm could be utilized to optimize the structure of neural networks. The experimental results show the TCMS based on proposed methods improve the predicted accuracy of tool wear, adapt to real-time monitoring and provide better performance than BPNN.

References

1. Li, X.L., Li, H.X., Guan, X.P., Du, R.: Fuzzy Estimation of Feed-Cutting Force from Current Measurement - A Case Study on Tool Wear Monitoring. IEEE Trans. Systems, Man, and Cybernetics Part C: Applications and Reviews **34** (4) (2004) 506-512
2. Jiaa, C.L., Dornfeld, D.A.: A Self-organizing Approach to the Prediction and Detection of Tool Wear. ISA Transactions **37** (1998) 239-255
3. Yao, Y.X., Li, X., Yuan, Z.J.: Tool Wear Detection with Fuzzy Classification and Wavelet Fuzzy Neural Network. Int. Journal of Machine Tools & Manufacture 39 (1999) 1525-1538
4. Sun, J., Rahman, M., Wong, Y.S., Hong, G.S.: Multiclassfication of Tool Wear with Support Vector Machine by Manufacturing Loss Consideration. International Journal of Machine Tools & Manufacture **44** (2004) 1179-1187
5. Dimla Sr, D.E.: Sensor Signals for Tool-Wear Monitoring in Metal Cutting Operations – A Review of Methods. Int. Journal of Machine Tools & Manufacture **40** (2000) 1073-1098
6. Li, X.L.: A Brief Review: Acoustic Emission Method for Tool Wear Monitoring During Turning. International Journal of Machine Tools and Manufacture **42** (2002) 157-165
7. Li, X., Djordjevich, A., Venuvinod P.K.: Current Sensor-Based Feed Cutting Force Intelligent Estimation and Tool Wear Condition Monitoring. IEEE Transactions on Industrial Electronics **47** (3) (2000) 697 – 702
8. Gao, H.L., Fu, P., Xu, M.H.: Tool Wear Monitoring Based on B-Spline Neural Networks. Mechanical Science and Technology **6** (2005) 751-754

Intelligent Classification of Cutting Tool Wear States

Pan Fu[1] and Anthony D. Hope[2]

[1] Mechanical Engineering Faculty, Southwest Jiao Tong University, Chengdu 610031, China
pfu@home.swjtu.edu.cn
[2] Systems Engineering Faculty, Southampton Institute, Southampton SO14 OYN, U.K.
tony.hope@solent.ac.uk

Abstract. In manufacturing processes, it is very important that the condition of the cutting tool, particularly the indications when it should be changed, can be monitored. Cutting tool condition monitoring is a very complex process and thus sensor fusion techniques and artificial intelligence signal processing algorithms are employed in this study. A unique fuzzy neural hybrid pattern recognition algorithm has been developed which combines the transparent representation of fuzzy system with the learning ability of neural networks. The algorithm has strong modeling and noise suppression ability. These leads to successful tool wear classification under a range of machining conditions.

1 Introduction

Many researchers have published results in the area of automatic tool condition monitoring. An approach was developed for in-process monitoring tool wear in milling using frequency signatures of the cutting force [1]. An analytical method was developed for the use of three mutually perpendicular components of the cutting forces and vibration signature measurements [2]. A tool condition monitoring system was then established for cutting tool-state classification [3]. In another study, the input features were derived from measurements of acoustic emission during machining and topography of the machined surfaces [4]. Li, X etc. showed that the frequency distribution of vibration changes as the tool wears [5]. A new on-line fuzzy neural network (FNN) model with four parts was developed to estimate flank and crater wear [6]. On-line and indirect tool wear estimation in turning was realized by using a physical process model describing the influence of cutting conditions on measured process parameters [7]. Two methods using Hidden Markov models, as well as several other methods that directly use force and power data were used to establish the health of a drilling tool [8]. In this study, a new fuzzy neural hybrid pattern recognition algorithm was developed to accomplish multi-sensor information integration and tool wear states classification.

2 Tool Condition Monitoring System

The tool wear monitoring system for milling operations is composed of four types of sensors, signal amplifying and collecting devices and the main computer, as shown in

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 964 – 969, 2006.

Fig. 1. The power consumption, cutting force (in three perpendicular directions), acoustic emission (AE) and vibration sensors chosen were found to provide healthy time domain signals for tool condition monitoring.

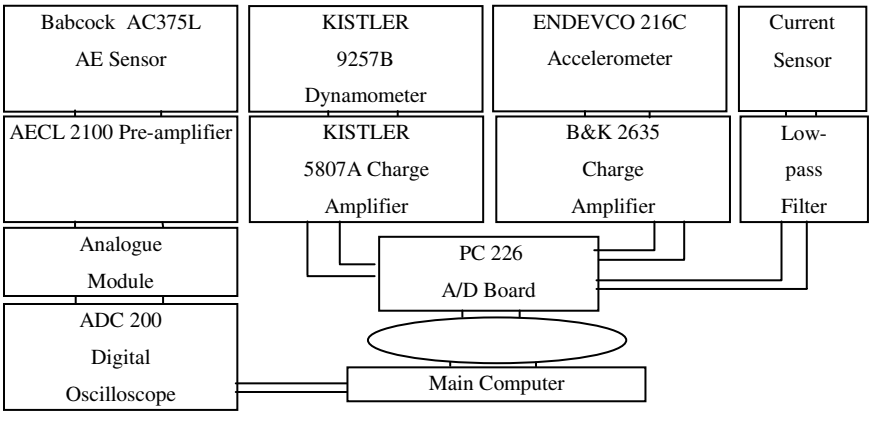


Fig. 1. The tool condition monitoring system

3 Feature Extraction

The original signals have large dimensions and can not be directly used to estimate tool wear value. The purpose of feature extraction is to greatly reduce the dimension of the raw signal but at the same time maintain the tool condition relevant information in the extracted features. This step is the foundation for the pattern recognition process. A typical group of features extracted from the time domain and frequency domain for the further pattern recognition are as follows. Power consumption signal: mean value; AE-RMS signal: mean value, skew and kutorsis; Cutting force, AE and vibration : mean value, standard deviation and the mean power in 10 frequency ranges. Experimental results had shown that these features comprehensively represent the development of tool flank wear.

4 Fuzzy Driven Neural Network

The features of sensor signals can reflect the tool wear states. Theoretical analysis and experimental results show that these features can be regarded as normal distribution fuzzy sets. Fuzzy approaching degree is an index that represents the fuzzy distance between two fuzzy sets (A and B). Approaching degree can be calculated by using different methods. Here the inner and outer products are used.

If $A, B \in \Im(X)$, $A \bullet B = \vee\{A(x) \wedge B(x) : x \in X\}$ is defined as the inner product of A and B and $A \oplus B = \wedge\{A(x) \vee B(x) : x \in X\}$ is defined as the outer

product of A and B. Finally, in the map $N : \Im(X) \times \Im(X) \to [0, 1]$, $N(A,B)$is the approaching degree of A and B.

$$N(A, B) = (A \bullet B) \wedge (A \oplus B)^c \tag{1}$$

In this study, ANNs are connected with a fuzzy logic system to establish a fuzzy driven neural network pattern recognition system. Its principle is shown in the following figure.

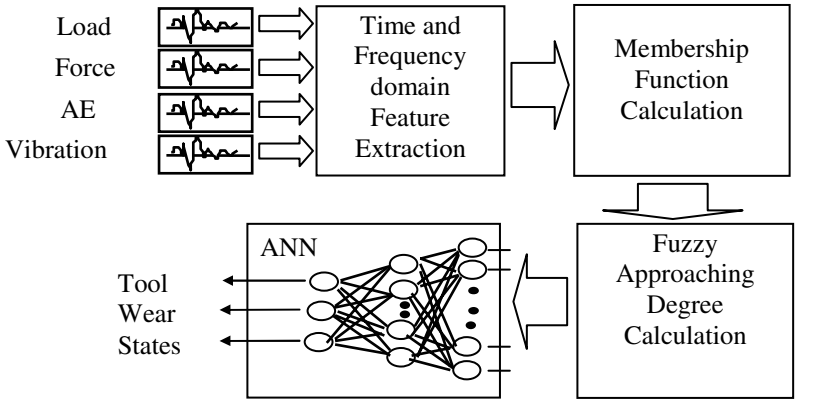


Fig. 2. The fuzzy driven neural network

Here a back propagation ANN is used to carry out tool wear classification. The approaching degree calculation results are the input of the ANN. The ANN can recognize the patterns of the features corresponding to certain tool wear state by assigning each feature a proper synthesized weight. This enables the tool wear classification process be more reliable.

5 Algebraic Neurofuzzy Networks

A neural fuzzy system has both the transparent representation of a fuzzy system and the learning ability of neural networks. The combination of the rule based representation and adaptive numeric processing can lead to a robust modeling system. Various applications of fuzzy neural integrated systems may be cited [9][10][11]. Many neurofuzzy systems use B-spline (Brown et al. 1995) or Gaussian (Wang. 1994) basis functions [9][11]. Adaptive B-spline based neurofuzzy system uses algebraic operators and B-spline fuzzy membership functions to simplify the overall system, produces more transparent models. Signal features can be treated as fuzzy sets, which can then be represented by fuzzy membership functions. Here, B-spline basis functions, piecewise polynomials of order ***k***, are used to represent fuzzy membership functions. The order and the knot vector determine the smoothness and shape of the basis functions. The knots partition the input space into a series of intervals on which the basis functions are defined. Multivariate B-spline basis functions are formed by taking tensor production

of ***n*** univariate basis functions, where only one univariate function is defined on each input axis. The multivariate basis functions are then defined on a lattice, which is generated from the projection of all the individual knot vectors parallel to the remaining input axes. The relation between signal features and tool wear values can be represented by a fuzzy rule. The union (fuzzy *OR*) of a group of fuzzy rules is called a fuzzy algorithm in which the knowledge of a fuzzy system is stored. So the set of all the confidences c_{ij} (rule confidence matrix) illustrates the complex relation between the input space and the output of the system. To fulfill the fuzzy rule set, functions must be chosen to implement the fuzzy logic functions, *AND, OR, IF* ( ), *THEN* (), etc. Recent research shows that the algebraic operators, *sum* and *product*, can produce smoother output than the traditional truncation operators, *min* and *max* [9][10]. Here, B-splines are used to implement the fuzzy membership functions. Singleton fuzzy sets are used to represent the crisp input. Algebraic operators are chosen to accomplish the fuzzy logic functions and the diffuzzification is realized by using a centre of gravity algorithm, and the rule confidences are normalized. The structure of the neurofuzzy system is shown in Fig. 5. In the figure, the multivariate fuzzy input sets (termed as *basis functions*) are defined on a lattice in the input space. The weight of a basis function is an estimate of the value of the network's output; given that the input lies within the set. The output of the network is linearly dependent on the weight set. This network structure allows an efficient linear learning strategy like *Conjugate Gradient* to be used to adapt the weights for optimal performance. The neurofuzzy system can be a powerful tool for cutting tool condition monitoring. In the training process, for all the signal features of each model (cutting tool with standard wear value), a group of feature values are put into the neurofuzzy network as the training input. A fuzzy rule base is then established to describe the mapping between the systems input and output states. So in the practical condition monitoring process, it can recognize the incoming feature pattern and associate the pattern with different models with corresponding classification confidence.

6 Fusion on Two Levels

To make the tool condition monitoring system more reliable, fusion on two levels is employed in this study. The first level is sensor fusion. The monitoring system is equipped with four kinds of sensors and multi-sensor signal features are fused by the intelligent data processing process. The second fusion is on a higher level: the fusion of two pattern recognition algorithms. It should be noticed that the two proposed algorithms have different characteristics and they can describe the tool wear process from different view points. The former has quite reliable recognition results, but in some cases the confidence of the classification may not be as high as it should. The latter is quite accurate for most circumstances，but exceptionally, where the rule base is not perfectly complete, the system may refuse to classify some individual objects. The authors of this paper argue that by combining the two algorithms to establish a fused pattern recognition process the tool wear classification results can be more reliable and this idea is supported by large amounts of experiment results.

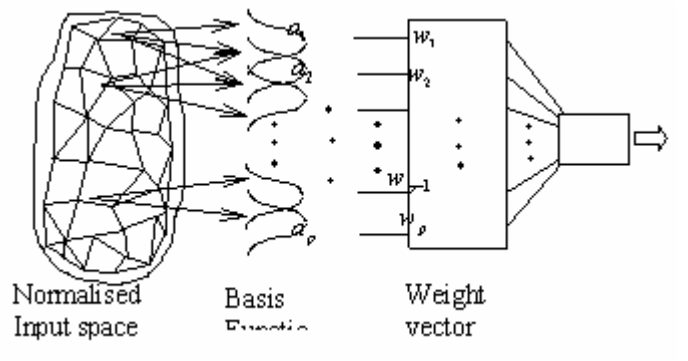


Fig. 3. Algebraic neurofuzzy network

7 Fuzzy Neural Hybrid Pattern Recognition System

The fuzzy neural hybrid pattern recognition system is established by the integration of the fuzzy driven neural network and the algebraic neurofuzzy network. Both the two systems provide the similarities between the object and different models and classify the object to the most similar model with a certain confidence value. These two confidence values are not necessarily equal, but combining them provides a more reliable and accurate result. A threshold is set by considering the difference between the classification confidence values and tool wear values of the two classification results. Should the two pattern recognition processes give different results, the system averages the results when the difference is within threshold and refuses to do the classification if the threshold is exceeded. The failure of the classification shows the incoming data is too noisy or the networks have not been fully trained and need to be improved. By doing this, the reliability of the classification process is improved.

Signal collected under 200 representative cutting conditions have been processed to verify the proposed fuzzy neural hybrid system (Experiments were partly carried out in the Advanced Manufacturing Lab. of Southampton Institute, U.K.). The system showed very good classification accuracy and reliability (successful recognition rate of higher than 97% with confidence of higher than 85%).

8 Conclusions

An intelligent tool condition monitoring system has been established. Tool wear classification is realized by applying a unique fuzzy neural hybrid pattern recognition system. On the basis of this investigation, the following conclusions can be made. Power consumption, vibration, AE and cutting force sensors are applicable for monitoring tool wear in metal cutting process. Many features extracted from time and frequency domains are found to be relevant to the changes of tool wear state. The developed hybrid neurofuzzy networks have a simplified structure and produces better and more transparent models than a general fuzzy system. Armed with the advanced

pattern recognition methodology, the established intelligent tool condition monitoring system has the advantages of being suitable for different machining conditions, robust to noise and tolerant to faults.

References

1. Elbestawi, M. A., Papazafiriou, T. A., Du, R.X.: In-process Monitoring of Tool Wear in Milling Using Cutting Force Signature. Int. J. Machine Tools and Manufacturing, 1 (1991) 55-73
2. Dimla, D. E., Lister, P. M.: On-line Metal Cutting Tool Condition Monitoring. I: Force and Vibration Analyses. Int. J. Machine Tools and Manufacturing. 4 (2000) 739-768
3. Dimla, D. E., Lister, P. M.: On-line Metal Cutting Tool Condition Monitoring. II: Tool-state Classification Using Multi-layer Perceptron Neural Networks. Int. J. of Machine Tools and Manufacturing, 5 (2000) 769-781
4. Wilkinson, P., Reuben, R. L., Jones, J. D. C.: Tool Wear Prediction from Acoustic Emission and Surface Characteristics via an Artificial Neural Network. Mechanical Systems and Signal Processing, 6 (1999) 955-966
5. Li, X., Dong, S., Venuvinod, P. K.: Hybrid Learning for Tool Wear Monitoring. Int. J. of Advanced Manufacturing Technology, 5 (2000) 303-307
6. Chungchoo, C., Saini, D.: On-line Tool Wear Estimation in CNC Turning Operations Using Fuzzy Neural Network Model. Int. J. of Machine Tools and Manufacture, 1 (2002) 29-40
7. Sick, B.: Tool Wear Monitoring in Turning: A Neural Network Application. Measurement and Control 7 (2001) 207-222
8. Ertunc, H. M., Loparo, K. A.: A Decision Fusion Algorithm for Tool Wear Condition Monitoring in Drilling. Int. J. of Machine Tools and Manufacture. 9 (2001) 1347-1362
9. Brown, M., Harris, C. J.: A Perspective and Critique of Adaptive Neurofuzzy Systems Used for Modeling and Control Applications. Int. J. of Neural Systems, 6 (1995) 197-220
10. Brown, M., Harris, C. J.: A Nonlinear Adaptive Controller: A Comparison between Fuzzy Logic Control and Neurocontrol. J. Math. Control and Info.. 8 (1991) 239-265
11. Wang, L. X.: Adaptive Fuzzy Systems and Control Design and Stability Analysis. Prentice Hall, Englewood Cliffs, NJ. (1994)

Neural Networks-Based In-Process Surface Roughness Adaptive Control System in Turning Operations

Julie Z. Zhang[1] and Joseph C. Chen[2]

[1] 37 ITC, University of Northern Iowa, Cedar Falls, IA 50614, USA
julie.zhang@uni.edu
[2] 221 I. Ed. II, Iowa State University, Ames, IA 50011, USA
cschen@iastate.edu

Abstract. Using a back-propagation neural networks algorithm and accelerometer sensor technique, this research developed an in-process surface roughness adaptive control (IPSRAC) system in turning operations. This system not only can predict surface roughness in real time, but can also provide an adaptive feed rate change in finishing turning to ensure the surface roughness can meet requirements.

1 Introduction

Surface roughness is an important indicator of quality of parts produced in turning operations since it may affect the aesthetic appeal of the machined part, the fatigue stress of the assembled part, economic costs, etc. The complex nature of turning decides that surface roughness can be influenced by a number of factors, such as cutting parameters, workpiece characteristics, tool geometry, and tool conditions [1], [2], [3]. If any of these factors shift from design specifications, the end product could vary from the desired specification. Traditionally, standard surface roughness monitoring depends heavily on stylus instruments [2]. However, as an offline quality monitoring technique, this method is a post-process approach as turning operations must be paused; therefore, it does not allow users to take full advantage of the CNC machines' merit. Since the 1980s, research has shifted to online or real time surface roughness prediction, which involves the modeling of surface roughness by incorporating a variety of machining parameters and machining process signals. For example, Lee & Tarng [4] and Ho et al. [2] employed a computer vision system to estimate the surface roughness of turned parts. In applying a neural networks algorithm, online surface roughness recognition models have been developed by including vibration or cutting force signals [1], [5]. Although these systems can recognize surface roughness without stopping the machining operation, they cannot correct the error. Therefore, there is a need to move the research of online surface roughness recognition to the new stage of developing an in-process surface roughness adaptive control (ISRAC) system that is able to automatically recognize a part's surface finish and further adjust process parameters while the finishing turning is in process to ensure the surface quality. In order to develop such an ISRAC system for turning operations, two important components must be considered: a real-time sensor

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 970 – 975, 2006.

technique for collecting cutting process signals related to surface roughness and a decision-making mechanism algorithm. A number of studies demonstrated that vibration signal plays an important role in influencing surface finish in turning operation [1], [6], [7], [8], [9]; therefore, an accelerometer sensor was selected in this study to detect vibration signals. The previous reviewed research also revealed that neural networks are an effective strategy for dealing with the non-linear and multivariable turning processes. In summary, this study attempts to develop an in-process artificial neural network-based surface roughness adaptive control (INNSRAC) system by incorporating real-time 3-D vibration signals. This system will be able to evaluate the surface quality of the turned parts and adaptively provide the feed rate change in order to consistently produce parts that meet specifications, since feed rate is the most significant factor affecting surface finish [1], [2], [5], [9], [10] and [11].

2 Artificial Neural Networks System

A neural network is a parallelly distributed information processing structure that mimics the human brain and the ability of nervous processes to learn from examples or mistakes [10]. Neural networks, based on their biological counterparts, attempt to model the parallel distributed processing nature of the human brain. In the simplest case, one neuron sums the weighted inputs ($x_i w_i$), passes the result through a non-linear transfer function, and provides an output Y_i

$$Y_i = f(\sum_{i=0}^{n-1} w_i x_i - \theta), \tag{1}$$

where θ is the internal threshold of a neuron and f is the non-linear transfer function, which is usually defined as

$$f(x) = \frac{1}{1+e^{-x}}. \tag{2}$$

A neural network follows a networking structure in which a set of artificial neurons are connected to each other through input and output. Each neuron in a layer receives weighted inputs from the neurons in the previous layer. The outputs of this neuron are the inputs to several other neurons in the following layer, which forms a network. Beyond the input and output layers, several other layers of neurons, called hidden layers, might be needed to build an effective neural network that is capable of solving problems.

Among the variety of neural network algorithms, back-propagation (BP) is the most commonly used for pattern recognition due to its simplicity and reasonable speed [11]. The training procedure is iterative, usually involves trial and error for a BPN, and consists of the following steps [12]:

Step 1: Initialize weights and offsets starting from a small random value.
Step 2: Present input variables and output variables to the neural network model.
Step 3: Calculate the actual outputs, y_m.

Step 4: Calculate the error between the output, y_m from the neural network and the desired output, d_m by E

$$E = \frac{1}{2m}\sum_m \sum_n (y_{m,n} - d_{m,n})^2 , \tag{3}$$

where m is the number of neurons in the output layer, and n is the number of training data set. If E is smaller than the required accuracy, then no other learning procedures are needed.

Step 5: If E is larger than the required accuracy, adjust the weights of the networks

Step 6: Repeat steps 3-6 until the error of the entire set is less than the required accuracy.

3 Structure of the INNSAC System

The INNSAC system in Figure 1 includes two subsystems: an in-process neural-network-based surface roughness prediction (INNSRP) subsystem and an in-process neural network-based adaptive parameter control (INNAPC) subsystem. As the turning process starts, the accelerometer sensor simultaneously records the cutting vibration in X, Y, and Z directions. When the turning is in process, the INNSRP subsystem is able to predict the workpiece's surface roughness (R_a^P) based on cutting parameters and the changing trends in cutting vibration. If R_a^P is better than the desired surface roughness R_a^D ($R_a^P < R_a^D$), the turning process will continue. If it is worse ($R_a^P > R_a^D$), the detected surface roughness difference ΔR_a will trigger the INNAPC subsystem to function.

$$\Delta R_a = R_a^P - R_a^D \tag{4}$$

As feed rate is the most significant cutting parameter influencing surface roughness, the INNAPC subsystem will provide the adaptive feed rate change (ΔFr).This output will be sent back to the turning operation in order to create products that meet customer needs.

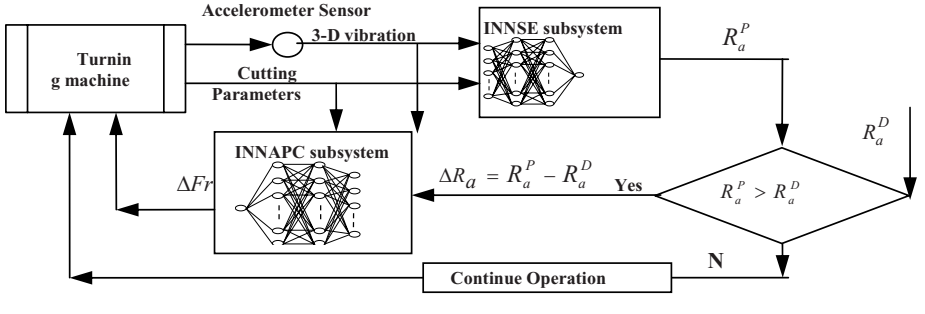


Fig. 1. The structure of the INNSAC system

4 Methodology

This section describes the experimental setup and design for the proposed in-process neural network-based surface roughness adaptive control (INNSRAC) system. The procedure employed to build up the system is also included.

The experimental hardware setup as shown in Figure 2 for this study includes a Storm CNC lathe (Fanuc Series 21i-T), an aluminum workpiece, a 3-axis accelerometer sensor (PCB Piezotronics, Inc.), a PCB battery power supply, and a DaqBook 100 data acquisition system (IOtech, Inc), an A/D board for converting data from analog to digital, and a PC for saving data. A Stylus profiler (Federal PocketSurf) is used for measuring roughness offline. The surface roughness is measured four times at different spots around the turned surface, and the average value is taken as the surface roughness. Tool wear is controlled by applying a brand new insert (CCGT432-AF) for collecting the experimental samples and randomizing the experimental sequence. Software consists of the following components:

1. An NC program that directs the CNC turning machine to cut the workpiece.
2. DaqView 11.8 (IOtech, Inc.), which records vibration signal.
3. JMP statistical program for correlation analysis
4. Microsoft Excel, which prepares the data for training neural network.
5. Neural network training software package.

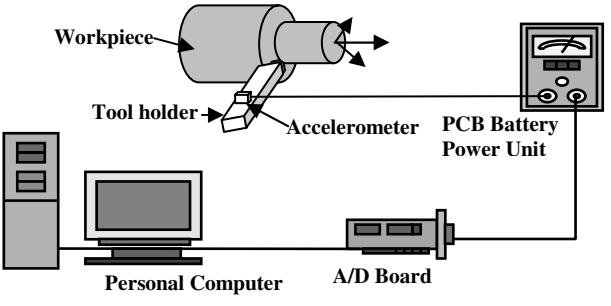


Fig. 2. Experimental setup for the INNSRAC system

A full factory design is used in the experimental design and spindle speed; feed rate and depth of cut (all cutting parameters) are the experimental factors. Spindle speed is set at three levels (2500, 3000 and 3500 revolutions per minute (RPM)), feed rate at six levels (0.002, 0.0034, 0.0048, 0.0062, 0.0076, 0.009, 0.0104, and 0.0118 inch per revolution (IPR)), and depth of cut at three levels (0.006, 0.012 and 0.018 inch). Each experimental combination is conducted twice. A total of 144 raw data sets were collected in the experiment. One turning insert will be used in the whole experiment process to better control the variation caused by different tools. In order to eliminate the systematic bias incurred by cutting sequence on surface roughness, the cutting sequence of the samples were randomized. The output of this subsystem is surface roughness. Besides the cutting parameters such as spindle speed, depth of cut, and feed rate, the inputs of this subsystem also include the significant vibration signals. The Pierson correlation analysis was carried out to determine the best representative

vibration factor. As seen in Table 1, all the vibration signals have significant correlation coefficients. An informal analysis of the trends in the data also indicated that vibration in all three directions is affected by depth of cut and spindle speed. To avoid multi-linearity, only the vibration with the highest correlation coefficient, V_z, is selected as an input. The artificial neural networks structure of the INNSE subsystem is presented in Figure 1. After the input variables of the INNSRP subsystem were identified, the neural network model for surface roughness prediction subsystem was trained following the training procedure as described in Section 2. The weights and offsets were initialized as 0.3 and 0.5 in the first training cycle, respectively. The acceptable accuracy of root mean square (RMS) was set at 0.05. The number of neurons in the hidden layers was determined through the trial-and-error method.

Table 1. Pearson correlation analysis of cutting parameters with the response

Variable	Pearson Correlation Coefficient*	
SP	-0.0213	
FR	0.9516	**
DC	-0.0010	
V_x	0.7637	**
V_y	0.7369	**
V_z	0.8488	**

*Response = R_a

**Significantly different from 0, with $\alpha = 0.01$.

After establishing the surface roughness model, an additional 24 samples were produced to test the performance of the proposed INNSRP subsystem. All the cutting parameter conditions were set within the experimental range, but differed from those in the experimental runs. The average prediction accuracy in this study was 93.5%, with a 0.3% standard deviation. This prediction accuracy rate suggests that the next subsystem should be developed for adaptive control system. The inputs of this subsystem include the spindle speed, feed rate, depth of cut, the Z direction vibrations and the recognized surface roughness difference, and the output is the adaptive degree of feed rate change. The structure of the INNAPC subsystem is demonstrated in Figure 1. The number of neurons in the hidden layers will be decided in the training process. To train this neural network model, two important observations were prepared. One was the detected surface roughness (ΔR_a) and the other was the feed rate change (ΔFr). To do so, the raw 72 cutting combinations were reorganized into nine data subgroups, each of which had the same spindle speed and depth of cut, but different feed rates. For example, subgroup #1 had 2500 RPM×0.006 inch, but had a feed rate from 0.002 through 0.0118. Within each subgroup, *each* sample was compared with every other sample. The larger measured surface roughness was assumed to be the predicted value (R_a^P) recognized by the INNSRP subsystem, while the smaller, actually-measured surface roughness was assumed to be the desired surface specification (R_a^D). In this way, theoretically, 28 ($=C_2^8$) observations can be formed for each subgroup, and a total of 252 ($=9C_2^6$) observations can be used for training the INNAPC subsystem. Once the training data are ready, the BP training procedures are followed to train the INNAPC subsystem until the RMS accuracy is reached.

5 Testing of the Completion of the Study

Integrating the INNSRP and INNAPC subsystems, the IPSRAC system was established for testing. When testing, first the workpiece was cut in half lengthwise and the INNSRP subsystem then provided a predicted surface roughness according to the surface roughness desired by customer. The INNAPC subsystem then generated the adaptive feed rate change ΔFr. By applying this ΔFr, the remaining half of the workpiece was cut. The surface roughness after applying ΔFr was then measured to see if it met the requirement. The overall success rate was then evaluated by the percentage of successful testing out of the total number of test runs.

References

1. Risbood, K.A., Dixit,U.S., Sahasrabudhe, A.D.: Prediction of Surface Roughness and Dimensional Deviation by Measuring Cutting Forces and Vibrations in Turning Process. J. of Mater. Process. Technol., 132(1-3) (2003) 203-214
2. Ho, S.-Y., Lee, K.-C., Chen, S-S, Ho, S-J: Accurate Modeling and Prediction of Surface Roughness by Computer Vision in Turning Operations Using an Adaptive Neuro-Fuzzy Inference System. Int. J. of Mach. Tools Manuf. 42(13) (2002) 1441-1446
3. Fang, X.D., Yao, Y.L.: In-process Evaluation of the Overall Machining Performance in Finish-turning via Single Data Source. Trans. of ASME, 119 (1997) 444-447
4. Lee, B.Y., Tarng, Y.S.: Surface Roughness Inspection by Computer Vision in Turning Operations. Int. J. Mach. Tools Manuf. 41(9) (2001) 1251-1263
5. Lee, S.S., Chen, J.C.: On-Line Surface Roughness Recognition System Using Artificial N System in Turning Operation. In. J. Ad. Manuf. Technol. 22 (2003) 498-509
6. Jang, Y.D., Choi, Y.G., Kim, H.G., Hsiao, A.: Study of the Correlation between Surface Roughness and Cutting Vibrations to Develop an On-Line Roughness Measuring Technique in Hard Turning. Int. J. Mach. Tools Manuf. 36(4) (1996) 453-464
7. Lin, S.C., Chang, M.F.: Study on the Effects of Vibrations on the Surface Finish Using a Surface Topography Simulation Model for Turning. Int. J. Mach. Tools Manuf. 38(7) (1998) 763-782
8. Huang, L., & Chen, J.C.: A Multiple Regression Model to Predict In-Process Surface Roughness in Turning Operation via Accelerometer. J. of Ind. Technol. 17(2) (2001)
9. Abouelatta, O.B., Madl, J.: Surface Roughness Prediction Based on Cutting Parameters and Tool Vibr. in Turning Operations. J. Mater. Process. Technol. 118 (2001) 269-277
10. Beauchamp, Y., Thomas, M. Youssef, Y.A., & Masounave, J.: Investigation of Cutting Parameter Effects on Surface Roughness in Lathe Boring Operation by Use of a Full Factorial Design. Comp. Ind. Eng. 31(3-4) (1996) 645-651
11. Kopac, J., Bahor, M., Sokovic, M.: Optimal Machining Parameters for Achieving the Desired Surface Roughness in Fine Turning of Cold Pre-Formed Steel Workpieces. Int. J. Mach. Tools Manuf., 42 (2002) 707-716
12. Freeman, J.A., Skapura, D.M.: Neural Networks: Algorithms, Applications, And Programming Techniques. Addison-Wesley, Reading, Massachusetts (1991)
13. McClelland, A., Rumelhart, P.: Explorations in Parallel Distributed Processing. MIT Press, Cambridge, Massachusetts (1988)
14. Lippmann, R.: An Introduction to Computing with Neural Nets. In: Mehra P. and Wah, B. W. (eds.): Artificial Neural Networks: Concepts and Theory. IEEE Computer Society Press, Los Alamitos, California (1999) 13-31

Modeling of Micro Spring Tension Force for Vertical Type Probe Card Fabrication

Chul Hong Min[1] and Tae Seon Kim[2,*]

[1] School of Computer Science & Engineering,
Catholic University of Korea, Korea
ventura@catholic.ac.kr
http://isl.catholic.ac.kr
[2] School of Information, Communications & Electronics Engineering,
Catholic University of Korea, Korea
tkim@catholic.ac.kr

Abstract. For design of micro spring based vertical type probe card, accurate micro spring tension force modeling is essential to guarantee the probe testing performance and reliability. In this paper, neural network based micro spring model was developed to find optimal spring height and shift value for appropriate tension force. Modeling results are applied to design and fabrication of vertical type probe card using 80μm and 100μm tungsten wires for micro spring type probing on silicon substrate. Compare to conventional statistical modeling scheme, neural network based model showed superior modeling accuracy with limited a priori information. Proposed high pad density probe card can be applied to high-density multi-die testing as well as advanced bumping type chip test.

1 Introduction

As the increase of device complexity and performance, number of signal input/output pads is drastically increased and it leads the difficulties on wafer level chip test. Therefore, for wafer level chip testing, successful development of probe card for high-density input/output pads has been considered as one of the crucial steps. Currently developed probe cards are mainly divided into three types, horizontal type probe card, MEMS (micro electro mechanical systems) type probe card, and vertical type probe card. The horizontal type probe cards are the most common in probe card design for last few years [1], [2]. However, with the limitations of probe stacking capability and tension control uniformity, the horizontal type probe cards cannot be applied to high density ICs (integrated circuits) such as high performance microprocessors and SoC (system on chip) type ICs. To overcome this limitation, semiconductor fabrication technique based MEMS type probe card design is considered as an alternative way for future probe card technology standard [3], [4]. Although it allows

* Corresponding author.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 976 – 981, 2006.

higher probing density, most of currently developed MEMS type probe cards still have several technical difficulties and limitations. For example, cantilever type MEMS probe cards can support relatively low level of tension force and it leads higher contact resistance. Another problem is bonding of silicon probe and PCB (printed circuit board) for electrical interconnection since the via formation on silicon probe card to PCB requires very high-density interconnections on limited space. Beside these technical difficulties, manufacturing cost of MEMS type probe card is too high to widely used [5]. In contrast, the vertical type probe card requires comparably lower fabrication cost than MEMS type probe card. However, accurate tension force control capability for reliable probing quality is big obstacle to fabricate. In this paper, new micro spring based vertical type probe card design is proposed. To fabricate micro spring based vertical probe, tungsten wires are vertically connected on ceramic board. To control the tension force of micro spring, neural network based micro spring model was designed to find optimal spring height and shift value for appropriate tension force. Modeling results are applied to design and fabrication of vertical type probe card using 80μm and 100μm tungsten wires for micro spring on silicon substrate.

2 Design of Vertical Type Probe Card Module

The proposed vertical type probe card is mainly divided into four parts, ceramic board, stiffener, bridge, and micro-spring using tungsten wire as shown in Figure 1.

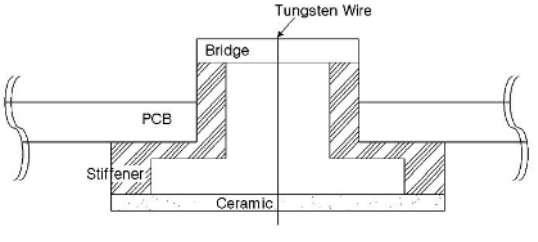


Fig. 1. A cross-section of proposed micro spring based vertical type probe card

2.1 Design of Ceramic Board

As shown in Figure 1, tungsten wire passed through the ceramic board for electrical probing of test sample. For this, the ceramic board must have holes that aligned to pad location of tested sample. The role of the ceramic board is guide each probe located on exact right upside of pad location. However, to allow tension force on tungsten wire, hole diameter must slightly bigger than diameter of tungsten wire and it made misalign between probe and pad location. For example, insertion of tungsten wire with 80μm diameter and 1200μm diameter height on ceramic hole of 90μm diameter made 10.5μm of misalign on pad location as shown in Figure 2. To compensate this error, offset values are added to origin of ceramic hole coordinates.

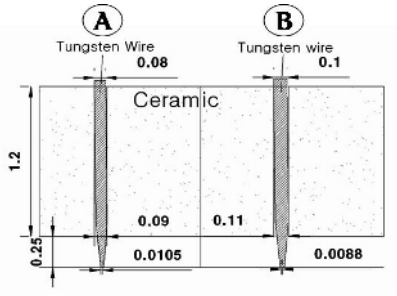


Fig. 2. Misaligned error caused by diameter difference between ceramic hole and tungsten wire (unit: mm)

2.2 Design of Micro Spring Bridge

The micro spring bridge on the top of probe card takes a role to sustain tungsten wires inserted in ceramic holes. As shown in Figure 3, the shapes of tungsten wire depend on the thickness of bridge and bridge hole size. For example, if thick bridge has small bridge holes, then micro springs have shapes of ⓐ in Figure 3. To minimize contact resistance in probing stage, probe needle requires tension force to allow acceptable amount of overdrive (OD). The amount of tension force varies on shapes of micro spring and they are selected by types of testing devices.

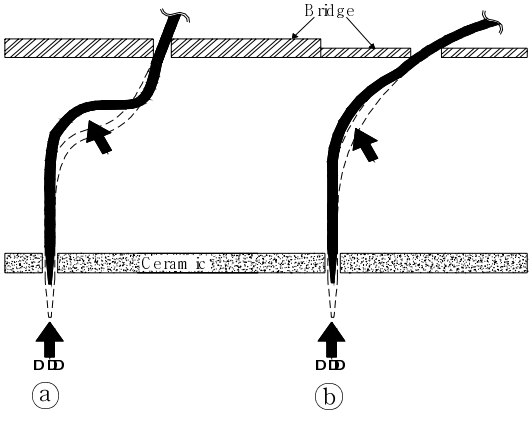


Fig. 3. Shapes of micro spring after overdrive is applied

2.3 Design of Micro Spring for Tension Control

Proposed probe card used vertically aligned tungsten wire as a probe needle. If OD is applied to tungsten wire, it acts like micro spring that has tension force and restitution force. The tension force is controlled by height of micro spring (H), shift (S), and the diameter of micro spring (D) as shown in Figure 4. For accurate probing, tension

force control is crucial and control of tension force uniformity is very critical since recently fabricated chips with higher probe density require tight testing margin. For design of probe needle stage, every tension force values should be predictable for all the range of H, S, and D. For this, accurate tension for model is required and it can not only reduce probe card design time and cost but also guarantee the probing quality. Traditionally, statistical modeling method such as regression modeling scheme is widely used for tension modeling. However, modeling accuracy is not high enough and also statistical methods require lots of designed experimental data which are not cost effective. In this paper, neural networks are used to find relation between tension force and probing conditions including D, H, S, and OD.

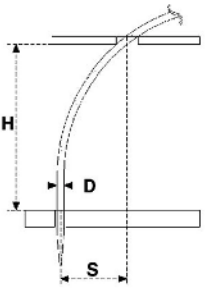


Fig. 4. Three variables of micro spring for tension control

3 Modeling Results

For modeling of micro spring tension force, three layered feed-forward error back-propagation neural network (BPNN) is applied and network weights were updated after each training vector (by "training by vector" method). Three micro spring parameters (H, D, and S) and OD values are used as network inputs and the network outputs are scaled tension force values. Table 1 shows the ranges of modeling input parameters. Based on the systemic statistical experiment, network learning parameters (learning rate, momentum) and structure (number of layers, number of hidden neurons) were optimized. Also, to optimize transfer function of BPNN, three kinds of log-sigmoid, tan-sigmoid, and linear-sigmoid functions are tested and log-sigmoidal transfer function showed best modeling results.

Table 1. Ranges of modeling input parameters

Parameter	Range	Units
Height (H)	8~12	mm
Shift (S)	5~7	mm
Diameter (D)	80~100	μm
Overdrive (OD)	10~250	μm

Pre-tested 300 tension force data were used for network training, and untrained extra data were used as test sample data. To evaluate prediction capability, performance of conventional multiple regression modeling scheme was compared with prediction results of proposed neural process model. The modeling performance was measured by the root mean squared error (RMSE), σ is calculated as shown in Equation (1).

$$\sigma = \sqrt{\frac{1}{(n-1)}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{1}$$

In this equation, n, y_i, and $\hat{y}_i$ represent the size of the test set, the measured tension force, and the model prediction value, respectively.

As shown in Table 2, the proposed neural based model showed only 1.36g of prediction RMSE and it shows 4.89g of improvement as compared to statistical prediction model using multiple regressions. Regardless of the RMSE improvement, the remarkable point is that proposed neural network model fits to entire tension force ranges without any bias as shown in Figure 5. In contrast, multiple regressions showed good modeling results only for limited ranges because of non-linearity nature of micro spring tension force.

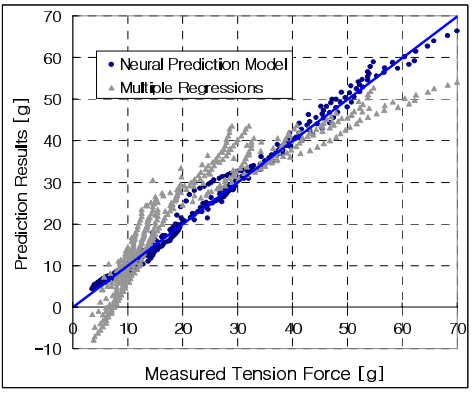


Fig. 5. Comparison of neural networks vs. multiple regressions model prediction results

Table 2. Prediction RMSE comparison

	Proposed model	Multiple Regressions	% Improvement
RMSE	1.36g	6.25g	78.2%

4 Conclusions

In this paper, micro spring based vertical type probe card is designed and fabricated for high pad density testing. For tension force control, neural network based micro

spring model was designed to find optimal spring height and shift value for appropriate tension force. Modeling results are applied to design and fabrication of vertical type probe card using 80μm and 100μm tungsten wires for micro spring on silicon substrate. Compare to conventional statistical modeling scheme, neural network based model showed superior modeling accuracy and fits to entire tension force ranges without any bias with limited a priori information. With completion of this work, proposed model based vertical type probe card design method can be applied to multi-die testing scheme as well as bumping type chip test.

Acknowledgement

This work was supported by the Catholic University of Korea Research Fund, 2006.

References

1. http://www.jem-net.co.jp/en/products/pro_han_kpc.html
2. Iwai, H., Nakayama, A., Itoga, N., Omata, K.: Cantilever type probe card for at-speed memory test on wafer. VLSI Test Symposium IEEE, May (2005)85–89
3. Kim, Y.M., Yoon, H.C., Lee, J.H.: Silicon Micro-probe Card Using Porous Silicon Micromachining Technology. ETRI Journal 8(2005)433–438
4. Itoh, T., Kawamura, S., Kataoka, K., Suga, T.: Contact Properties of Micromachined Ni Probes. Proceedings of the Forty-Ninth IEEE Holm Conf. (2003) 223–227
5. Tracy, N.L., Rothenberger, R., Copper, C., Corman, N., Biddle, G., Matthews, A., McCarthy, S.: Array sockets and connectors using MicroSpringTM technology. Electronics Manufacturing Technology Symposium, Twenty-Sixth IEEE/CPMT International, (2000) 129–140

Identification of Crack Location and Depth in Rotating Machinery Based on Artificial Neural Network

Tao Yu, Qing-Kai Han, Zhao-Ye Qin, and Bang-Chun Wen

School of Mechanical Engineering & Automation, Northeastern University, 110004, Shenyang, P.R. China
yt_126@126.com, qingkai_han@sohu.com, neuapostle@163.com, Bcwen1930@vip.sina.com

Abstract. With the characteristics of ANN's strong capability on nonlinear approximation, a new method by combining an artificial neural network with back-propagation learning algorithm and modal analysis via finite element model of cracked rotor system is proposed for fast identification of crack fault with high accuracy in rotating machinery. First, based on fracture mechanics and the energy principle of Paris, the training data are generated by a set of FE-model-based equations in different crack cases. Then the validation of the method is verified by several selected crack cases. The results show that the trained ANN models have good performance to identify the crack location and depth with higher accuracy and efficiency, further, can be used in fast identification of crack fault in rotating machinery.

1 Introduction

Rotating machinery, such as steam turbo, compressor, aeroengine and blower etc., are widely used in many industrial fields. The fault identification and diagnosis of rotating machinery, i.e. rotor system, have become a vigorous area of work during past decade[1].

Among the important rotor faults, the crack fault, which can lead to catastrophic failure and cause injuries and severe damage to machinery if undetected in its early stages, is most difficult to detect efficiently with traditional methods like waveform analysis, orbital analysis etc. Although the dynamic behaviors of rotor with transverse crack have been studied relatively enough[2], [3], [4], [5] and many methods have been introduced to detect crack(s) for non-rotating structures, they are not very suitable for fast identification of cracks in rotating machine.

In the paper, based on the truth of the change of the mode shapes of the cracked structure - the most obvious effect of a structural crack, a new method, as shown in Fig.1, by combining accurate FE (Finite Element) model of rotor with transverse crack(s) and artificial neural network (ANN) is proposed to identify the location and depth of a crack in rotating machinery. Initially the accurate FE model of the object, a rotor system with a localized transverse on-edge non-propagating open crack, is built to produce the specific mode shape. Then a set of different mode shapes of a rotor system with localized crack in several different position and depth, which will be

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 982 – 990, 2006.

treated as the input of the designed ANN model, can be obtained by repeating the above step. To insure the accuracy and quickness of the training process for the designed ANN model, a nonlinear model of a neuron is employed and a back-propagation learning algorism with Levenberg-Marquardt method is used. At last, with several selected crack cases, the errors between the results obtained by using the trained ANN model and FEM ones are compared and illustrated at last.

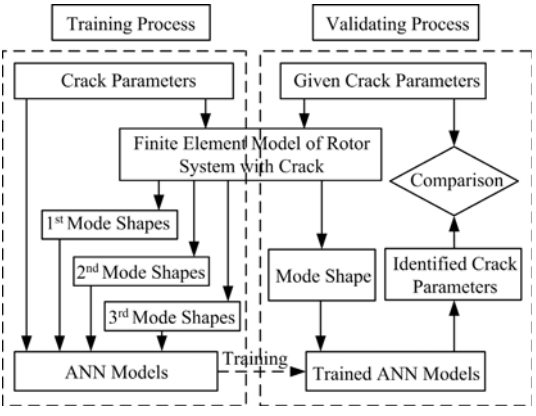


Fig. 1. Flow chart of identification scheme

2 Modes Calculation Based on FE Model of the Rotor System with a Transverse Crack

A type of rotating machinery classified to the category with high speed and light load, can be considered as a jeffcott rotor, as shown in Fig.2.

Fig. 2. A single-span-single-disc (jeffcott) rotor system

It is assumed that the crack changes only the stiffness of the structure whereas the mass and damping coefficients remains unchanged. Cracks occurring in structures are responsible for local stiffness variations, which in consequence affect the mode shapes of the system. The finite element model of rotor system with a transverse crack and a single centrally situated disc, and also the geometry of the cracked section of the shaft are shown in Fig 3.

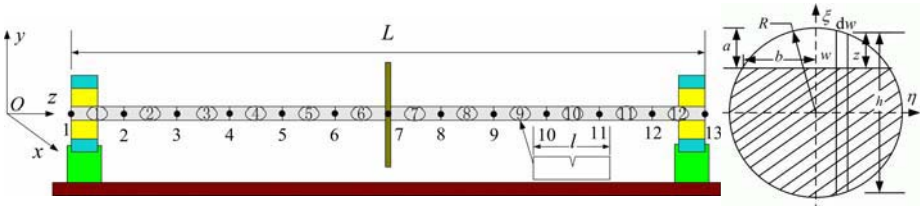


Fig. 3. The FE model of the rotor system and the geometry of the cracked section of the shaft

2.1 Calculation of Total Stiffness Matrix of the Crack Element

According to the fracture mechanics and the energy principle of Paris, the additional strain energy due to a crack is given by the following equations:

$$U = \int_A J(A)\mathrm{d}A \,. \tag{1}$$

where $J(A)$ is strain energy density function with only bending deformation taken into consideration and is expressed as:

$$J(A) = \frac{1}{E'} K_{\mathrm{I}}^2 \,. \tag{2}$$

where $E' = E/(1-\nu)$, ν is the Poisson ratio and K_{I} is the stress intensity factors corresponding to the bending moment M in plane crossing the axis of the beam

The expression of K_{I} is

$$K_{\mathrm{I}} = \sigma\sqrt{\pi z} F_2(z/h) \,. \tag{3}$$

where $\sigma = 4M\sqrt{R^2-\eta^2}/\pi R^4$, $h = 2\sqrt{R^2-\eta^2}$ is the local height of the strip, and

$$F_2(\frac{z}{h}) = \sqrt{\frac{2h}{\pi z}\tan\frac{\pi z}{2h}}\,\frac{0.923+0.199[1-\sin(\pi z/2h)]^4}{\cos(\pi z/2h)} \,. \tag{4}$$

Substituting the Eq. (2)~(4) into Eq. (1), the local flexibility due to the crack in the ξ-axis direction can be written as

$$c_\xi = \frac{\partial^2 U}{\partial M^2} = \frac{1-\nu^2}{E}\int_{-b}^{b}\mathrm{d}\eta\int_0^z \frac{32}{\pi^2 R}(R^2-\eta^2)\pi z F_2^2(\frac{z}{h})\mathrm{d}z \,. \tag{5}$$

where b is the crack width ($b = \sqrt{R^2-(R-a)^2}$)[6], z is the local depth of crack strip.

The similar expression of local flexibility in the η-axis direction can be written as

$$c_\eta = \frac{\partial^2 U}{\partial M^2} = \frac{1-\nu^2}{E}\int_{-b}^{b}\mathrm{d}\eta\int_0^z \frac{32}{\pi^2 R}\eta^2\pi z F_1^2(\frac{z}{h})\mathrm{d}z \,. \tag{6}$$

where $F_1(\frac{z}{h}) = \sqrt{\frac{2h}{\pi z}\tan\frac{\pi z}{2h}}\,\frac{0.752+2.02(z/h)+0.37[1-\sin(\pi z/2h)]^3}{\cos(\pi z/2h)}$

Now we can get the local flexibility matrix $\boldsymbol{C}_1$ of the element with a crack

$$\boldsymbol{C}_1 = \begin{bmatrix} c_\xi & 0 \\ 0 & c_\eta \end{bmatrix}. \tag{7}$$

Here, the coupled flexibility is neglected since it is much less than the element in the main diagonal of the $\boldsymbol{C}_1$. Let $\boldsymbol{C}_0$ is the flexibility matrix of uncracked shaft element. The total flexibility matrix $\boldsymbol{C}$ is expressed as

$$\boldsymbol{C} = \boldsymbol{C}_0 + \boldsymbol{C}_1. \tag{8}$$

Thus, the total stiffness matrix of the crack element is written as

$$\boldsymbol{K}_c = \boldsymbol{T}\boldsymbol{C}\boldsymbol{T}^{\mathrm{T}}. \tag{9}$$

where $\boldsymbol{T}$ is the transformation matrix and given by

$$\boldsymbol{T} = \begin{bmatrix} -1 & -l & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}. \tag{10}$$

2.2 FE Modeling of the Rotor System

Assembling all the element mass, damping and stiffness matrices of the rotor system in stationary coordinate system, the equation of motion in stationary coordinate is

$$\boldsymbol{M}\ddot{z} + \boldsymbol{C}\dot{z} + \boldsymbol{K}z = \boldsymbol{F}\ . \tag{11}$$

where $\boldsymbol{M}$, $\boldsymbol{C}$, $\boldsymbol{K}$ and $\boldsymbol{F}$ are total mass, damping, stiffness and external exciting force matrices of rotor system respectively. z is the displacement of the element node.

2.3 Mode Shape of the Rotor System

For the rotor system considered in the paper, each beam element has two nodes and each node has two degrees of freedom representing transverse and deflecting displacements of the corresponding cross-section. Here, only the mode shape in the ξ-axis direction is discussed by assuming the rotor system is rigid supported at the bearing position.

The mode shape can be obtained by solving the homogeneous part of Eq.11 without considering the effect of the damping.

$$\boldsymbol{M}\ddot{x} + \boldsymbol{K}x = 0. \tag{12}$$

Substituting $x_i = A^{(i)}\sin(\omega_i t + \varphi_i)$ into Eq.12, we get

$$(-\omega_i^2\boldsymbol{M} + \boldsymbol{K})A^{(i)} = 0. \tag{13}$$

where ω_i and $A^{(i)}$ is the *i*-th nature frequency and eigenvector (mode shape).

3 Artificial Neural Network Design

The architecture of a typical ANN model with back-propagation algorithm (BPA), as shown schematically in Fig.4, is found to be efficient and perform well in many practical applications [7]. During the training process, the weights moved in the direction of the negative gradient, and together with biases of the network are iteratively adjusted to minimize the network performance function - mean square error (MSE), which is the average squared error between the network outputs and the target outputs. The Levenberg-Marquardt back-propagation algorism and Nguyen-Widrow function [8] are used in the ANN training and initialization of weights and biases of the hidden layer for faster convergence and accuracy.

The rotor system discussed in the present study, consisting of a 10 mm diameter shaft carrying a centrally situated steel disc with 50 mm diameter and 5 mm width, is divided into 12 elements as shown in Fig.3. Using the method described before, the modes of different cases are calculated by introducing a transverse crack deliberately in the middle of certain element, as shown in Fig.5. For the central symmetry of the rotor system, only the modes with crack in first 6 elements are illustrated.

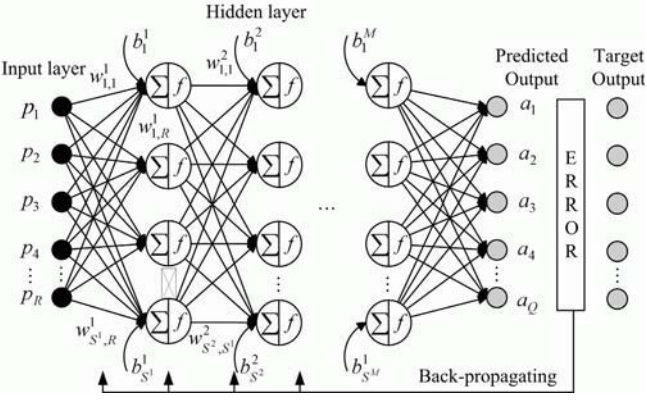
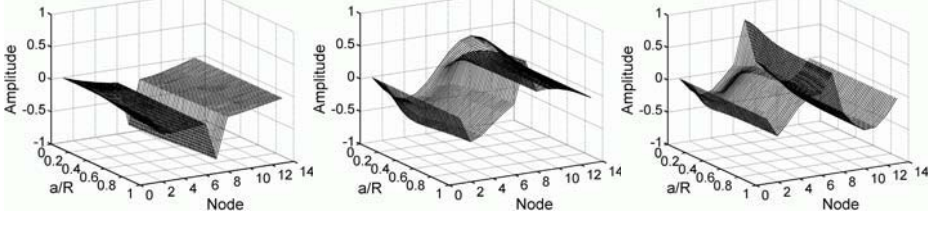
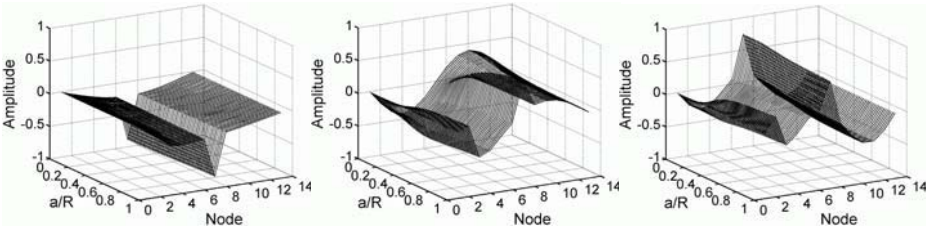


Fig. 4. The architecture of a BP ANN model

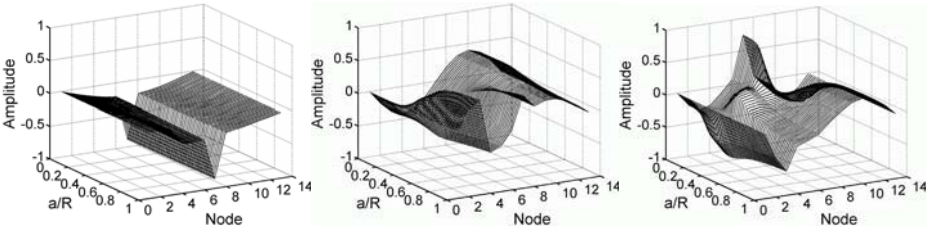


(a) modes of rotor system with crack in No.1 element

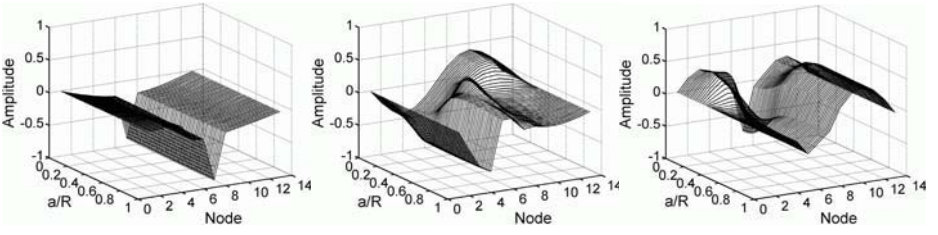
Fig. 5. The 1st, 2nd and 3rd modes of rotor system when crack in different element



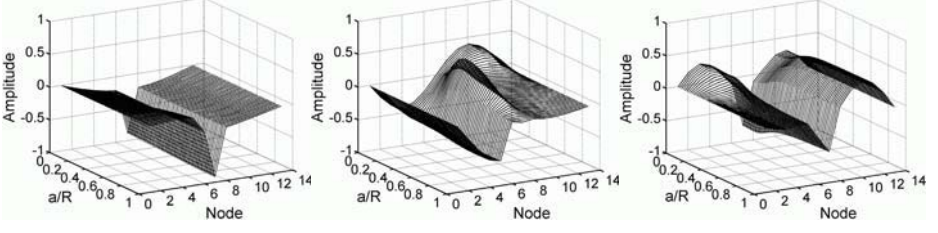
(b) modes of rotor system with crack in No.2 element



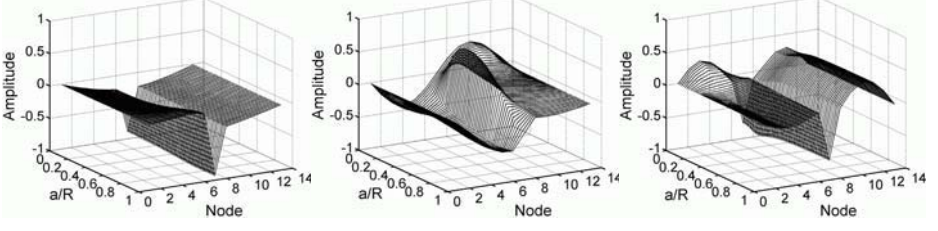
(c) modes of rotor system with crack in No.3 element



(d) modes of rotor system with crack in No.4 elementr



(e) modes of rotor system with crack in No.5 element



(f) modes of rotor system with crack in No.6 element

Fig. 5. (*continued*)

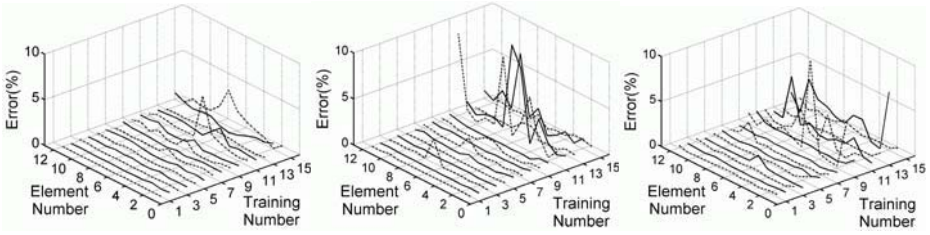
In each case, the modes corresponding to a/R ratios from 0.01 to 1 at a interval of 0.01 are the vectors consisting of 13 elements, which represent the amplitude of each node in the x-direction. The network architecture is built in terms of the numerals as N_{I}, N_{H1}, ⋯, $N_{\mathrm{H}n}$ and N_{O}, which denotes the number of neurons in the input layer, $\mathrm{H}_n^{\mathrm{th}}$ hidden layer and output layer respectively. The input and target output for training are expressed respectively as $\boldsymbol{P}=[\boldsymbol{P}_1\ \boldsymbol{P}_2\ \cdots\ \boldsymbol{P}_n]$ and $\boldsymbol{T}=[\boldsymbol{T}_1\ \boldsymbol{T}_2\ \cdots\ \boldsymbol{T}_n]$, where $\boldsymbol{P}_i=[\boldsymbol{p}_1\ \boldsymbol{p}_2\ \cdots\ \boldsymbol{p}_m]_{13\times m}$ (i=1, 2, ⋯, n), denotes the modes matrix corresponding to crack at a certain position with m kinds crack depth; $\boldsymbol{T}_i=[\boldsymbol{t}_1\ \boldsymbol{t}_2\ \cdots\ \boldsymbol{t}_m]_{12\times m}$ (i=1, 2, ⋯, n), denotes the crack depth vectors corresponding with different position. That is to say, the target output vector is filled with values equal to 0 or a number between 0 and 1, and a value not equal to 0 is symbolic of the localization and depth of a crack. For example, the column vector $[0\ \ 0\ \ 0.5\ \ \cdots\ \ 0]^{\mathrm{T}}_{12\times 1}$ indicates the presence of a crack with a a/R ratio equal to 0.5 in the middle of No. 3 element.

Since the dimensions of input and output vectors have been decided, the ANN needs to have a 13-(N_{H1}…N_{Hn})-12 architecture. But how many hidden layers and neurons in each hidden layers should be adopted has not a routine way to decide. Here, a single hidden layer with 20 neurons is designed for the following training process and proved to perform well with proper training data.

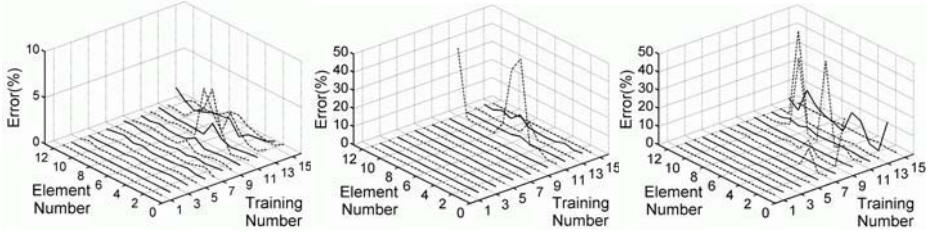
4 Training and Validation of the ANN

The training process can be started with the prepared training data and ANN model. To evaluate the effects of training data on the success of crack identification, different training scheme is designed. That is to say, with total 12 shaft elements and 100 kinds of depth in each element corresponding to any type of 3 modes, the training data are arranged at intervals of 0, 1, 2, … and 14 respectively. Thus, 15 ANN models corresponding to each mode are trained. Obviously, ANN models trained with an interval of 0 have the best performance of almost zero error for the full use of the training data. To compare the performances of the other trained ANN models, several crack cases excluded in the training data of different schemes are adopted as test cases, and the errors between predicted crack parameters (location and depth) and actual ones under different trained ANN models and test cases are illustrated in Fig.6.

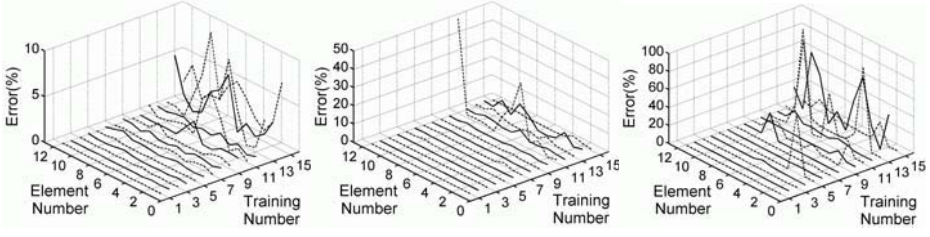
It can be seen from Fig.6 that under condition of a given testing case, i.e. with a constant crack depth ratio (a/R), the increase of the system modes leads to performances of trained ANN models dropping down, and the differences between them are more and more remarkable with the decrease of the testing crack depth (a/R); under all training schemes, the performances of ANN models trained by 1^{st} modes training data vary little when the crack depth ranges from 0.4~1. For the little effect of a tiny crack on the stiffness of the rotor system, the performances of trained ANN models may relatively lower when the training scheme with a large interval of training data is adopted. And for the effects of the discs on the modes of the rotor system, the errors are magnified with the decrease of the quantity of training data.



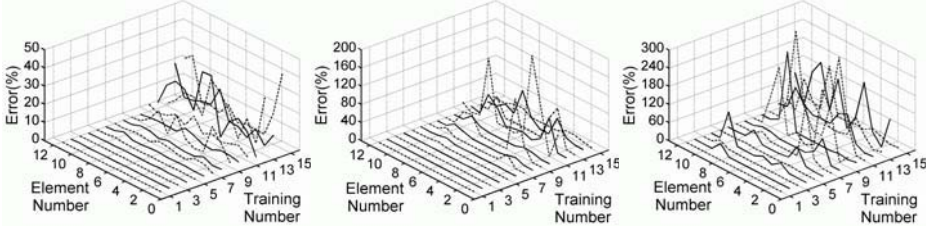
(a) testing case with a depth ratio (a/R) 0.80



(b) testing case with a depth ratio (a/R) 0.62



(c) testing case with a depth ratio (a/R) 0.48



(d) testing case with a depth ratio (a/R) 0.32

Fig. 6. Performances of different trained ANN models with several testing cases (*figures from left to right in each case correspond to the performances of ANN models trained with 1st modes, 2nd modes and 3rd modes of the cracked rotor system respectively*)

With the method and measured 1st mode shapes of cracked rotor system as shown in Fig.2, the crack location and depth can be identified quantificationally to a good degree of accuracy with the trained ANN models.

5 Conclusion

By combining FE model of the rotor system and ANN models with architecture of 3-layer and 20 neurons in hidden layer, a novel method to non-destructively identify a crack location and depth in rotating machinery is presented in the paper. With the ANN models trained by modes, especially the 1st modes of cracked rotor system, the given testing crack cases are identified to a good degree of accuracy. It is proved that the effectiveness of identification process depends to a good extent on the number of training data when the crack is tiny or crack depth is shallow. But it has also been found that there are few effects of the quantity of training data on the performances of trained ANN models when the training modes are reasonably selected.

However, for the relative simplicity of ANN models designed here and the complexity of 2nd and 3rd modes, the performances of ANN models trained by them are limited. Research to improve the crack identification method is continuing as the follows: First, more accurate FE model of the rotor system and efficient modes measuring method should be developed according to the practical rotor machinery. Second, the attempt to apply the method on detecting multiple cracks should be conducted, though the single crack case has been studied in the paper.

Acknowledgements

The paper is supported by the Science & Technology Foundation of Liaoning Province (No. 20044003) and the Key Project of National Natural Science Foundation of China (No. 50535010). The authors would like to thank for their financial support.

References

1. Wauer, J.: On the Dynamics of the Cracked Rotors: A Literature Survey. Applied Mechanics Review 43 (1) (1990) 13–17
2. Dimarogonas, A.D.: Vibration of Cracked Structures: A State of the Art Review. Engineering Fracture Mechanics 55 (5) (1996) 831–857
3. Bachschmid, N., Pennacchi, P., Tanzi, E., Audebert, S.: Identification of Transverse Cracks in Rotors Systems. Proc. ISROMAC-8 Conf. Honolulu Hawaii (2000) 1065–1072
4. Wen, B.C., Wu, X.H., Ding, Q., Han, Q.K.: Dynamics and Experiments of Fault Rotors (in Chinese). 1st edn. Science Press, Beijing (2004)
5. Wen, B.C., Gu, J.L., Xia, S.B., Wang, Z.: Advanced Rotor Dynamics (in Chinese). 1st edn. China Machine Press, Beijing (2000)
6. Zou, J., Chen, J., Niu, J.C., Geng, Z.M.: Discussion on the Local Flexibility Due To the Crack in A Cracked Rotor System. Journal of Sound and Vibration 262 (2003) 365–369
7. Vyas, N.S., Satishkumar, D.: Artificial Neural Network Design for Fault Identification in a Rotor-bearing System. Mechanism and Machine Theory 36 (2001) 157–175
8. Hagan, M.T., Demuth, H.B., Beale, M.H.: Neural Network Design. PWS Publishing Company, Boston (1996)

Natural Color Recognition Using Fuzzification and a Neural Network for Industrial Applications

Yountae Kim[1], Hyeon Bae[1], Sungshin Kim[1], Kwang-Baek Kim[2], and Hoon Kang[3]

[1] School of Electrical and Computer Engineering, Pusan National University, Changjeon-dong, Keumjeong-ku, Busan 609-735, Korea
{dream0561, baehyeon, sskim}@pusan.ac.kr
[2] Department of Computer Engineering, Silla University
gbkim@silla.ac.kr
[3] School of Electrical and Electronics Engineering, Chung-Ang University
hkang@cau.ac.kr

Abstract. The Conventional methods of color separation in computer-based machine vision offer only weak performance because of environmental factors such as light source, camera sensitivity, and others. In this paper, we propose an improved color separation method using fuzzy membership for feature implementation and a neural network for feature classification. In addition, we choose HLS color coordination. The HLS includes hue, light, and saturation. There are the most human-like color recognition elements. A proposed color recognition algorithm is applied to a line order detection system of harness. The detection system was designed and implemented as a testbed to evaluate the physical performance. The proposed color separation algorithm is tested with different kinds of harness line.

1 Introduction

The status of the target is recognized through a suitable digital image processing algorithm or numerical solution. Many aspects of several functions in artificial vision are implemented and used for industrial equipment. But, the aspect of color recognition still has many problems to be solved. One of the underlying problems is the subjectivity of measurements that come from several values according to the strength, direction, and color temperature of light and the quality of the material of an object. We normally try to recognize and represent an object as one color, but an object consists of many color elements, which together compose natural color. Many intelligent methods have been developed to overcome these problems.

In this paper, we propose a simple algorithm, with a low computational cost, for natural color recognition. First, we introduce the HLS color coordination system. The HLS color coordinate system is similar to human's natural color recognition system. Second, we use a statistical idea from a fuzzy membership for handle of natural colors. fuzzy memberships based on RGB and HLS information are represented. Third, the classification result of each color is computed and compared by neural network for color recognition. Finally, the proposed algorithm is applied to the harness line color detection system, and the performance is discussed.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 991 – 996, 2006.

2 HLS Color Coordination

2.1 HLS

HLS color coordination (hue, light and saturation) [1] is based on human color recognition of an object. Hue represents the color frequency. Saturation mixing with white and density expresses the luminance or realized brightness of an object. If a target color is chromatic, we can obtain that color's hue value. Using the hue value, we can determine the origin of color. The fact that more correct color recognition than RGB is possible has been proved by H. Palus and D. Bereska [2].

2.2 Fuzzification of Color Distribution

In this research, our color information was acquired from a harness image of nine differently colored lines each 100 color elements (pixels). The following among line of other colors, red line color information will *R*, *G*, *B*, *L* and *S* distributions of each line. Fig. 1 (a) is a non-convex form. It does not treat to fuzzy set. Then we connect out-line of these distribution plots apply concept to compensate for information that was lost by digitizing (It likes an anti-aliasing function) and then changed in form of fuzzy membership as shown in Fig. 1 (b). To make these memberships, we made the distribution of red, green, blue, light, saturation elements of an acquired natural color area. The hue element is remained for another purpose. We don't make the hue membership information. The prepared data was used to generate membership information through the following calculation:

$$\mu_M(i) = \frac{n}{N},\ i = 0,1,\cdots,255 \quad for\ R,G,B,L,S \tag{1}$$

where μ_M are membership values, n is the amount of membership data and N is the amount of whole data.

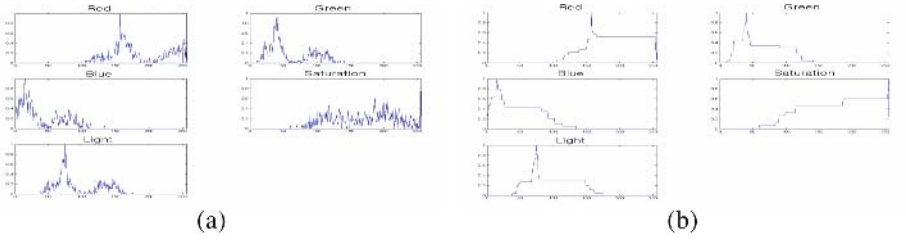


Fig. 1. Distribution (a) and fuzzification (b) graphs of R, G, B, L, S (red line) - In door, LED light source

2.3 Fuzzy Similarity Measure

Because we use a statistical idea from fuzzy membership, we need comparative and evaluative methods for fuzzy memberships. We introduce fuzzy measures. The measure of fuzziness is an interesting topic in the fields of pattern recognition and decision

theory. The measure of a crisp set can be preformed by classical mathematical study, whereas the concepts of fuzzy measures and fuzzy integrals have been proposed by Sugeno[3]. Recently, Liu suggested three axiomatic definitions of fuzzy entropy, distance measure and similarity measure [4]. Among these definitions, we used fuzzy similarity's concept of color recognition.

Definition 2.1. [4] A real function $s:F^2 \rightarrow R^+$ is called a similarity measure on $F(X)$ if s satisfies the following properties:

(S1) $s(A,B)=s(B,A),\ \forall A,B\in F(X)$

(S2) $s(A,A^c)=0\ \ \forall A\in F(X)$

(S3) $s(D,D)=\max_{A,B\in F} s(A,B),\ \forall D\in P(X),\ \forall A,B\in F(X)$

(S4) $\forall A,B,C\in F(X), if\ A\subset B\subset C, then\ s(A,B)\le s(A,C)\ and\ s(B,C)<s(A,C)$.

where, s means similarity measure, A, B, C and D are fuzzy memberships.

We used a similarity measure that satisfies Definition 2.1.

$$sim(M_1,M_2)=\frac{\sum_{i=1}^{n}\min(\mu_{M_1}(i),\mu_{M_2}(i))}{\sum_{i=1}^{n}\max(\mu_{M_1}(i),\mu_{M_2}(i))} \tag{2}$$

3 The Color Recognition Algorithm

In Fig. 2, our proposed algorithm has three stages. There are a fuzzification stage, a template matching stage, and a classification stage. The fuzzification stage was introduced in sub-section 2.2. This section introduces remained two stages.

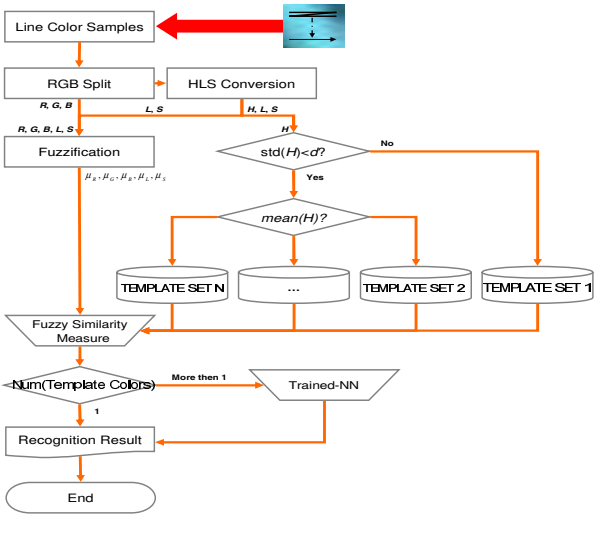


Fig. 2. Algorithm flow-chart for color recognition

3.1 Template Matching Using Similarity Measure

The proposed algorithm uses a selective template matching method by the hue standard deviation and mean value of an object color area. First step, we construct templates. Templates are constructed from memberships of each color's red, green, blue, light, saturation elements. Each 40 sets from every color area are used to construct memberships through the following expression:

$$Template = \sup(\mu_i), \quad i:1 \sim 40 \tag{3}$$

where μ_i are i-th membership values.

In example, Fig. 3 shows the process of light templates from 40 light memberships of the red color area. As the result, each color has five templates (red, green, blue, light, and saturation). Then, we separate templates into 4 groups as shown in table 1. In this research, we have 8 color areas for color recognition from harness lines. We define purity-colors those color areas have low standard deviation values under 30 of hue. Because of sensitive characteristics of light in CCD sensors, most of dark colors have large standard deviation value.

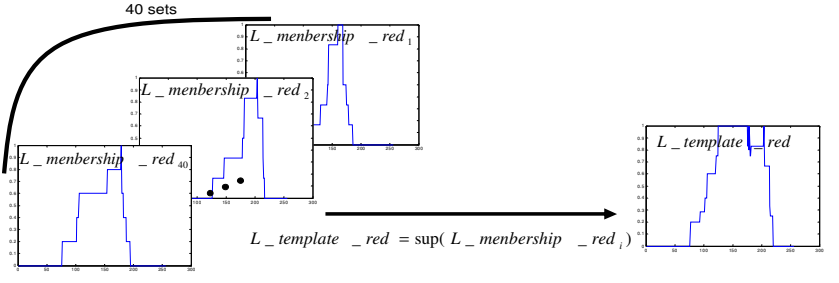


Fig. 3. Construction of a template

Table 1. Template separation by hue

Hue value		TEMPLATE NUMBER	Colors
std(H)≥30 Impurity		1	BLACK, GRAY, PURPLE
std(H)<30 Purity	-30(330)~ 30	2	BROWN, RED, ORANGE
	30~90	3	YELLOW
	90~210	4	GREEN
	210~270	5	BLUE

If color information came from an unknown color area then the first, we find standard deviation of hue. For example, if it less then 30 and averaged hue value is 62 then this unknown data compare with only yellow templates and the algorithm reports similarity measure. If, the result is admissible, the algorithm reports color recognition result and be completed. But, if it less then 30 and averaged hue value is 0 then this unknown data compare with brown, red, and orange color templates and produced similarity values goes to the trained neural networks to find a color of input area.

3.2 Neural Networks for Color Recognition

Neural Networks are used for color judgment using reported results of similarity measures from several color templates. It is judged by neural networks. For example, if an unknown input is an impurity color, then it compared with TEMPLATE 1 in Table 1. It has 15 similarity results (5 similarity results for each color). The neural network judges color from neural network inputs that consist of these results. The input is showed in Table 2.

Table 2. Inputs for neural networks

	TEMPLATEs	1	2
Inputs	1	Average of input hues	Average of input hues
	2	STD of input hues	STD of input hues
	3~7	*Sim* of L, S, R ,G ,B for Black	*Sim* of L, S, R, G, B for Brown
	8~12	*Sim* of L, S, R ,G ,B for Gray	*Sim* of L, S, R ,G ,B for Red
	13~17	*Sim* of L, S, R ,G ,B for Purple	*Sim* of L, S, R ,G ,B for Orange

We select a multi-layer feed forward back-propagation neural network. Network parameters are shows in Table 3. The network result is consists of two bit binary codes. Before using the neural network, we have training work. We make 30 training input sets for neural network training and training parameters listed in Table 4.

Table 3. Parameters for neural networks

Type	Feed-forward back-propagation			
Training function	Levenberg-Marquardt backpropagation			
Layers	4			
Neurons	Layers			
	1st	2nd	3rd	4th
	17	24	10	2

Table 4. Training parameters

Epochs	100
Learning rate	0.001
Num. of training inputs	30

4 Experimental Results

We set color templates using these fuzzy memberships in white LED light source with indoor condition. The color of harness line that was used in this research includes the 9 colors, has blue, black, purple, green, yellow, orange, red, brown, and gray. After making membership functions for the 9 colors as templates, five unknown fuzzy membership functions that came from each unseen line color was compared with the stored templates. These color inputs are purity colors, purity colors (except red like

Table 5. Results of color recognition (gray line: NN result)

Input data	Gray line				
Compare TEMPLATE 1	**L**	**S**	**R**	**G**	**B**
Black [0, 0]	0.2084	0	0.0219	0.0329	0.0635
Gray [0, 1]	**0.2344**	**0.4468**	**0.4624**	**0.5773**	**0.3813**
Purple [1, 0]	0.0044	0.0152	0.2181	0.5549	0.0223
NN-Result	[0, 1]				

colors: brown, red, and orange) are well separated to use of hue elements distributions. The recognition results of the proposed algorithm in TEMPATE 1 case is shown in Table 5 for the gray line inputs.

5 Conclusion

Current methods of color recognition based on the color information of a pixel or the mean value around pixels are applied with difficulty in industrial fields because of the subjective measurements that come from several values according to the strength, direction, and color temperature of light and the quality of the material of an object. In this paper, the proposed algorithm could recognize harness line colors with an HLS color coordination system that is similar to human's color-cognitive process. A fuzzy similarity measure of color distribution was employed to compare fuzzy memberships in the region of the area of interest.

Acknowledgement

This work was supported by "Research Center for Logistics Information Technology (LIT)" hosted by the Ministry of Education & Human Resources Development in Korea.

References

1. Chien, B., Cheng, M.: A Color Image Segmentation Approach Based on Fuzzy Similarity Measure. Proceedings of IEEE International Conf. on Fuzzy Systems, Vol. 1. (2002) 449-454
2. Palus, H., Bereska, D.: The Comparison between Transformations from RGB Colour Space to IHS Colour Space, Used For Object Recognition, Image Processing and Its Applications. Fifth International Conference (1995) 825-827
3. Wang , Z., Klir, G. T.: Fuzzy Measure Theory. New York: Plenum Press (1992)
4. Kosko, B.: Neural Networks and Fuzzy Systems. Prentice-Hall, Englewood Cliffs, NJ (1992)

Design of a High Precision Temperature Measurement System

Kenan Danisman[1], Ilker Dalkiran[1], and Fatih V. Celebi[2]

[1] Department of Electrical and Electronics Engineering, Erciyes University, Muhendislik, Fakultesi, 38035, Kayseri-Turkey
{danismak, ilkerd}@erciyes.edu.tr
[2] Faculty of Engineering, Baskent University, Baglica Kampusu, 06530, Ankara-Turkey
fatih@baskent.edu.tr

Abstract. An experimental method is designed and proposed in order to estimate the non-linearity, test and the calibration of a thermocouple using artificial neural network (ANN) based algorithms integrated in a virtual instrument (VI). An ANN and a data acquisition board with signal conditioning unit designed are used for data optimization and to collect experimental data respectively. In both training and testing phases of the ANN, Wavetek 9100 calibration unit is used to obtain the experimental data. After the successful training completion of the ANN, it is used as a neural linearizer to calculate the temperature from the thermocouple's output voltage.

1 Introduction

Sensors are one of the most important elements used in many instrumentation circuits. They are used in many industrial applications [1] and take a certain form of input (temperature, pressure, altitude, etc.) and convert it into readings that can be interpreted. Many types of sensors are nonlinear in nature from which a linear output is desired. There are many different sensors for temperature measurement and thermocouples are the most commonly used [2]. They are preferred in industrial applications due to their low cost, wide operation range, fast response time and accurate when their peculiarities are understood. Thermocouples have also outputs nonlinearly related to temperature. Therefore, sensor modeling and linearization techniques are necessary. To solve the linearization problem of a sensor, there are generally two methods proposed. The first one requires nonlinear analog circuit and the second uses numerical methods that are computed by microprocessor or computer [1-4]. Analog circuits are frequently used for improving the linearity of the sensor characteristics, which implies additional analog hardware and typical problems associated to analog circuits such as temperature drift, gain and offset error. Using the second method, sensor nonlinearities can be compensated by means of arithmetic operations, if an accurate sensor model is available (direct computation of the polynomials), or use of a multidimensional look-up table. Direct computation of the polynomial method is more accurate but takes a long time for computation, while the look-up table method, though faster, is not very accurate [4-7].

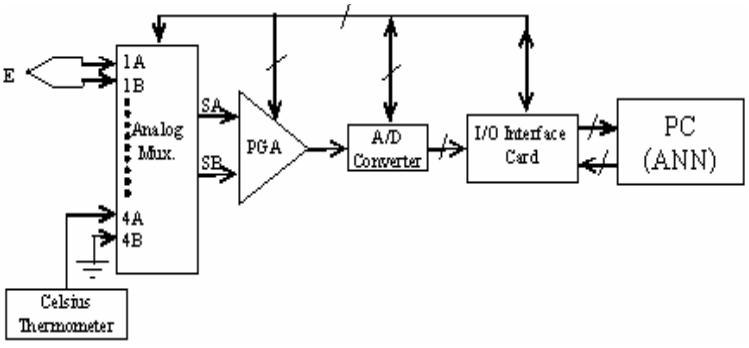
J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 997 – 1004, 2006.

In recent years, application of ANNs has emerged as a promising area of research in the field of instrumentation and measurement [8-12]. It provides a neurocomputing approach for solving complex problems especially in nonlinear system modelling which the network itself is a nonlinear system. This is extremely useful when the area of interest is absolutely nonlinear including the experimental data that is used for training. One of the most powerful uses of ANNs is in function approximation (curve fitting) [13, 14]. Interpolation based on ANN provides lower interpolation errors when compared with conventional numerical interpolation [10, 15].

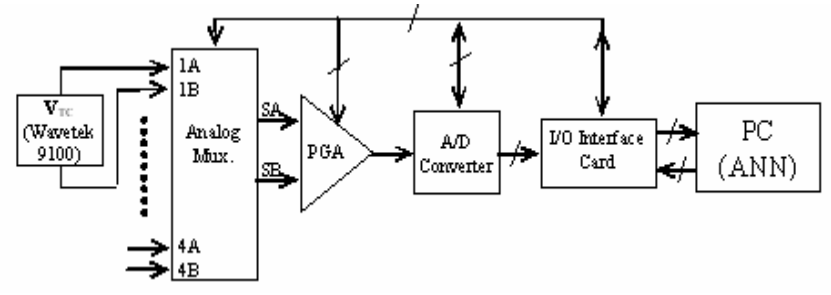
In the work we present here, high precision temperature measurement system based on ANN approach is proposed. The calibrating data is obtained by Wavetek 9100 calibration unit that is necessary for the training and the testing phases of the ANN. The hardware and software parts of the system are integrated in a VI used for system operation and calibration. The ANN is matched to the calibrating data by providing a desired final error. The mean square error between calibration and the ANN modeled data is minimized in terms of the structure, number of layers, and number of neurons by the developed software.

2 System Hardware

A thermocouple generates a voltage proportional to the measurement junction temperature at mV levels while the cold junction temperature is constant. In order to make an accurate measurement the cold junction temperature must be known. Figure 1(a) shows the block diagram of the temperature measurement system designed via an ANN in the operation phase. It consists of a thermocouple (type E) exposed to a desired temperature, including signal conditioning circuit with 16-bit analog to digital converter (ADC) and Input/Output interface card interfacing with a computer. The designed signal conditioning circuit has a programmable gain instrumentation amplifier (PGA204BP) with the gain of 1, 10, 100 and 1000, a 16-bit ADC (AD976A), an AD595 monolithic thermocouple amplifier with cold junction compensation which is configured as a stand-alone Celsius thermometer and a 4 channel analog multiplexer (ADG529A) which select the thermocouple or output of Celsius thermometer. The AD976A is a high speed, low power 16-bit A/D converter that operates from a single 5V supply. This part contains a successive approximation, switched capacitor ADC, an internal 2.5V reference and a high speed parallel interface. Accuracy of the system depends directly on step size of ADC. With a ±10V inputs, one LSB of AD976A is 305μV. When AD595 is used as a Celsius thermometer, the thermocouple is omitted, and the differential inputs are shunted together to common. In this mode, AD595 generate a voltage with a scale factor of 10mV/°C and its output is used for cold junction temperature data that the written software is used. Some important characteristics of the AD595 are: operation temperature range -55 to 125°C; stability vs. temperature: ±0.05°C/°C and sensitivity: 10mV/°C. Output signal of PGA204BP is digitized by AD976A which its output is connected to the I/O interface card and transferred to a personal computer where data reduction and optimization are implemented.



(a)



(b)

Fig. 1. Measurement system block diagram: (a) operational phase, (b) calibration phase

To establish the ANNs weights and biases, during the calibrating phase (ANN training phase), Wavetek 9100 calibration unit, with the accuracy of ±0.006%+4.16μV in the range of 000.000mV to 320.000mV, is connected to the terminals of analog multiplexer to generate tabled thermocouple voltages as shown in Figure 1 (b). This voltage is used as the input of the ANN, and thermocouple temperature without cold junction compensation is the output of the ANN. In the operation phase (Figure 1(a)), in order to make the cold junction compensation, data taken from Celsius thermometer output is used. The output value of ANN is shifted by the environment temperature that is obtained by Celsius thermometer. Then this value is displayed on the VI as the thermocouple temperature.

The developed VI is used to acquire the data for ANN training phase and to show the calculated temperature in the operation phase. Figure 2 shows the front panel of the VI. The main features associated with this instrument are: display of the measured temperature and corresponding output voltage from conditioning circuit for collecting the data in the calibrating phase and actual temperature with cold junction compensation in the operation phase. The system is controlled by the software written in both operation and calibration phases.

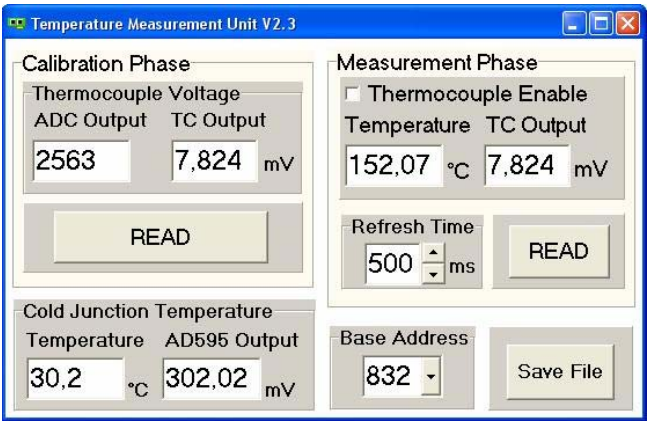


Fig. 2. Front panel of the Virtual Instrument

3 Artificial Neural Network

ANNs are based on the mechanism of the biologically inspired brain model. ANNs are feed-forward networks and universal approximators. They are trained and learned through experience not from programming. They are formed by interconnections of simple processing elements, or neurons with adjustable weights, which constitute the neural structure and are organized in layers. Each artificial neuron has weighted inputs, summation and activation functions and output. The behaviour of the overall ANN depends upon the operations mentioned on the artificial neurons, the learning rule and the architecture of the network. During the training (learning), the weights between the neurons are adjusted according to some criterion (The mean square error between the target output and the measured value for all the training set falls below a predetermined threshold) or the maximum allowable number of epochs is reached. Although the training is a time consuming process, it can be done beforehand, offline. The trained neural network is then tested using data was previously unseen during training.

MLPs are the simplest and most commonly used neural network architectures [19]. They consists of input, output and one or more hidden layers with a predefined number of neurons. The neurons in the input layer only act as buffers for distributing the input signals x_i to neurons in the hidden layer. Each neuron j in the hidden layer sums up its input signals x_i, after weighting them with the strengths of the respective connections w_{ji} from the input layer and computes its output y_j as a function f of the sum, namely

$$y_j = f\left(\sum w_{ji} x_i\right) \tag{1}$$

where f is one of the activation functions used in ANN architecture.

Training a neural network consists of adjusting the network weights using different learning algorithms. A learning algorithm gives $\Delta w_{ji}(t)$ in the weight of a connection

between neurons i and j at time t. The weights are then updated according to the following formula:

$$w_{ji}(t+1) = w_{ji}(t) + \Delta w_{ji}(t+1) \tag{2}$$

There are many available learning algorithms in the literature [13-21]. The algorithms used to train ANNs in this study are Levenberg–Marquardt (LM) [15, 16], Broyden–Fletcher–Goldfarb–Shanno (BFGS) [17], Bayesian Regularization (BR) [18,19], Conjugate gradient backpropagation with Fletcher-Reeves updates (CGF) [20], and Resilient back-propagation (RP) [21] algorithms.

3.1 Neural Linearization

In this paper, the multilayered perceptron (MLP) neural network architecture is used as a neural linearizer. The proposed technique involves an ANN to evaluate the thermocouple temperature (ANN output) when thermocouple output voltage is given as input. Training the ANN with the use of mentioned learning algorithm to calculate the temperature involves presenting it with different sets of input values and corresponding measured values. Differences between the target output and the actual output of the ANN are evaluated by the learning algorithm to adapt the weights using equations (1) and (2).

The experimental data taken from thermocouple data sheets are used in this investigation [22]. These data sheets are prepared for a particular junction temperature (usually 0°C). The ANN is trained with 80 thermocouple temperatures that is uniformly distributed between -200 and 1000°C which is obtained in the calibration phase. However the performance of the final network with the training set is not an unbiased estimate of its performance on the universe of possible inputs, and an independent test set is required to evaluate the network performance after training. Therefore, the other data set of 20 thermocouple temperatures that is uniformly distributed between -200 and 1000°C, is used in the test process.

The input and output data tuples are normalized between -1.0 and 1.0 before training. After several trials with different learning algorithms and with different network configurations, it is found that the most suitable network configuration is 1 X 7 X 3 X 1 with the LM algorithm. This means that the number of neurons is 7 for the first hidden layer and 3 for the second hidden layer respectively. The input and output layers have the linear activation function and the hidden layers have the hyperbolic tangent sigmoid activation function. The number of epoch is 1000 for training. It is important to note that the criteria for too small and too big hidden layer neuron numbers depend on a large number of factors, like ANN type, training set characteristics and type of application. This topic is still under special attention of artificial intelligence researchers today.

4 Results and Conclusion

The developed ANN models are trained and tested with the use of different learning algorithms called LM, BR, CGF, RP and BFGS to obtain better performance and faster convergence with simpler structure. Table 1 shows the errors from the complete

learning algorithms used in the analysis for the same network configuration mentioned above. When the performances of the neural models are compared with each other, the best result for the training and the test are obtained from the model trained with the LM algorithm. The training and test errors (MSE, mean square error) of the network for the LM algorithm are $0.7x10^{-9}$ and $1.3x10^{-4}$ respectively. As it is clearly seen from Table 1, the next solution which is closer to LM is obtained from BR algorithm. Among neural models presented here, the worst results are obtained from the RP method for this particular application. It should be emphasized that the accuracy of learning algorithms in general depends on selecting appropriate learning parameters, network configurations and initializations.

Table 1. The mean square errors (MSE) obtained from ANN models trained with different learning algorithms for the calculation of temperature

ANN models trained with	MSE in training (°C)	MSE in test (°C)
LM	$0.7x10^{-9}$	$1.3x10^{-4}$
BR	$0.8x10^{-9}$	$1.9x10^{-4}$
BFGS	$0.6x10^{-7}$	$1.96x10^{-2}$
CGF	$3.7x10^{-6}$	4.2
RP	$0.86x10^{-5}$	4.9

Figure 3 represents the percentage test error of the network trained with LM for type E thermocouple. As it is clearly seen from Figure 3, the maximum percentage error becomes lower than 0.3%. The average percentage error is greater than 0.1% for temperatures between -200 and 200°C, the reason being that in this range the thermocouples are strongly nonlinear. However, it is obvious for best fit in the range -200 to 200°C that the number of training data set must be increased. The normalized error convergence curves in the learning algorithms used in the analysis for 1000

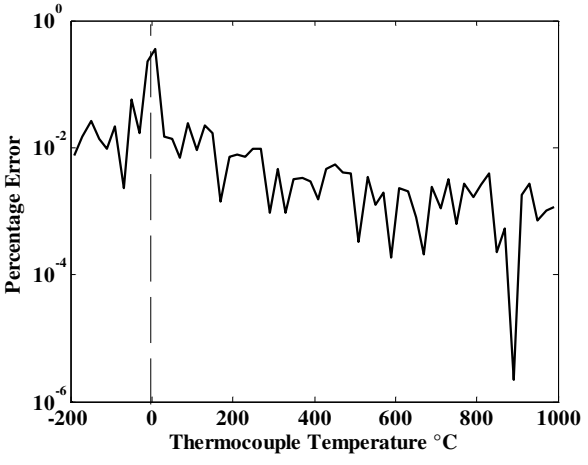


Fig. 3. Percentage measurement error of the ANN trained with LM algorithm

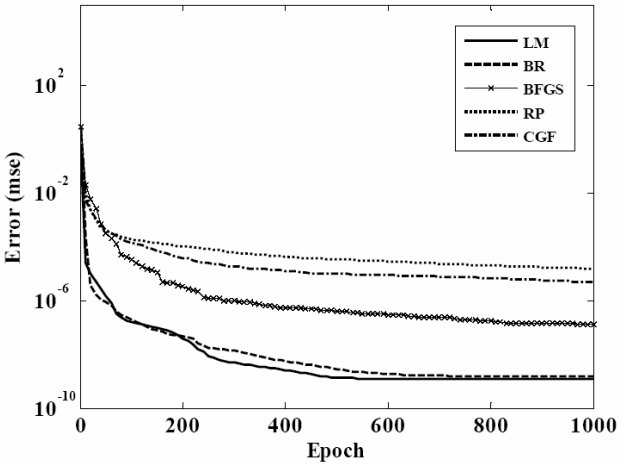


Fig. 4. Learning (convergence) characteristics of the ANN for different learning algorithms used in the analysis for 1000 epoch

epochs are graphically shown in Figure 4. For simulating the neural model, the ANN is trained, to minimize the MSE. As the learning proceeds, the mean square error progressively decreases and finally attains a steady state minimum value as shown in Figure 4.

As a conclusion, a technique for high precision temperature measurement based on an ANN model is proposed in this paper. The training process for MLP ANNs is performed successfully in this study with the use of LM algorithm which gives the best result among other learning algorithms. Gain and offset errors of the signal conditioning circuit are automatically cancelled as a consequence of the usage of the ANN technique. The proposed method has a large area of applications in all sensor based measurement systems where the sensor nonlinearity is the main factor to be considered. The technique has a potential future in the field of instrumentation and measurement.

Acknowledgment

The authors would like to thank to the collective of Laboratory of Electrical Measurement and Calibration of Kayseri ve Civari Elektrik T.A.S. for providing the technical support.

References

1. Areny, R.P., Webster, J.G.: Sensors and Signal Conditioning, 2nd Edn, Wiley, New York (2001)
2. Michalski, L., Eckersdorf, K., Kucharski, J., Mcghee, J.: Temperature Measurement, 2nd Edn, Wiley Interscience (2001)

3. Klopfenstein, L.R.: Software Linearization Techniques for Thermocouples, Thermistors, and Rtds. Isa Transactions 33 (1994) 293-305
4. Patranabis, D., Ghosh, S., Bakshi, C.: Linearizing Transducer Characteristics. IEEE Trans. Inst. Meas. 37 (1988) 66–69
5. Mahana, P.N., Trofimenkoff, F.N.: Transducer Output Signal Processing Using an Eight-Bit Microcomputer. IEEE Trans. Instrum. Meas. 35 (1986) 182-186
6. Anvekar, D.K., Sonde, B.S.: Transducer Output Signal Processing Using Dual and Triple Microprocessor Systems. IEEE Trans. Instrum. Meas. 38 (1989) 834-336
7. Flammini, A., Marioli, D., Taroni, A.: Transducer Output Signal Processing Using an Optimal Look-Up Table in Microcontroller-Based Systems. Electron. Lett. 33 (1997) 1197-1198
8. Meireles, M.R. G., Almedia, P.E.M., Simoes, M.G.: A Comprehensive Review for Industrial Applicability of Artificial Neural Networks. IEEE Trans. Ind. Electron. 50 (2003) 585-601
9. Daponte, P., Grimaldi, D.: Artificial Neural Networks in Measurements. Measurement 23 (1998) 93-115
10. Attari, M., Boudjema, F., Heniche, M.: An Artificial Neural Network to Linearize a G (Tungsten Vs. Tungsten 26% Rhenium) Thermocouple Characteristic in the Range of Zero to 2000^{o}C. Proc. of the IEEE Int. Symp. on Industrial Electronics Isie '95 1 (1995) 176-180
11. Marques, N.J.M., Martin-del-Brio, B.: Sensor Linearization with Neural Networks. IEEE Trans. Ind. Electron. 48 (2001) 1288-1290
12. Pereira, J.M.D., Postolache, O., Girao, P.M.B.S., Cretu, M.: Minimizing Temperature Drift Errors Conditioning Circuits Using Artificial Neural Networks. IEEE Trans. Instrum. Meas. 49 (2000) 1122-1127
13. Haykin, S.: Neural Networks: A Comprehensive Foundation. 2nd edn, Prentice Hall, New Jersey (1999)
14. Rojas, R.: Neural Networks, A Systematic Introduction. Springer-Verlag, New York, (1996)
15. Ham, F.M., Kostanic, I.: Principles of Neurocomputing for Science and Engineering. Mc-Graw Hill, New York (2001)
16. Hagan, M.T., Menhaj, M.: Training Feedforward Networks with the Marquardt Algorithm. IEEE Neural Networks 5(6) (1994) 989-993
17. Dennis, J. E., Schnabel, R.B.: Numerical Methods for Unconstrained Optimization and Nonlinear Equations. Prentice-Hall, New Jersey (1983)
18. Foresee, F.D., Hagan, M.T.: Gauss-Newton Approximation to Bayesian Regularization, Proc. of Int. Conference on Neural Networks ICNN'97 (1997) 1930-1935
19. MacKay, D.J.C.: Bayesian Interpolation. Neural Computation 4 (1992) 415-447
20. Hagan, M.T., Demuth, H.B., Beale, M.H.: Neural Network Design. PWS Publishing, Boston (1996)
21. Riedmiller, M., Braun, H.: A Direct Adaptive Method for Faster Backpropagation Learning: the RPROP Algorithm. Proc. of the IEEE Int. Con. on Neural Networks 1 (1993) 586–591
22. NIST ITS-90 Thermocouple Database, http://srdata.nist.gov/its90/main/.

Integrating Computational Fluid Dynamics and Neural Networks to Predict Temperature Distribution of the Semiconductor Chip with Multi-heat Sources

Yean-Der Kuan[1], Yao-Wen Hsueh[1], Hsin-Chung Lien[1], and Wen-Ping Chen[2]

[1] Department of Mechanical Engineering,
Northern Taiwan Institute of Science and Technology,
2, Xue Yuan Rd., Peitou, Taipei 112, Taiwan, R.O.C.
[2] Institute of Mechatronic Engineering,
Northern Taiwan Institute of Science and Technology,
2, Xue Yuan Rd., Peitou, Taipei 112, Taiwan, R.O.C.
{ydkuan, ywhsueh, hclien}@ntist.edu.tw,
m9311102@st.ntist.edu.tw

Abstract. In this paper, an artificial intelligent system to predict the temperature distribution of the semiconductor chip with multi-heat sources is presented by integrating the back-propagation neural network (BNN) and the computational fluid dynamics (CFD) techniques. Six randomly generated coordinates of three power sections on the chip die are the inputs and sixty-four temperature monitoring points on the top of the chip die are the outputs. In the present methodology, one hundred sets of training data obtained from the CFD simulations results were sent to the BNN for the intelligent training. There are other sixteen generated input sets to be the test data and compared the results between CFD simulation and BNN, it shows that the BNN model is able to accurately estimate the corresponding temperature distribution as well as the maximum temperature values under different power distribution after well trained.

1 Introduction

Thermal management in semiconductor electronic and packaging worlds is facing increasing challenges in the task of dissipating the heat from integrated circuit while still maintaining in the acceptable junction temperatures. The continuous miniaturization and intensive application of the electronic devices result in rapid increase in the power density on the electronic dies. International Technology Roadmap for Semiconductor projections [1] indicates that thermal management challenges will significantly increase in the future due to increasing power, decreasing junction temperature, and a continuous need to have cost effective solutions. The assemblage in small volumes yields a tremendous heat generation even if the heat production is low in the majority of electronic devices. The rate of heat removal for those components with high heat flux has become very important in order to keep the electronics devices at an acceptable operating temperature [2]. Kraus and Bar-Cohen [3] introduced the concept to apply the conventional heat transfer to the field of the cooling in the electronics cooling. In general, thermal management in the electronics includes the levels of the system, heatsink, PCB board, and package. The detailed

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1005 – 1013, 2006.

introduction and review of the basic principle, design, and analysis for the heatsinks are shown in the book of Kraus and Bar-Cohen [4]. Mostly in the past, thermal design and analysis, the chip power was mostly assumed as uniform distribution. However, in modern versatile-function demands and advanced semiconductor chip packaging technologies, the chips tend to non-uniform power distribution which increases the possibility of damage due to local hot spots. Therefore, a good prediction on the maximum temperature of a chip die is important to avoid the thermal damage.

The computational fluid dynamics (CFD) has been tremendously applied to the heatsink design and the heat transfer in the electronic systems [5-7]. But very few literatures discussed a chip with non-uniform or multi-hest sources. Goh et al. [8] made the Lagrangian interpolation to predict the temperature distribution of a chip die divided by 25 heat sources through the one hundred CFD simulations by ANSYS. Moreover, they made another 150 simulation data to derive a nonlinear equation and optimal estimation via genetic algorithm [9]. However, the coordinate of how sources were fixed and not randomly located. Chen et al. [10] integrated the finite element method (FEA) and response surface methods to make highly-nonlinear mathematical relationships between two and three heat source locations and junction temperatures of the chip. However, the mathematic relationships were too complicated to derive in many of real problems.

Even the CFD simulation could reduce the cost and time of the design cycle, but the thermal designers still need to make several trial and errors to reach an acceptable result in each case. Therefore, a more efficient methodology is desired to conform the diversified requirements. BNN (back-propagation neural network) has been successfully applied to many fields such as efficiently resolving problems with classification and prediction studies [11-15].

However, very limit literatures could be found to apply neural network to the field electronics cooling. Kos [16] tried to use BNN to make the better placement of the cooling components of an electronic power system, none applied to the heatsink design in the literature. Kuan and Lien [17] developed an intelligent system to make the heatsink designs under the combination of the parameters of heatsink length, heatsink width, fin thickness, fin gap, fin height, and heatsink base height. In this paper, the methodology of integrating BNN and CFD is further applied to make the temperature distribution on the chip die with multi-heat sources.

2 Problem Description

In this paper, the design on the locations of three dies in a FP-PBGA package type of semiconductor chip is presented by predicting temperature distribution via integrating BNN and CFD techniques. Figure 1 illustrates the structure, dimensions, and related thermal properties of the FP-PBGA package chip. The chip is 42.5mm x 42.5mm x 3.04mm, with three planar heat sources at the bottom of die. Each heat source is 6.4mm x 3.6mm and generates 25W heat. The coordinates at the left-bottom vertex of the three heat sources, (X1, Y1), (X2, Y2), and (X3, Y3), are the 6 inputs of the intelligent BNN system as shown in the Figure 2. In addition, there are 64 outputs of the BNN, which are the CFD simulation results on the top surface temperature values of the chip die (Figure 3).

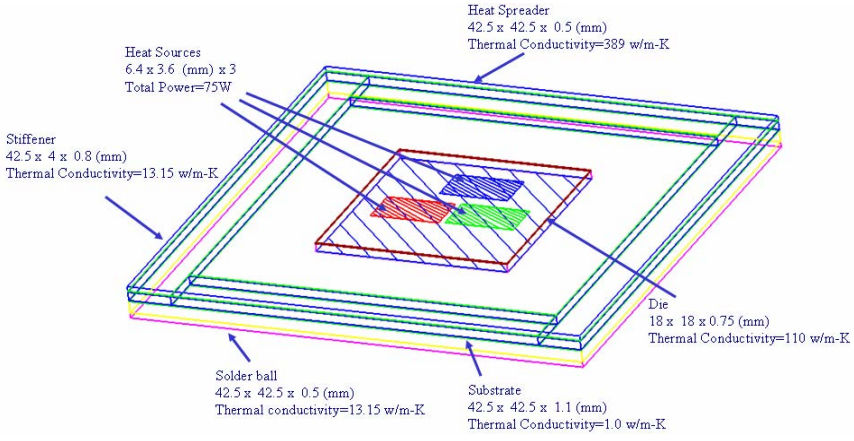


Fig. 1. The illustration of the structure/dimensions and related thermal properties of the FP-PBGA package chip

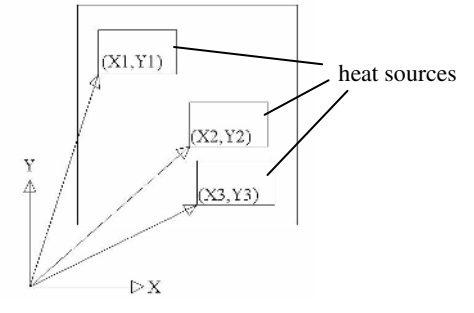


Fig. 2. The description of six inputs in the intelligent system

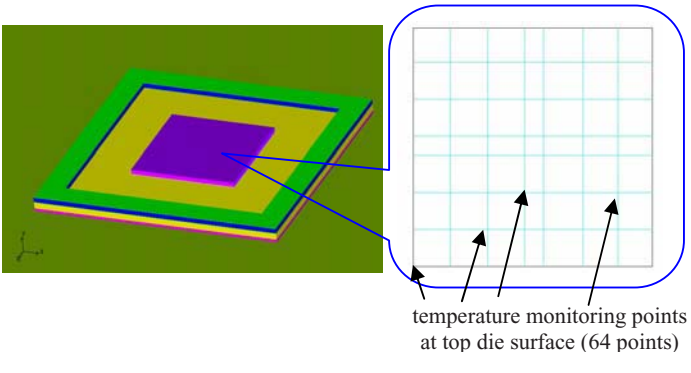


Fig. 3. The 64 temperature monitoring points at the top die surface of the chip

3 Basic Theories

3.1 Backpropagation Neural Network (BNN)

The neural network has three layers, the input, the hidden, and the output layer. The BNN forward learning process optimizes the connection weighting u_{ij} from the input layer node x_i to the hidden layer nodes h_j and w_{jk} from the hidden layer node h_j to the output layer node y_k based on the input properties x_i. This is shown in equation 2 and 3:

$$h_j = f(h_{input(j)}) = \frac{1}{1+e^{-h_{input(j)}}} \tag{1}$$

$$y_k = f(y_{input(k)}) = \frac{1}{1+e^{-y_{input(k)}}} \tag{2}$$

where $h_{input(j)} = \sum_i (u_{ij} x_i - \theta_j)$ and $y_{input(k)} = \sum_j (w_{jk} h_j - \theta_k)$ represent all inputs to the hidden layer node *j* and all inputs to the output layer node *k*, respectively; and θ is the bias. The difference between the theoretical output y_k and the actual output t_k is the error of the output node *k*. That is, the neural network error function is $e_k = (t_k - y_k)$ and the cost function is $E = \frac{1}{2}\sum_k (t_k - y_k)^2$. The BNN backward learning process calculates backpropagation error functions, $\delta_k^w = (t_k - y_k) y_k (1 - y_k)$ and $\delta_j^u = (\sum_{k=1}^{m} \delta_k^w w_{jk}) h_j (1 - h_j)$.

3.2 Computation Fluid Dynamics (CFD)

The CFD is to use numerical process and make the iterative calculation to solve the heat and fluid related governing equations. Through the numerical simulation, the fluid flow, heat transfer, mass transfer, chemical reactions, and related phenomena could be predicted. Fluid flow and heat transfer could be solved simultaneously in the CFD process. One of the CFD scheme, finite volume method (FVM) is widely used in the computational fluid dynamics field. In the FVM, the domain is discredited into a finite set of control volumes or cells. The general conservation (transport) equation for mass, momentum, energy, etc., are discredited into algebraic equations. The general conversation equation is shown in the Equation 3. The CFD simulation is done by Icepak, a finite-volume based CFD software [18].

$$\underbrace{\frac{\partial}{\partial t}\int_V \rho\phi dV}_{\text{Unsteady}} + \underbrace{\oint_A \rho\phi V \cdot dA}_{\text{Convection}} = \underbrace{\oint_A \Gamma\nabla\phi \cdot dA}_{\text{Diffusion}} + \underbrace{\int_V S_\phi dV}_{\text{Generation}} \tag{3}$$

Eqn.	ϕ
Continuity	1
x-mom.	u
y-mom.	v
energy	h

3.3 The Integration of BNN and CFD

The present methodology is adopting CFD to run 116 trials under the limitation of parameter (geometry) inputs and those input values are random generated. In the 124 sets of data, 100 are taken as the training data, and the rest 16 are taken as the validation data. The accuracy of the BNN model could be estimated after comparison with the CFD results.

4 Results and Discussion

4.1 CFD Simulation

In this research, the chip was mounted on a 100.2mm x 114.2mm x1.6mm PCB and assumed the thermal conductivity of the PCB is 13W/m-K under 35^{O}C ambient temperature. In order to accelerate the calculation speed, the average convection heater transfer coefficients were given to surfaces of the heat-spreader and PCB to simulate the heat removal to the heatsink from the heat spreader and heat exchange between the PCB surface and environment. Figure 4 (a) shows the model boundary conditions of the CFD simulation and Figure 4(b) is the temperature contour plot at the top die surface of the first training data.

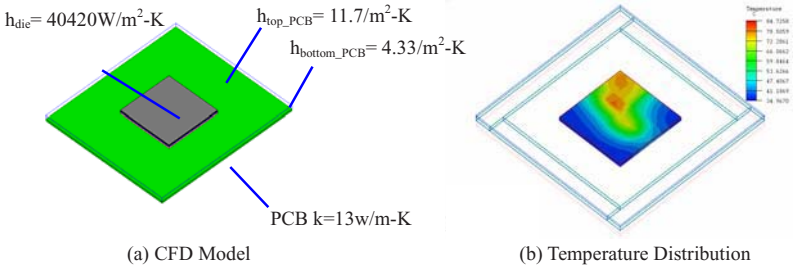


Fig. 4. The CFD model and temperature distribution of the die at the first training data

4.2 BNN Training Results

Figure 5 is the convergent plot of the BNN training process. After 49951 epochs, the BNN model tends to convergent, and MAX error is about 11.73 %, and the RMS is about 4.52%. On the other hand, the BNN training module shows a pretty high estimated accuracy of 95.48%. Figure 6(a) and (b) are the 3D temperature distribution contour plots at top die surface by Icepak and BNN calculations and both are very close to each other in shape and values, moreover, Figure 7 illustrates the 3D error distribution contour plot at that surface for the first training data, the maximum error is 5% and RMS error is 2%. In semiconductor chip thermal design, the most important thing is to keep the maximum die temperature below the acceptable range. Therefore, to make an accurate maximum die surface temperature prediction would significantly help the chip inner layout design. Figure 8 is the comparison of the maximum die

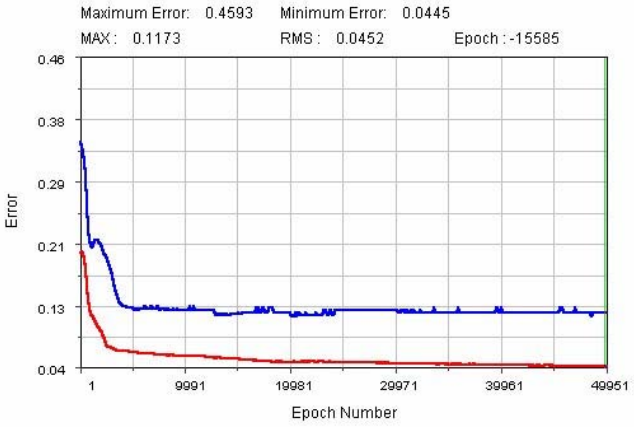
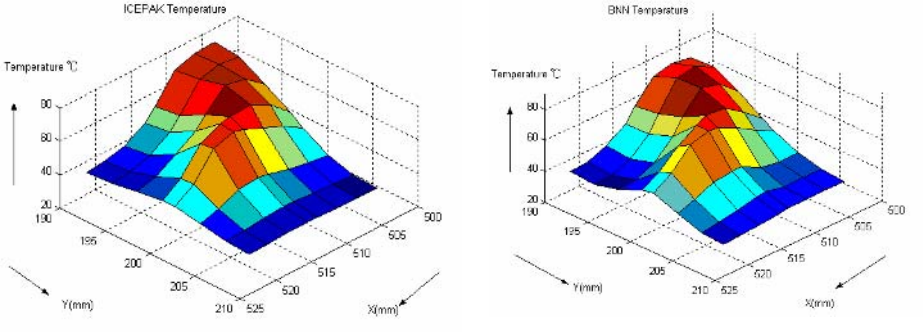


Fig. 5. The convergent plot of the model under BNN training based on 100 training data, the RMS falls into 4.52% and the maximum error is about 11.73%



(a) Icepak Results (b) BNN Results

Fig. 6. The 3D contour plots of the temperature distribution at the top die surface by Icepak and BNN results for the first training data

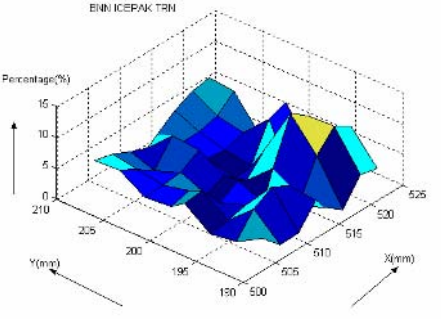


Fig. 7. The 3D contour plots of the error percentage distribution at top die surface by Icepak and BNN results for the first training data

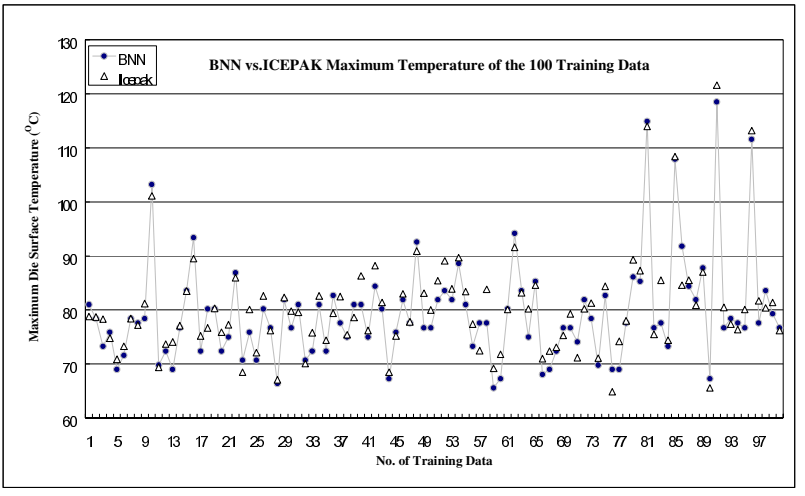


Fig. 8. The comparison of the maximum die surface temperature between BNN and Icepak results for the 100 training data

surface temperature between BNN and Icepak simulations. The results show the average error is 2.93% and maximum error is 9.24% which implies over 97% accuracy in the maximum temperature prediction.

4.3 Validation

The validation of the intelligent was done by 16 randomly generated training data. Figure 9 is the comparison of the maximum die surface temperature between BNN and

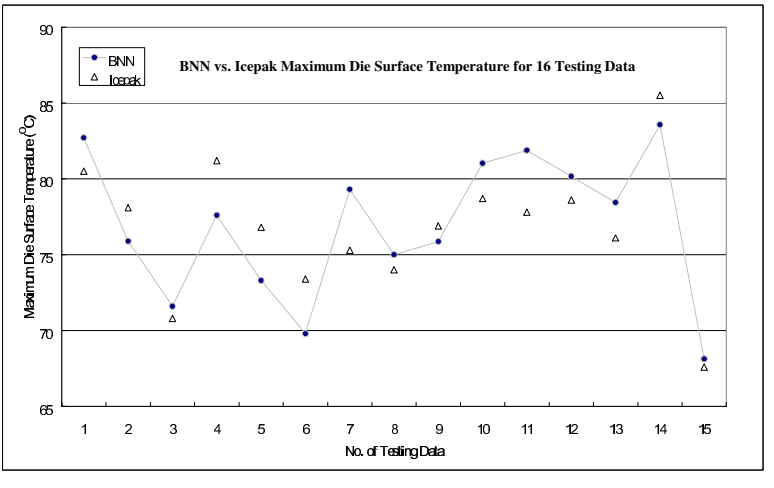


Fig. 9. The comparison of the maximum die surface temperature between BNN and Icepak results for the 100 testing data

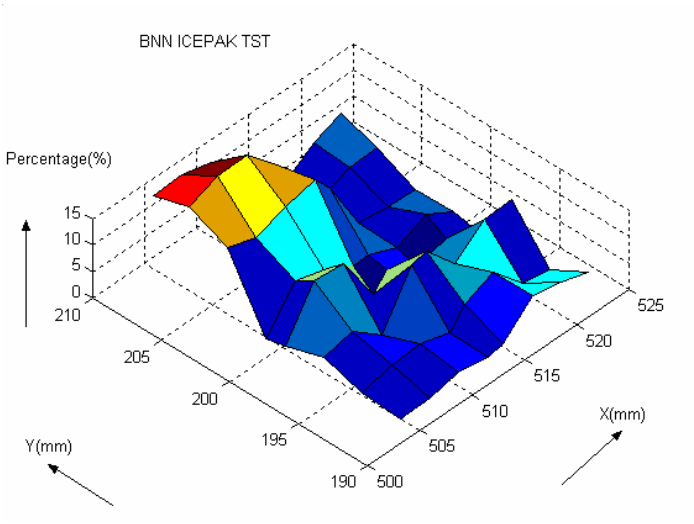


Fig. 10. The 3D contour plots of the error percentage distribution at top die surface by Icepak and BNN results for the first training data

Icepak simulations. The results show the average error is 2.96% and maximum error is 5.31%, which implies over 97% accuracy in the maximum temperature prediction on the prediction of maximum die surface temperature based on those 16 testing data. Figure 10 is the 3D error percentage distribution at the top surface of the chip die for the first training data, the maximum error is 7% and RMS error is 3%.

5 Conclusions

In this paper, the integration of BNN and computational fluid dynamics (CFD) has been successfully applied to the extrude heatsink design. The CFD simulations makes 116 sets of data, 100 of them are taken as the training samples, and other are taken to be the validation ones. According to the comparison between the BNN and CFD results, the maximum error is about 16.43% and the RMS is about 7.63%, and the BNN model could make a very fast estimation under acceptable accuracy. So after well trained under the training and testing data taken from the CFD, the BNN model could give quick temperature distribution as well as maximum die surface temperature under three heat sources at different locations; moreover, the BNN model could even help to make a better design of the heat locations on the chip die during the layout. Following the BNN designs, the CFD could help to make the final adjustments and this will save a lot of design cycle and cost.

Acknowledgements

The authors appreciate the finial support by the Taiwan Semiconductor Manufacturing Company, Ltd., and the acknowledgement would be extended to the assistance from Northern Taiwan Institute of Science and Technology throughout the research.

References

1. The International Technology Roadmap for Semiconductors (2001)
2. Soule, C.A.: Future Trends in Heat Sink Design, Electronics Cooling 7(1) (2001) 18-27
3. Kraus, A. and Bar-Cohen, A.: Thermal Analysis and Control of Electronic Equipment. Hemisphere Publishing Corp. (1983)
4. Kraus, A.D. and Bar-Cohen, A.: Design and Analysis of Heat Sinks. John Wiley & Sons, Inc. (1995)
5. Wong, H. and Lee, T. Y.: Thermal Evaluation of a PowerPC 620 Multi-Processor in a Multi-Processor Computer. IEEE Transaction on Components, Packaging, and Manufacturing Technology – Part A 19(4) (Dec. 1996) 469-477
6. Chang. J. Y., Yu, C. W., and Webb, R. L.: Identification of Minimum Air Flow Design for a Desktop Computer Using CFD Modeling. Journal of Electronic Packaging. Transactions of the ASME 123 (2001) 225-231
7. Yu, C. W. and Webb, R. L.: Thermal Design of a Desktop System Using CFD Analysis, Seventeenth IEEE SEMI-THERM Symposium (2001) 18-26
8. Goh, T. J., Seetharamu, K. N., Quadir, G. A., Zainal, Z. A. and Jeevan, K.: Prediction of Temperature in Silicon Chip with Non-uniform power: A Lagrangian Interpolation Approach. Microelectronics International 21(2) (2004) 29-35
9. Goh, T. J., Seetharamu, K. N., Quadir, G. A., Zainal, Z. A. and Ganeshamoorthy, K. J.: Thermal Investigations of Microelectronic Chip with Non-Uniform Power Distribution: Temperature Prediction and Thermal Placement Design Optimization. Microelectronics International 21(3) (2004) 29-43
10. Cheng, H.-C., Chen, W.-H., and Chung I.-C.: Integration of Simulation and Response Surface Methods for Thermal Design of Multichip Modules. IEEE Transactions on Components and Packaging Technologies 27 (2) (2004) 359-372
11. Lien, H.C., and Lee, S.: A Method of Feature Selection for Textile Yarn Grading Using the Effective Distance Between Clusters, Textile Res. J. 72 (10) (2002) 870-878
12. Lien H.C., and Lee S., Applying Pattern Recognition Principles to Grading Textile Yarns, Textile Res. J. 72 (4) (2002) 320-326
13. Lien, H.C., and Lee, S., Applications of Neural Networks for Grading Textile Yarns, Neural Computing and Applications 13 (2004) 185-193
14. Ludwig L, Sapozhnikova E, Lunin V, Rosenstiel W.: Error Classification and Yield Prediction of Chips in Semiconductor Industrial Applications. Neural Computing & Applications 9 (2000) 202-210
15. Verikas, A., Malmqvist K, Bergman L, Signahl M.: Color Classification by Neural Networks in Graphic Arts, Neural Computing & Applications 7 (1998) 52-64
16. Kos, A.: Approach to Thermal Placement in Power Electronics Using Neural Networks, Proceedings - IEEE International Symposium on Circuits and Systems 4 (1993) 2427-2430
17. Kuan, Y. D. and Lien, H.C.: The Integration of the Neural Network and Computational Fluid Dynamics for the Heatsink Design, Lecture Notes in Computer Science: Advances in Neural Networks -ISNN2005 3498 (2005) 933-938
18. Icepak 4.1 User's Guide, Fluent Inc. (2003)

Modeling and Characterization of Plasma Processes Using Modular Neural Network

Seung Soo Han[1], Dong Sun Seo[2], and Sang Jeen Hong[2]

[1] NPTC & Department of Information Engineering, Myongji University, Yongin, Korea
shan@mju.ac.kr
[2] Department of Electronic Engineering, Myongji University, Yongin, Korea
{sdsphoto, samhong}@mju.ac.kr

Abstract. In semiconductor manufacturing, complex and nonlinear fabrication processes are ubiquitous. Plasma processing such as plasma enhanced chemical vapor deposition (PECVD) and reactive ion etching (RIE) are workhorses in semiconductor fabrication, but also play as yield limiters due the nature of complexity of plasma process. In this paper, modular neural network (MNN) is applied for the purpose of plasma process modeling and characterization in the area of semiconductor manufacturing. MNN consists of a number of local expert networks (LENs) and one gating network. LENs compete using supervised learning to learn different regions of the data space under the supervision of gating network. Once proper MNNs for various responses of interest are established, response surfaces are generated to visually assist the characterization of the processes. As either an alternative or an augmentation to existing methods, this can provide more reliable and flexible flat form of process modeling and characterization in semiconductor manufacturing environment.

1 Introduction

The semiconductor manufacturing trend towards smaller geometry can provide electronic product customers possessing more capable items within their budgets, but this trend also allows no more margins for chip manufacturer in terms of fabrication. To ensure a successful integration of fabrication processes, complete understanding of unit process cannot be over emphasized. Therefore, characterization of unit processes has to be performed in advance to integration.

In semiconductor manufacturing, complex and nonlinear fabrication processes are ubiquitous, and plasma processes, such as plasma enhanced chemical vapor deposition (PECVD) and reactive ion etching (RIE) are typical examples. Both PECVD and RIE are workhorses in semiconductor fabrication, but also play as yield limiter as well, due the nature of complexity and non-linearity of plasma process itself. PECVD and RIE are good candidates for the demonstration of modeling and characterization examples.

Previous studies employed back-propagation algorithm, but modular neural networks (MNNs) have been constructed and evaluated in this study. The advantages of modular neural networks are: 1) learning speed, 2) generalization, and 3) representation [1]. In addition, automated surface response generation routine was augmented

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1014 – 1019, 2006.

for the purpose of process characterization as well. Once an acceptable process model is established, 3-D response surface associated with any two selected process parameters can be generated. As either an alternative or an augmentation to statistical approach, this will provide more reliable and flexible platform of process modeling and characterization. This paper is organized as follows. Section 2 suggests modular neural networks with gradient based Fuzzy *C*-means. Section 3 illustrates modeling and characterization results of the two applications for semiconductor fabrication process.

2 Modular Neural Network

A neural network is said to be modular if the computation performed by the network can be decomposed into two or more modules (subsystems) that operate on distinct inputs without communicating with each other. The outputs of the modules are mediated by an integrating unit, which is not permitted to feed information back to the modules. Recently, modular neural networks (MNNs) have been used for fault diagnosis [2], and for microelectronic circuit yield optimization [3]. The MNNs consists of a number of local expert networks (LENs) and a gating network. LENs compete using supervised learning to learn different regions of the data space under the supervision of gating network, while the same inputs are provided. The gating network, which is unsupervised, has as many outputs as the number of LENs, and its outputs are normalized to sum to unity. The normalized output values are used to weight the output vector from the corresponding LEN. The combination of the supervised and unsupervised learning paradigms takes place in a seamless fashion during training.

Among many competitive learning procedures that have been developed for unsupervised learning in artificial neural networks, the fuzzy *C*-means (FCM) clustering algorithm is commonly used competitive clustering algorithm for general classification tasks [4].

The goal of the learning algorithm is to model the probability distribution of the set of training patterns {**x** and **d**}. Based on the multivariate *Gaussian* distribution of the desired response vector **d** with given input vector **x,** the conditional probability density function is defined as:

$$f(\mathbf{d} \mid \mathbf{x}) = \sum_{i=1}^{K} g_i f(\mathbf{d} \mid \mathbf{x}, i) = \frac{1}{(2\pi)^{q/2}} \sum_{i=1}^{K} g_i \cdot \exp\left(-\frac{1}{2} \left\| \mathbf{d} - \mathbf{y}_i \right\|^2 \right), \quad i = 1,2,\cdots,K \tag{1}$$

where $\|\bullet\|$ is the Euclidean norm of the enclosed vector, q is the dimension of input vector **x**, g is activation generated from the gating network, and i denotes the i^{th} local expert network. Taking natural logarithm of $f(\mathbf{d} \mid \mathbf{x})$ for a monotonically increasing function of its argument and ignoring the constant term, a log-likelihood function can be defined as:

$$l(\mathbf{w}, \mathbf{g}) = \ln f(\mathbf{d} \mid \mathbf{x}) = \ln \sum_{i=1}^{K} g_i \cdot \exp\left(-\frac{1}{2} \left\| \mathbf{d} - \mathbf{y}_i \right\|^2 \right). \tag{2}$$

By employing the membership grade of the FCM, a prior probability μ_i and posterior probability h_i are defined as:

$$\mu_i = 1 \Big/ \sum_{j=1}^{K} \left(\beta_i / \beta_j\right)^2 \text{ and } h_i = \frac{\mu_i \exp\left(-\frac{1}{2}\left\|\mathbf{d}-\mathbf{y}_i\right\|^2\right)}{\sum_{j=1}^{K} \mu_j \exp\left(-\frac{1}{2}\left\|\mathbf{d}-\mathbf{y}_i\right\|^2\right)}, \tag{3}$$

where β_i is the distance from the center of the i^{th} cluster and K is the number of local expert networks. Apply the chain rule for $\Delta\mathbf{w}$:

$$\frac{\partial l}{\partial \mathbf{w}_i} = \frac{\partial l}{\partial y_i}\frac{\partial y_i}{\partial \mathbf{w}_i} = h_i\left(\mathbf{d}-\mathbf{y}_i\right)\mathbf{x}, \tag{4}$$

$$\frac{\partial l}{\partial y_i} = \frac{\mu_i \exp\left(-\frac{1}{2}\left\|\mathbf{d}-\mathbf{y}_i\right\|^2\right)}{\sum_{j=1}^{K} \mu_i \exp\left(-\frac{1}{2}\left\|\mathbf{d}-\mathbf{y}_i\right\|^2\right)} \cdot \left(\mathbf{d}-\mathbf{y}_i\right) = h_i\left(\mathbf{d}-\mathbf{y}_i\right) \text{ and } \frac{\partial y_i}{\partial \mathbf{w}_i} = \mathbf{x}. \tag{5}$$

Therefore, the weight adapting equation for the local expert networks is

$$\mathbf{w}_i(n+1) = \mathbf{w}_i(n) + \Delta\mathbf{w}_i(n) = \mathbf{w}_i(n) + \eta_{LEN} h_i\left(\mathbf{d}-\mathbf{y}_i\right)\mathbf{x}, \tag{6}$$

where η_{LEN} is learning rate of a local expert network.

In the similar manner, the learning algorithm for the gating network can be derived by applying chain rule:

$$\frac{\partial l}{\partial \mathbf{v}_i} = \frac{\partial l}{\partial \mu_i}\frac{\partial \mu_i}{\partial \beta_i}\frac{\partial \beta_i}{\partial \mathbf{v}_i}, \tag{7}$$

where $i = 1,2,\cdots,K$ and $\mathbf{v}_i$ denotes the weight vector of the gating network associated with i^{th} local expert network.

$$\begin{aligned}\frac{\partial l}{\partial \mu_i} &= \frac{\partial}{\partial \mu_i}\ln \sum_{i=1}^{K} \mu_i \exp\left(-\frac{1}{2}\left\|\mathbf{d}-\mathbf{y}_i\right\|\right)^2 = \frac{\exp\left(-\frac{1}{2}\left\|\mathbf{d}-\mathbf{y}_i\right\|^2\right)}{\sum_{i=1}^{K} \mu_i \exp\left(-\frac{1}{2}\left\|\mathbf{d}-\mathbf{y}_i\right\|^2\right)} \\ &= \frac{1}{\mu_i}\frac{\mu_i \exp\left(-\frac{1}{2}\left\|\mathbf{d}-\mathbf{y}_i\right\|^2\right)}{\sum_{i=1}^{K} \mu_i \exp\left(-\frac{1}{2}\left\|\mathbf{d}-\mathbf{y}_i\right\|^2\right)} = \frac{h_i}{\mu_i}\end{aligned} \tag{8}$$

$$\frac{\partial \mu_i}{\partial \beta_i} = -\left(\frac{2}{\beta_i}\cdot\frac{1-u_i}{u_i}\right) \Big/ u_i^{-2} = \frac{2u_i(u_i-1)}{\beta_i} \text{ and } \frac{\partial \beta_i}{\partial \mathbf{v}_i} = \mathbf{x}. \tag{9}$$

Finally, $\partial l/\partial \mathbf{v}_i$ for $\Delta \mathbf{v}_i$ becomes:

$$\frac{\partial l}{\partial \mathbf{v}} = \frac{\partial l}{\partial \mu_i} \cdot \frac{\partial \mu_i}{\partial \beta_i} \cdot \frac{\partial \beta_i}{\partial \mathbf{v}} = \frac{h_i}{\mu_i} \cdot \frac{2\mu_i(\mu_i - 1)}{\beta_i} \cdot \mathbf{x} = \frac{2h_i(\mu_i - 1)}{\beta_i} \mathbf{x} \ . \tag{10}$$

Therefore, updating the gating network can be defined by:

$$\mathbf{v}_i(n+1) = \mathbf{v}_i(n) + \Delta \mathbf{v}_i(n) = \mathbf{v}_i(n) + \eta_{GN} \cdot \frac{2h_i \cdot (\mu_i - 1)}{\beta_i} \cdot \mathbf{x} \ , \tag{11}$$

where η_{GN} denotes the learning rate of the gating network.

3 Modeling and Characterization of the Plasma Processes

3.1 Plasma Enhanced Chemical Vapor Deposition (PECVD)

Employing Plasma Term 700 series dual chamber PECVD, SiO_2 thin film deposition was performed followed by a 2^{5-1} factorial design to increase the efficiency of experiment. The parameters are RF power, Pressure, Temperature, and gas flow rates of O_2 and SiN_4. Once the film quality was measured, the data set was randomly separated 3:1 ratio for training and testing, respectively.

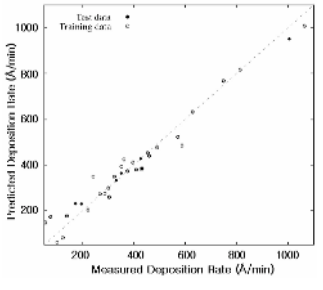


Fig. 1. Model verification with data set

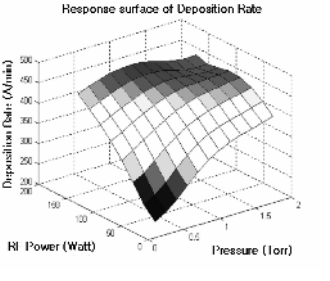


Fig. 2. Surface response plot of PECVD

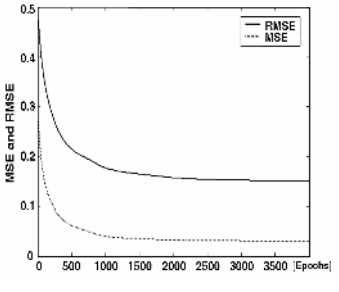


Fig. 3(a). Learning curve for RIE Etch rate

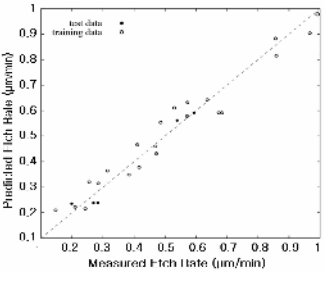


Fig. 3(b). Model verification for Etch rate

Once the model was tested, response surface with respect to the process variables was continuously generated to illustrate the relationships between any two parameters and the deposition rate in Fig. 2. From the response surface, in order to acquire adequate deposition rate, more than 100 *watts* of RF power is desired. It also can be inferred that too low pressure (below 1 *Torr*) slows down the deposition rate even with large amount of RF power. This can be interpreted that fewer molecules to be deposited may reside in the chamber at lower pressure environment, so deposition may slow down.

3.2 Reactive Ion Etching (RIE)

To reduce the number of experiments, a 2^4 factorial design with three center points was initially considered and further augmented with central composite circumscribed design [5, 6]. After the responses of interest were measured, the MNN was trained and tested with data set separated in the similar manner to the PECVD exercise.

Once the modular neural networks (MNNs) were trained, they were tested with the data that never introduced during training. The established models of etch rate and uniformity were graphically evaluated in Fig. 4 (a) and (b). Then, response surfaces were continuously generated to illustrate the relationships between any two parameters and response of interest, which are etch rate of BCB and uniformity in this exercise. Any two selected variables of three were simultaneously varied within their ranges while the rest parameter was set to be its center level, then the prediction of the neural process model was plotted in 3-D (see Fig. 2 and 4). The figures illustrate how two process parameters are related to each other with respect to a etch response of interest. As illustrated in the figures, Maximum amount of gases, O_2=80 sccm and SF_6=10 sccm, may provide highest etch rate, but to improve the Uniformity, 60 sccm of oxygen is desired. We can infer that SF_6 has strong correlation with both responses in the etching of BCB, and this is true for BCB etching because it contains silicon component and fluorinated gas species can break *Si* bond in BCB chemistry.

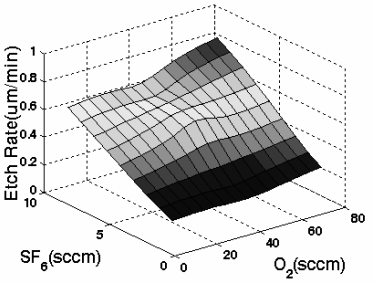


Fig. 4(a). Response surface for Etch rate

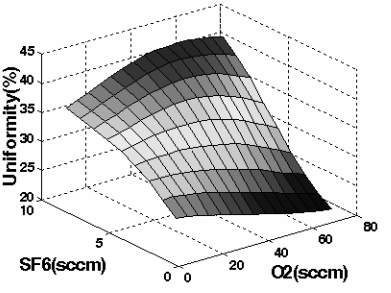


Fig. 4(b). Response surface for Uniformity

4 Conclusions and Future Work

Modular neural network, as a semiconductor manufacturing process modeling and characterization, is introduced as a mean of constructing accurate and robust process

models in the area of semiconductor manufacturing. Process characterization is highly desirable for semiconductor manufacturing in order to maximize the productivity and confront any undesired process change. Considering high volume manufacturing environment, process characterization becomes more valuable. It is possible to perform as many experiment to collect all combinations of process conditions associated with multiple process parameters as possible, but this is obviously time consuming and extremely expensive.

To alleviate these concerns, modular neural networks were employed to increase the efficiency as well as accuracy of process characterization. Generated surface response plot based on the established modular neural networks assisted identifying the relationship between parameters with respect to important process responses. Deposition rate of PECVD and etch rate and uniformity of RIE process was successfully visualized to provide process engineers better understanding of microscopic plasma processes. In this paper, process characterization is limited to plasma process, but it can be further applied to almost all other processes in semiconductor manufacturing process. In addition, suggested MNN can add its value for fault detection and classification (FDC) problems by augmenting time series modeling.

Acknowledgement

This work was supported by ERC program of MOST/KOSEF (Next-generation Power Technology Center) and Inha University ERC, Korea.

References

1. Haykin, S.: Neural Networks. Macmillan College Publishing Company, New York, NY (1994)
2. Rodriguez, C., Rementeria, S., Martin, J., Lafuente, A., Muguerza, J., Perez, J.: A modular neural network approach to fault diagnosis. IEEE Trans. Neur. Net. 7(2) (1996) 326-340
3. Ilumoka, A.: A Modular Neural Network Approach to Microelectronic Circuit Yield Optimization. Micrlelectron. Reliab. 38(4) (1998) 571-580
4. Denoeux, T.: A neural network classifier based on Dempster-Shafer Theory. IEEE Trans. Syst., Man, Cybern 30(2) (2000) 131-150
5. Box, G., Hunter, W., Hunter, J.: Statistics for Experimenters. Wiley, New York NY (1978)
6. Hong, S., May, G., Park, D.: Neural Network Modeling of Reactive Ion Etch Using Optical Emission Spectroscopy Data. IEEE Trans. Semi. Manufac. 16(4) (2003) 598-608

Prediction of Plasma Enhanced Deposition Process Using GA-Optimized GRNN

Byungwhan Kim[1], Dukwoo Lee[1], and Seung Soo Han[2]

[1] Department of Electronic Engineering, Sejong University, Seoul, Korea
kbwhan@sejong.ac.kr
[2] Department of Information Engineering, Myongji University, Yongin, Korea
shan@mju.ac.kr

Abstract. A genetic algorithm (GA)-based optimization of generalized regression neural network (GRNN) was presented and evaluated with statistically characterized plasma deposition data. The film characteristics to model were deposition rate and positive charge density. Model performance was evaluated as a function of two training factors, the spread range and a factor employed for balancing training and prediction errors. For comparison, GRNN models were constructed as well as four types of statistical regression models. Compared to conventional GRNN models, GA-GRNN models improved the prediction accuracy considerably by about 50% for either film characteristic. The improvements over statistical regression models were more pronounced and they were more than 60%. There results clearly reveal that the presented technique can significantly improve conventional GRNN predictions.

1 Introduction

Plasma enhanced chemical vapor deposition (PECVD) plays a critical role in depositing thin films for integrated circuit manufacturing. Many process parameters are involved in PECVD process, including a radio frequency power, pressure, or gases. Due to complex relationships between the process parameters and plasma dynamics, it is extremely difficult to predict deposition characteristics such as the deposition rate or refractive index. To manufacture PECVD in a cost effective way, a computer prediction model is highly in demand. A viable approach to PECVD modeling is to use a neural network technique in conjunction with a statistical experimental design. This is mainly attributed to the fact that the neural network can learn any complex relationships between the process parameters and film characteristics without any assumptions, which are typically required in constructing physical models. Among neural network paradigms, a backpropagation neural network (BPNN) [1] is the most widely adopted in modeling plasma-driven etching or deposition processes [2-3]. Despite the popularity, constructing a BPNN model is complicated since many training parameters are involved in the training process [4]. Another alternative is the generalized regression neural network (GRNN) [5]. Compared to BPNN, adopting GRNN is advantageous in that it contains only one single training factor called *spread*. Unfortunately, the prediction performance of GRNN was much smaller than BPNN as

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1020 – 1027, 2006.

illustrated in predicting pure plasma discharges [6]. Low prediction of GRNN can mainly be ascribed to the fact that all gaussian functions in the pattern layer have the same spread. It is expected that by adopting multi-parameterized training factors the GRNN performance could be considerably improved.

In this study, a technique to construct a GRNN model of PECVD is presented. The drawback stated earlier is circumvented by applying a genetic algorithm (GA) [7] to optimize multi-parameterized spreads. The performance of GA-GRNN model was evaluated with the PECVD data, collected during the deposition of silicon nitride films in a SiH4-NH3-N2 plasma [8]. Previously, the PECVD process has been modeled by BPNN [3, 8] or fuzzy logic [9]. The film characteristics to model are the deposition rate and positive charge density. For comparison, conventional GRNN and statistical regression models were constructed.

Table 1. Experimental Parameters and Ranges

Parameters	Range	Unit
Substrate Temperature	200-400	^{o}C
Pressure	0.6-1.2	Torr
RF Power	20-40	W
NH_3 Flow	1-1.4	sccm
SiH_4 Flow	180-260	sccm
N_2 Flow	0-1000	sccm

2 Experimental Data and GRNN

2.1 Data Acquisition

The SiN films were deposited by using a Plasma-Therm 700 series batch reactor operating at 13.56 MHz. The distance between the electrodes was 2.29 cm and the electrode diameter is 11 inches. The PECVD process was characterized by the Box-Wilson experimental design, consisted of 2^{6-1} fractional factorial experiment and 12 axial points [10]. The resulting 32 experiments plus one experiment corresponding to one center point were used to train GRNN. Prediction performance of the trained GRNN was tested with additional 12 experiments not pertaining to the training data. The experimental parameters that were varied in the design are included in Table 1. Four-inch, float zone p-type silicon wafers of (100) orientation with a resistivity of 2.0 Ω-cm, were used as the substrate. During the deposition, SiH_4 was diluted to 2% in nitrogen. Approximately 0.05 μm silicon nitride was deposited. The deposition rate was measured using a Metricon 2010 prism coupler. Five measurements for each experiment were collected to calculate the average of deposition rates. The charge density was measured by using C-V measurements.

2.2 Generalized Regression Neural Network

GRNN was used to construct a prediction model of PECVD data. The GRNN consists of four layers, including the input layer, pattern layer, summation layer, and output

layer. The first layer is fully connected to the second, pattern layer, where each unit represents a training pattern and its output is a measure of the distance of the input from the stored patterns. Each pattern layer unit is connected to the two neurons in the summation layer: S-summation neuron and D-summation neuron. The S-summation neuron computes the sum of the weighted outputs of the pattern layer while the D-summation neuron calculates the unweighted outputs of the pattern neurons. In the output layer, a prediction to an unknown input vector x is computed as

$$\hat{y}_i(x) = \frac{\sum_{i=1}^{n} y_i \exp[-D(x,x_i)]}{\sum_{i=1}^{n} \exp[-D(x,x_i)]} \tag{1}$$

where n indicates the number of training patterns and the Gaussian D function in (1) is defined as

$$D(x,x_i) = \sum_{j=1}^{p} \left(\frac{x_j - x_{i_j}}{\zeta}\right)^2 \tag{2}$$

where p indicates the number of elements of an input vector. The x_j and x_{i_j} represent the jth element of x and x_i, respectively. The ζ is generally referred to as the spread, whose optimal value is often determined experimentally.

3 Results

3.1 GA-Optimization

Rather than one singe spread, multi-parameterized spreads were adopted for the D function defined in (2). Their complex effects on model performance were optimized by applying GA. Model performance was evaluated in terms of training and prediction error, represented as T-RMSE and P-RMSE, respectively. In GA optimization, a random generator was employed to assign random values to the spreads for all gaussian functions in the pattern layer. The random values were generated within a spread range varying between 0.3 and 1.0. Meanwhile, both errors were balanced in terms of the performance index (PI) defined as

$$PI = \theta \times (\text{P-RMSE}) + (1- \theta) \times (\text{T-RMSE}) \tag{3}$$

where θ is the factor to control certain trade-offs between the two errors. In this study, θ was varied from 0.2 to 0.8 with an increment of 0.2 for a given spread range.

As an illustration, a deposition rate model was constructed at the spread range [0, 0.5] with θ set at 0.5. The initial population was set at 100 potential solutions. Each chromosome was coded in the real value. The roulette wheel selection was employed

for the reproduction. The probabilities of crossover and mutation were set at 0.95 and 0.05, respectively. A particular input setting generated by GA meets a given fitness function expressed as

$$F = \frac{1}{1 + PI} \tag{4}$$

The search process was terminated as the generation number reaches a predefined value of 100. The resulting P-RMSE and fitness are plotted in Fig. 1 as a function of the generation number.

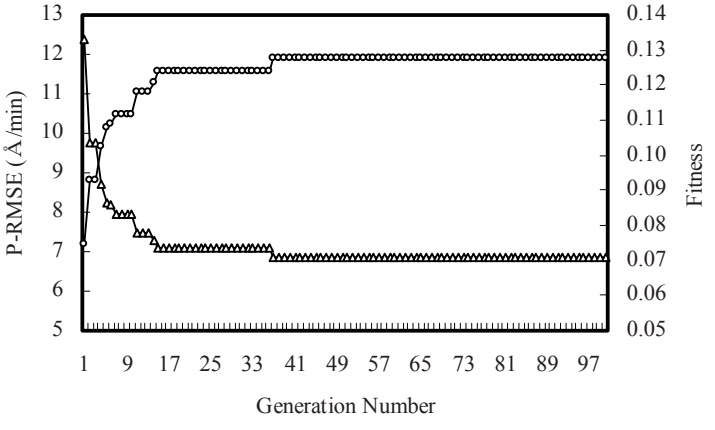


Fig. 1. Performance of GA-GRNN as a function of generation number

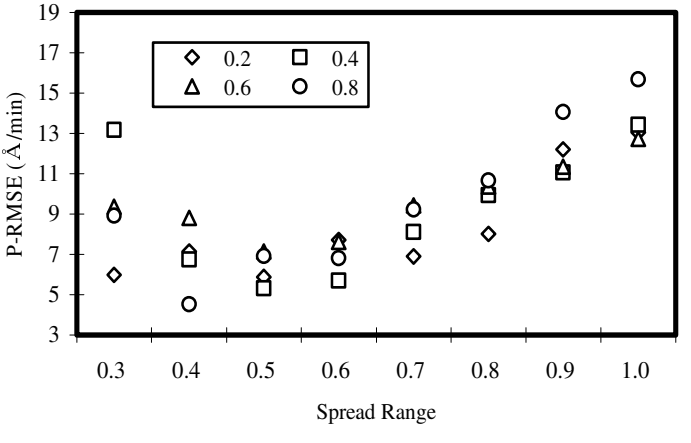


Fig. 2. Prediction performance of GA-GRNN as a function of spread range and balancing factor

In Fig. 1, the symbols of triangle and circle represent the P-RMSE and fitness, respectively. Each P-RMSE determined at each spread corresponds to the smallest pre diction error selected among 100 candidates. From Fig. 1, one best model occurs at a generation number of 37 and the corresponding P-RMSE is about 6.84 Å/min. The corresponding fitness is about 0.128. In this way, GRNN models were constructed with the variations in other spread ranges and θ . The resulting variations in the P-RMSE are plotted in Fig. 2. As shown in Fig. 2, the P-RMSE initially decreases and then increases with increasing the spread range. Another observation noticeable is that the θ effect is quite complex depending on the spread range. Table 2 contains the performances of best GA-GRNN models determined for the deposition rate and charge density. Table 2 reveals that one optimized model of the deposition rate is obtained at the spread range and θ of 0.4 and 0.8, respectively. The corresponding P-RMSE is about 4.53 Å /min. For the charge density, the optimized model is obtained at the spread range and θ of 0.5 and 0.8, respectively. The corresponding P-RMSE is about $0.677\times10^{12}/cm^3$.

Table 2. Performance of optimized GA-GRNN models with variations in the range and θ

		Deposition Rate (Å /min)			Charge Density ($10^{12}/cm^3$)	
Range	θ	T-RMSE	P-RMSE	θ	T-RMSE	P-RMSE
0.3	0.2	0.04	5.99	0.4	0.001	0.747
0.4	0.8	0.29	4.53	0.6	0.036	0.702
0.5	0.4	1.31	5.31	0.8	0.048	0.677
0.6	0.4	3.00	5.70	0.2	0.281	0.860
0.7	0.2	5.06	6.90	0.8	0.215	0.977
0.8	0.2	8.31	8.02	0.6	0.904	0.985
0.9	0.2	9.74	11.07	0.2	0.736	0.913

3.2 Comparison with Conventional GRNN Model

For comparison, GRNN model was constructed in conventional way. In other words, the spread was incrementally increased from 0.3 to 1.4 with an increment of 0.1. The GRNN performance is shown in Fig. 3 as a function of spread. In Fig. 3, both P-RMSE and T-RMSE are represented in the circle and triangle, respectively. As shown in Fig. 3, the P-RMSE variation is considerably different from that for the best models shown in Fig. 2. In Fig. 3, the best P-RMSE is obtained at 1.2 and is about 15.7 Å/min. Compared to this error, the smallest P-RMSE (4.5 Å/min) for the GA-GRNN model demonstrates an improvement of about 71.3%. For the T-RMSE, the improvement is more drastic. In other words, the T-RMSE (0.29 Å/min) for GA-GRNN model yields an improvement of about 97% compared to that (9.4 Å/min) for conventional GRNN model. In the same way, GRNN model for the charge density was constructed in the same range of spreads. In the range between 0.3 and 0.6, the P-RMSEs for the GRNN models were comparable and they were about 1.74 Å/min. Compared to this model, the GA-GRNN model therefore demonstrates an improvement of about 61%. In consequence, the GA-GRNN models significantly improved the prediction performances of conventional GRNN models for the two film characteristics.

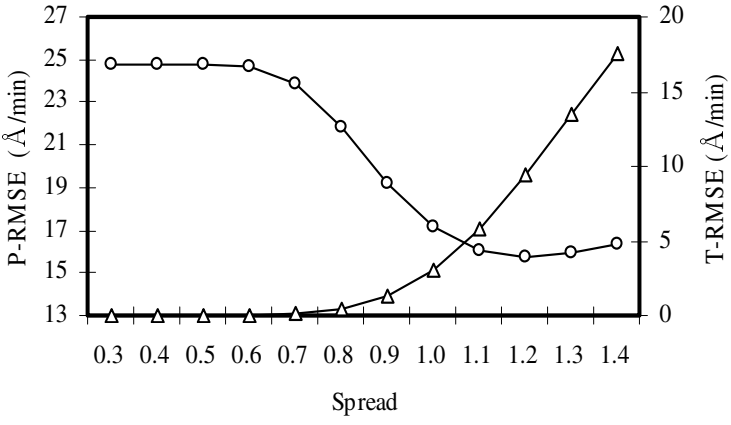


Fig. 3. Performance of conventional GRNN model as a function of spread

3.3 Comparison with Statistical Regression Model

The GA-GRNN model is further compared to statistical regression models. A typical formula for a regression model is represented as

$$y = \beta_o + \sum_{i=1}^{k} \beta_i x_i + \sum_{i}^{k} \beta_{ii} x_i^2 + \sum_{i} \sum_{j} \beta_{ij} x_i x_j \tag{5}$$

where y is one of the deposition characteristics, β_i and β_{ij} are the regression coefficients, and x_i is the regressor variable corresponding to a process parameter. An index k denotes the total number of the process parameters, here six. Four types of regression models were constructed. Type I model consists of only single regressor variables in (5). Type III and IV models are the extended versions of type I. In other words, type III and IV models correspond to type I model with only either interaction or quadratic terms included, respectively. The remaining type II model is exactly identical to the present form of (5). The performances of regression models calculated with the test data are shown in Table 3. For the deposition rate, Table 3 reveals that both prediction errors for type I and III models are identical. Of them, type III model is selected for comparison since it attained a much smaller T-RMSE compared to that for type I model. This is true for the charge density model. In other words, the same type III model is determined to be the best model. Compared to the P-RMSEs for the optimized type III regression models, the GA-GRNN models demonstrate an improvement of about 55% and 65% for the deposition rate and charge density, respectively. For the T-RMSE, the improvements were about 95% and 96.5%. Similar to earlier comparison with conventional GRNN models, the GA-GRNN models exhibited much improved training and prediction performance over statistical regression models.

Table 3. Performance of statistical regression models

	Deposition Rate (Å /min)		Charge Density (10^{12}/cm^3)	
Types	T-RMSE	P-RMSE	T-RMSE	P-RMSE
I	12.0	10.0	1.408	1.942
II	6.2	145.8	0.756	23.991
III	6.2	10.0	0.759	1.942
IV	12.0	36.9	1.407	1.791

4 Conclusion

A technique to improve the prediction performance of GRNN was presented. This was accomplished by a multi-parameterization of training factors and GA optimization. Model performance was examined in terms of the training and prediction errors. The technique was evaluated with the deposition rate and charge density data, collected during a plasma enhanced chemical vapor deposition of silicon nitride films. For comparison, statistical regression and GRNN models were constructed. Comparisons revealed that GA-GRNN models yielded significantly improved prediction and training errors. The presented technique is general in that it can be applied to any plasma or complex process data.

Acknowledgements

This study was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment). This work was supported by the ERC program of MOST/KOSEF (Next-Generation Power Technology Center).

References

1. Rummelhart, D.E., McClelland, J.L.: Parallel Distributed Processing. M.I.T. Press, Cambridge (1986)
2. Kim, B, Kwon K.: Qualitative Modeling of Silica Plasma Etching Using Neural Network. J. Appl. Phys. 94(1) (2003) 76-82
3. Kim, B, Han, S.S., Kim, T.S., Kim, B.S., Shim I.J.: Shim, Modeling Refraction Characteristics of Silicon Nitride Film Deposited in a SiH4/NH3/N2 Plasma Using Neural Network. IEEE Trans Plasma Sci. 31(3) (2003) 317-323
4. Kim, B., Park S.: An Optimal Neural Network Plasma Model: a Case Study. Chmommetr. Intell. Lab. Syst. 56(1) (2001) 39-50
5. Specht, D.F.: A General Regression Neural Network. IEEE Trans. Neural Networks 2(5) (1991) 568-576
6. Kim, B., Park, S.: Modeling of Process Plasma Using a Radial Basis Function Network: A Case Study. Trans. Contr. Autom. Syst. Eng. 2(4) (2000) 268-273

7. Goldberg, D.E.: Genetic Algorithms in Search, Optimization & Machine Learning. Addison Wesley, Reading MA (1989)
8. Han, S.S., Cai, L., May, G.S., Rohatgi, A.: Modeling the Growth of PECVD Silicon Nitride Films for Solar Cell Applications Using Neural Networks. IEEE Trans. Semicond. Manufact. 9(3) (1996) 303-311
9. Geisler, J.P., Lee, C.S.G., May G.S.: Neruofuzzy Modeling of Chemical Vapor Deposition Process. IEEE Semicond. Manufact. 13(1) (2000) 46-60
10. Montgomery, D.C.: Design and Analysis of Experiments. John Wiley & Sons, Singapore (1991)

Prediction of Radio Frequency Impedance Matching in Plasma Equipment Using Neural Network

Byungwhan Kim[1], Donghwan Kim[2], and Seung Soo Han[3]

[1] Department of Electronic Engineering, Sejong University, Seoul, Korea
kbwhan@sejong.ac.kr
[2] School of Mechanical Design & Automation Engineering,
Seoul National University of Technology, Seoul, Korea
dhkim@snut.ac.kr
[3] Department of Information Engineering, Myongji University, Yongin, Korea
shan@mju.ac.kr

Abstract. Optimizing a plasma impedance match process requires construction of prediction model. In this study, generalized regression neural network (GRNN) combined with genetic algorithm (GA) was used to build a match prediction model. A real-time match monitor system was used to collect steady match positions according to a statistical experimental design. GA-GRNN models were compared to GRNN and statistical regression models. Compared to GRNN models, GA-GRNN models demonstrated improved predictions of about 81% and 77% for the impedance and phase positions, respectively. With respect to statistical regression models, GA-GRNN models yielded an improvement of about 80% and 78%, respectively. Moreover, for either model type, the improvements for the training errors were more than about 90% for both positions.

1 Introduction

Plasma plays a crucial role in depositing or etching fine patterns in manufacturing integrated circuits. As an important component in plasma etch equipment, a radio frequency (RF) match network plays a role of balancing impedances between the equipment plasma and RF generator. The match process is typically conducted by controlling two mechanical motors. Their steady positions can vary considerably with the variations in process parameters or in circuit components comprising the match network. To achieve a fast matching and less reflected power, a technique to model the relationship between the process or match network parameters and match positions is demanded.

In this study, a RF match process was modeled using generalized regression neural network (GRNN) [1]. GRNN has been applied to build prediction models of various plasma processes [2-5]. A real-time match monitor system was utilized to collected matching data [6]. For a systematic modeling, the match process was characterized by a statistical experimental design. The presented technique was evaluated with the match data collected from an inductively coupled plasma. The match variations modeled are the impedance and phase positions. The model performance was optimized

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1028 – 1035, 2006.

by applying genetic algorithm (GA) [7] to search for multi-parameterized GRNN training factors. For comparison, conventional GRNN and statistical regression models were also constructed.

2 Match Data and GRNN

2.1 Match Monitor System

Schematic diagram of a real-time monitor system is shown in Fig. 1 [6]. A low pressure planar plasma is generated by an array of six magnetic cores, mounted on the top of vacuum chamber Al anodized. Each ferrite core was equally spaced by 40 mm and has three turns of coils. The rf power to the coils was delivered by an RF generator (RF 30S). Due to the time-varying magnetic fields between poles of opposite polarity, free electrons in the gas are accelerated in the direction parallel to the fields and, subsequently, the plasma is produced. A ceramic plate was employed to separate the plasma source from the chamber, with an area of 430 mm by 430 mm and a height of 290 mm. An electrostatic chuck has a diameter of 200 mm and is driven by another RF power generator (RF 5S) at the same 13.56 MHz.

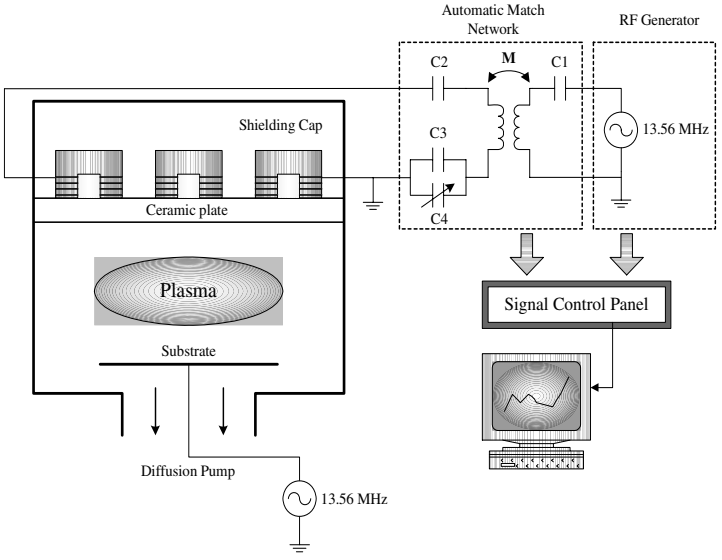


Fig. 1. Schematic of radio frequency impedance matching monitor system

As depicted in Fig. 1, the match network consists of three fixed capacitors (C_1, C_2, and C_3), a variable vacuum capacitor (C_4), and a variable transformer denoted as a mutual inductance (M). The values for those fixed capacitors are 94.0 pF, 62.6 pF and 31.3 pF for C_1, C_2 and C_3, respectively. The variable transformer comprises a primary coil with four turns and a secondary coil with three turns. Its action is achieved by varying the coupling of mutual inductance between the two coils. As the primary coil is rotated inside the secondary by an *impedance motor*, its inductance is varied from a

maximum (140Ω) for zero coupling and to a minimum (100Ω) for maximum coupling. Since the inductive reactance of the primary coil with more than four turns usually exceeds the output impedance of generator, C_1 is connected in series with the coil to balance it. The coil is rotated until the secondary circuit load coupled via M leads to a 50Ω impedance at the rf source. At the same time, the secondary coil is rotated using another *phase motor* to make the impedance purely resistive.

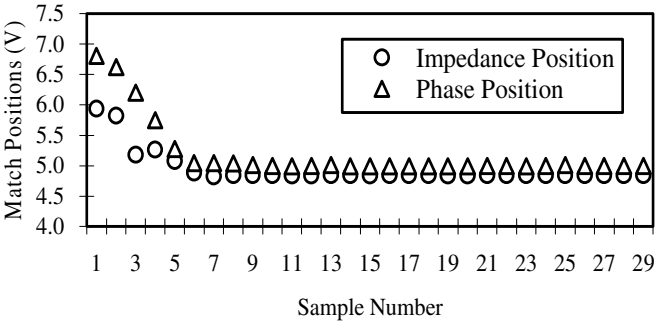


Fig. 2. Matching characteristics at 1000 W source power, 75 sccm O_2, and 15 mTorr

To collect real-time matching data under various plasma conditions, a multifunction board (PCI-20428W-1) equipped to a PC was interfaced to a signal control panel (EASYDAS-5BP), where I/Os are communicated with board's programming and data registers via a I^3 Bus™. Upon running a flow diagram coded using Visual Designer, monitored variables are then displayed on the PC. Using the match monitor system, steady data for the two matching positions, impedance and phase, were collected. The positions in each experiment were initially set at 5.94 and 6.80 volts, for impedance and phase motors, respectively. As an illustration, the position data are shown in Fig. 2. The data were collected during the matching at 1000 W source power, 75 sccm O_2, and 15 mTorr pressure. As shown in Fig. 2, the positions appear to reach their steady values after experiencing a transient period.

Table 1. Process parameters and ranges

Parameters	Ranges	Units
RF Source Power	500-1000	W
O_2 Flow Rate	50-100	sccm
Pressure	10-20	mTorr

To construct a prediction model, the experiments were conducted according to 2^3 full factorial experiment [8]. The three process parameters and their ranges are shown in Table 1. The resulting 8 experiments including one center point were used to train GRNN. The appropriateness of the trained model was tested with another 8 experiments, not belonging to the training data.

2.2 Generalized Regression Neural Network

GRNN was used to construct a prediction model of impedance matching. The GRNN consists of four layers, including the input layer, pattern layer, summation layer, and output layer. The first layer is fully connected to the second, pattern layer, where each unit represents a training input pattern and its output is a measure of the distance of the input from the stored patterns. Each pattern layer unit is connected to the two neurons in the summation layer: S-summation neuron and D-summation neuron. The S-summation neuron computes the sum of the weighted outputs of the pattern layer while the D-summation neuron calculates the unweighted outputs of the pattern neurons. In the output layer, a prediction to an unknown input vector x is computed as

$$\hat{y}_i(x) = \frac{\sum_{i=1}^{n} y_i \exp[-D(x,x_i)]}{\sum_{i=1}^{n} \exp[-D(x,x_i)]} \tag{1}$$

where n indicates the number of training patterns and the Gaussian D function in (1) is defined as

$$D(x,x_i) = \sum_{j=1}^{p} \left(\frac{x_j - x_{i_j}}{\zeta}\right)^2 \tag{2}$$

where p indicates the number of elements of an input vector. The x_j and x_{i_j} represent the jth element of x and x_i, respectively. The ζ is generally referred to as the spread, whose optimal value is often determined experimentally.

3 Results

3.1 GA-Optimization

The impedance and phase positions were modeled separately. In other words, the GRNN structure consisted of three inputs and one output. Rather than one singe spread, multi-parameterized spreads were adopted for the D function defined in (2). Their complex effects on model performance were optimized by applying GA. Model performance was evaluated in terms of training and prediction error, represented as T-RMSE and P-RMSE, respectively. In GA optimization, a random generator was employed to assign random values to the spreads for all gaussian functions in the pattern layer. The random values were generated within a spread range given between 0.3 and 0.9. As an illustration, an impedance match position model was constructed at the spread range [0, 1.0]. The initial population was set at 100 potential solutions. Each chromosome was coded in the real value. The roulette wheel selection was employed for the reproduction. The probabilities of crossover and mutation were set to 0.95 and 0.05, respectively. A particular input setting generated by the GA meets a given fitness function expressed as

$$F = \frac{1}{1 + T - RMSE} \tag{3}$$

The search process was completed as the generation number reaches a predefined value of 100. The resulting two errors are plotted in Fig. 3 as a function of the generation number.

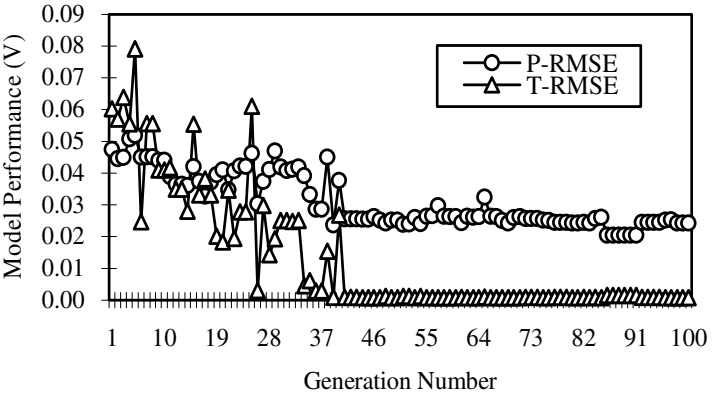


Fig. 3. Performance of GA-GRNN model of impedance position

In Fig. 3, the symbols of triangle and circle represent the T-RMSE and P-RMSE, respectively. The P-RMSE at each spread represents the smallest prediction error determined among 100 candidates. The T-RMSE is the corresponding training error. From Fig. 3, the smallest P-RMSE starts to occur at 86th generation and is equal to about 0.020 V. The corresponding T-RMSE is about 0.00138 V. In this way, GA-GRNN models were constructed with the variations in other spread ranges. This was conducted for either position and the results are shown in Table 2. As shown in Table 2, the smallest P-RMSE for the impedance position model is obtained at the spared range of 0.3. The corresponding P-RMSE is about 0.0103 V. For the phase position model, meanwhile, the smallest P-RMSE obtained at the spread range of 0.6 is about 0.0235 V.

Table 2. Performance of optimized GA-GRNN models with variations in the spread range

	Impedance Position (V)		Phase Position (V)	
Range	T-RMSE	P-RMSE	T-RMSE	P-RMSE
0.3	2.05E-05	0.0103	0.00150	0.0255
0.4	0.00088	0.0156	0.00289	0.0249
0.5	0.00030	0.0136	0.00013	0.0294
0.6	0.00967	0.0185	0.00233	0.0235
0.7	0.01064	0.0214	0.00518	0.0299
0.8	5.42E-05	0.0167	0.00230	0.0377
0.9	0.00499	0.0166	0.00079	0.0609

3.2 Comparison with Conventional GRNN Model

For comparison, GRNN model was constructed in conventional way. In other words, the spread was incrementally increased from 0.3 to 0.9 with an increment of 0.1. The GRNN performance is shown in Table 3 as a function of spread. As shown in Table 3, the T-RMSE and P-RMSE are seen to increase and decrease with increasing the spread value.

Table 3. Performance of optimized GRNN models with variations in the spread

	Impedance Position (V)		Phase Position (V)	
Spread	T-RMSE	P-RMSE	T-RMSE	P-RMSE
0.3	1.54E-11	0.0921	4.09E-11	0.2187
0.4	3.90E-07	0.0796	1.03E-06	0.1863
0.5	4.72E-05	0.0673	0.00012	0.1541
0.6	0.00071	0.0598	0.00185	0.1342
0.7	0.00385	0.0562	0.00997	0.1247
0.8	0.01153	0.0553	0.02977	0.1222
0.9	0.02384	0.0559	0.06139	0.1244

For the impedance position model, as shown in Table 3, the smallest P-RMSE is obtained at the spread of 0.8 and is about 0.0553 V. Compared to this error, the optimized P-RMSE for the GA-GRNN model yields an improvement of about 81%. Meanwhile, the smallest P-RMSE for the phase position obtained at the same 0.8 is about 0.1222 V as shown in Table 3. In this case, the improvement of the GA-GRNN model over the GRNN model is about 80%. From the standpoint of T-RMSE, the GA-GRNN models demonstrate an improvement of about 99 and 92% for the impedance and phase positions, respectively. These comparisons reveal that GA-GRNN model can significantly improve conventional GRNN model performance.

3.3 Comparison with Statistical Regression Model

The GA-GRNN models are further compared to statistical regression models. A typical formula for a regression model is represented as

$$y = \beta_o + \sum_{i=1}^{k} \beta_i x_i + \sum_{i}^{k} \beta_{ii} x_i^2 + \sum_{i} \sum_{j} \beta_{ij} x_i x_j \quad (4)$$

where y is the deposition rate, β_i and β_{ij} are the regression coefficients, and x_i is the regressor variable corresponding to a process parameter. An index k denotes the total number of the process parameters, here three. Four types of regression models were constructed. Type I model consists of only single regressor variables in (4). Type III and IV models are the extended versions of type I. In other words, type III and IV models correspond to type I model with only either interaction or quadratic

Table 4. Performance of statistical regression models

	Impedance Position (V)		Phase Position (V)	
Types	T-RMSE	P-RMSE	T-RMSE	P-RMSE
I	0.0808	0.0464	0.2177	0.1168
II	0.0391	0.0982	0.1174	0.2941
III	0.0391	0.0464	0.1187	0.1168
IV	0.0808	0.1737	0.2169	0.8520

terms included, respectively. The remaining type II model is exactly identical to the present form of (4). The performances of regression models calculated are shown in Table 4.

For the impedance position data, Table 4 reveals that both P-RMSEs for type I and III models are identical. Of them, type III model is selected for comparison since it attained a much smaller T-RMSE. This is true for the phase position model. In other words, the same type III model is determined to be the best model. Compared to the P-RMSEs for the optimized type III regression models, the GA-GRNN models demonstrate an improvement of about 77% and 78% for the impedance and phase positions, respectively. From the standpoint of T-RMSE, the improvements were about 99% and 98%, respectively. Similar to earlier comparisons with conventional GRNN models, the GA-GRNN models exhibited much improved training and prediction performance over statistical regression models.

4 Conclusion

A prediction model of radio frequency plasma impedance matching was constructed by using GRNN. The GRNN prediction performance was optimized by using GA to search for one optimized set of training factors. A match monitor system was used to collect two electrical position data for mechanical match motors. The comparisons revealed that GA-GRNN models were superior to conventional GRNN or statistical regression models in both training and prediction accuracies. This indicates that GA-GRNN models can be used to construct accurate match models in optimizing match characteristics such as the matching time or reflected power. Moreover, match models can be utilized to detect plasma faults on a run-to-run basis as in-situ diagnostic data such as optical emission spectroscopy is used as a network input pattern. The presented technique can also be extended to optimize match network itself by building a model between the match network circuits and positions.

Acknowledgements

This study was supported by the University IT Research Center Project, and partly bythe ERC program of MOST/KOSEF (Next-Generation Power Technology Center).

References

1. Specht, D.F.: A General Regression Neural Network. IEEE Trans. Neural Network 2(5) (1991) 568-576
2. Kim, B., Kim, S., Lee, B.T.: Modeling SiC Surface Roughness Using Neural Network and Atomic Force Microscopy. J. Vac. Sci. Technol. B 22(5) (2004) 2467-2472
3. Kim, B., Lee B.T.: Prediction of SiC Etching in a NF_3/CH_4 Plasma Using Neural Network. J. Vac. Sci. Technol. A 22(6) (2004) 2517-2522
4. Kim, B, Lee, D., Kwon K.H.: Prediction of Etch Microtrenching Using a Neural Network. J. Appl. Phys. 96(7) (2004) 3612-3616
5. Kim, B, Park, K, Lee, D.: Use of Neural Network to Model a Deposition Rate of PECVD-Silicon Nitride Films. Plasma Sources Sci. Technol. 14 (2005) 83-88
6. Kim, B., Lee, C.: Monitoring Plasma Impedance Match Characteristics in a Multipole Inductively Coupled Plasma for Process Control. J. Vac. Sci. Technol. A 18(1) (2000) 58-62
7. Goldberg, D.E.: Genetic Algorithms in Search, Optimization & Machine Learning. Addison Wesley, Reading MA (1989)
8. Montgomery, D.C.: Design and Analysis of Experiments. John Wiley & Sons, Singapore (1991)

Recognition of Plasma-Induced X-Ray Photoelectron Spectroscopy Fault Pattern Using Wavelet and Neural Network

Byungwhan Kim[1], Sooyoun Kim[1], and Sang Jeen Hong[2]

[1] Department of Electronic Engineering, Sejong University, Seoul, Korea
kbwhan@sejong.ac.kr
[2] Department of Electronic Engineering, Myongji University, Yongin, Korea
samhong@mju.ac.kr

Abstract. To improve device yield and throughput, faults in plasma processing equipment should be quickly and accurately diagnosed. Despite many useful information of ex-situ sensor measurements, their applications to recognize plasma faults have not been investigated. In this study, a new technique to identify fault causes by recognizing X-ray photoelectron spectroscopy (XPS) using neural network and continuous wavelet transformation (CWT). The presented technique was evaluated with the plasma etch data. A total of 17 experiments were conducted for model construction. Model performance was investigated from the perspectives of training error, testing error, and recognition accuracy with respect to various thresholds. CWT-based BPNN models demonstrated a higher prediction accuracy of about 26%. Their advantages over pure XPS-based models were conspicuous in all three measures at small networks.

1 Introduction

Plasma is a key means to deposit thin films or to etch fine patterns in manufacturing integrated circuits. To maintain process quality, equipment throughput, and device yield, plasma processes should be tightly monitored and diagnosed. The diagnosis is conducted in two fashions. One is to detect a new fault. The other is to recognize a previously encountered fault. For fault detection, in-situ sensors such as optical emission spectroscopy [1], radio frequency (RF) impedance sensor [2], or RF matching monitor [3] has been utilized to monitor plasma states. The OES [4-5] or matching [6] data were also combined with neural network to predict film characteristics. During the in-line diagnosis, ex-situ wafer measurements such as the etch rate, meanwhile, have been used to identify process faults [7]. Recognition of previous fault patterns is of importance since it can save time and efforts needed for fault detection as the current fault pertains to one of the recognized faults. Moreover, the recognition process facilitates detecting a new, unrecognized fault. During the on-line diagnosis, in-situ sensor data were used to recognize plasma fault by constructing a neural network time-series model [4]. However, fault recognition with conventional film measurements is limited in that their number is not large enough to form a fault symptom.

In this study, a new technique to construct a recognizer of plasma faults is presented. This is accomplished by combining neural network, wavelet, and X-ray

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1036 – 1042, 2006.

photoelectron spectroscopy (XPS). As the fault recognizer, the backpropagation neural network [8] (BPNN) was adopted. The use of XPS as a fault symptom is advantageous in that it provides detailed variations of chemical states on film surface. Moreover, XPS data consist of a large number of kinetic energy levels, thereby facilitating training neural network. Filtering XPS data through wavelet may improve the network prediction by providing less noisy data to neural network. The proposed technique was evaluated with the data collected during the etching of silicon carbide (SiC) in a NF_4-CH_4 inductively coupled plasma [9].

2 Experimental Data

A schematic of an ICP etch system is shown in Fig. 1. A plasma is generated inside a chamber and isolated from planar-coupled coils by a dielectric quartz window. The cylindrical chamber has a radius and height of 80 mm and 40 mm, respectively. Prior to feeding etching gases, the chamber was evacuated using turbo (TUROVAC 3430MC) and rotary (Edward High Vacuum E2M40) pumps, thereby maintaining the base pressure at about 10^{-6} Torr. Flow rates of gases were precisely controlled through mass flow controllers, and the process pressure, measured by a baratron gauge, was controlled with a throttle valve. A coolant was fed to the chuck holder to minimize damage to the equipment from a surge in temperature during the etching.

Test patterns were fabricated on n-type, 2 inch 4H-SiC wafers of <0001> orientation. The "4" and "H" indicate a stacking sequence and a hexagonal type of crystal structure, respectively. Using a Cr negative mask, photo resist (PR) patterns were first

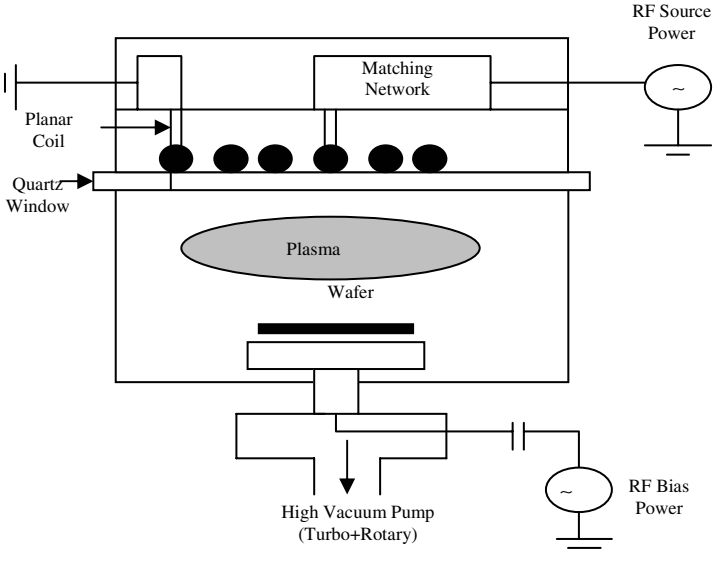


Fig. 1. Schematic of plasma etch system

formed. A magnetron sputtering method was used to subsequently deposit Ni film of about 0.18 μm on the patterned photoresist. By removing the photoresist with acetone, Ni mask layer was formed. The SiC films were then etched in NF_3/CH_4 plasma. In all experiments, the etching was conducted for the same 5 min. The process parameters examined include the rf source power, bias power, pressure, and NF_3 percentage. The percentage was defined as a ratio of the NF_3 flow rate to the total flow rate, i.e., the sum of NF_3 and CH_4 flow rates. The total flow rate was set to 30 sccm in every experiment. Chemical states on the etched surface were measured by XPS. The XPS measurement was performed by using a EscaLab 250. Over the entire range of kinetic energy, 686.1-1486.6 eV, the XPS measurement was collected with an increment of 0.9996 eV. A total of 801 values could therefore be collected. To construct a neural network recognizer, plasma faults were simulated with the variations in the process parameters. For each process parameter, four faults were generated. The resulting 16 experiments plus one experiment corresponding to the center point in the design were used to train BPNN. It should be noted that the center experiment represents a normal condition. The trained BPNN was tested with 16 generated faults. The input of BPNN consisted of 801 XPS elements and the output was composed of four fault causes corresponding to one of the four process parameters. Fault severity was varied from -1.0 to 1.0 depending on the magnitude of parameter variations. For example, the output fault pattern for the center point (i.e., normal condition) was expressed as (0, 0, 0, 0).

3 Results

3.1 CWT of XPS Data

As an illustration, XPS data are shown in Fig. 2 as a function of source power. As shown in Fig. 2, the source power was varied from 700 to 900 W with an increment of 50 W. As shown in Fig. 2, the XPS patterns are unique to each power variation. This feature is advantageous to detect a plasma fault. The XPS data were then transformed by applying the CWT. The results are shown in Fig. 3.

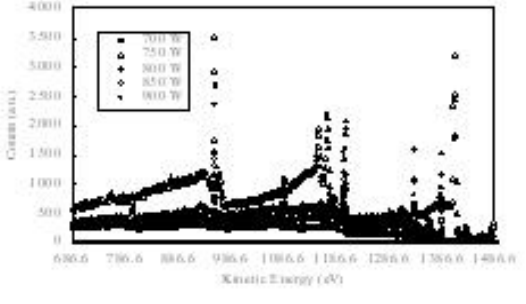


Fig. 2. XPS data collected with the variations in source power

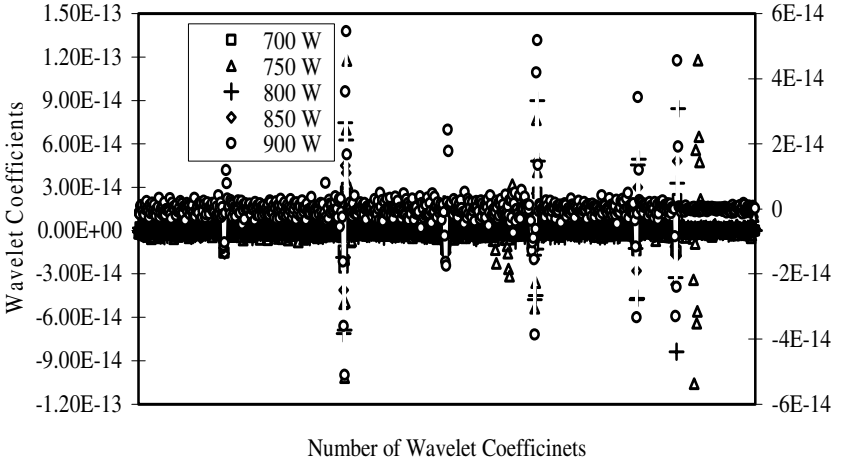


Fig. 3. Patterns of CWT of XPS shown in Fig. 2 as a function of source power

3.2 Recognition of XPS Fault Patterns

First, BPNN was trained with 17 XPS patterns and the trained BPNN was subsequently tested with 16 experiments. This was evaluated with the variations in the number of hidden neurons (NHN) from 3 to 9. The other training tolerance, initial weight distribution, gradient of activation functions were set at 0.1, ±1.0, and 1.0, respectively. The physical meanings of each training factor were described in detail [10]. The training and testing errors for the BPNN models constructed with XPS data are shown in Table 1. The errors were measured by the typical root mean-squared error. As shown in Table 1, the testing error is smallest at 8 or 9 hidden neurons, which is numerically equal to about 0.038. Then, the recognition accuracy was measured by comparing model prediction (MP) and actual fault severity (FS) with respect to the fault threshold. This is expressed as

$$|\mathrm{MP} - \mathrm{FS}| < \text{threshold} \tag{1}$$

Table 1. Training and testing errors of BPNN models constructed with XPS patterns

NHN	Training Error	Testing Error
3	0.357	0.367
4	0.258	0.264
5	0.262	0.269
6	0.187	0.193
7	0.038	0.039
8	0.038	0.038
9	0.038	0.038

Table 2. Recognition accuracy of faults with respect to fault threshold

NHN	0.5	0.4	0.3	0.2	0.1
3	3	2	2	2	2
4	9	9	9	8	3
5	8	7	6	6	4
6	12	12	11	11	11
7	16	16	16	16	16
8	16	16	16	16	16
9	16	16	16	16	15

The fault severity was varied from 0.1 to 0.5 with an increment of 0.1. The resulting recognition accuracy is shown in Table 2. As shown in Table 2, a prefect recognition is obtained at NHN of 7 or 8 at all thresholds. Of the two models, the latter model is determined to be the optimal recognizer due to its smaller testing error than the former model.

3.3 Recognition of CWT-XPS Fault Patterns

Next, BPNN was applied to recognize CWT-XPS data. The performances of constructed models are shown in Table 3. As shown in Table 3, the smallest testing error is achieved at a NHN of 7, and the corresponding error is about 0.028. This shows an improvement of about 26.3% compared to that for the optimized model with pure XPS data. This clearly demonstrates the advantage of CWT in recognizing complex XPS data. Meanwhile, the recognition accuracy of models constructed with CWT-XPS patterns is shown in Table 4. As shown in Table 4, almost perfect fault recognition is obtained at several NHNs ranging between 6 and 9. Therefore, it is observed that at relatively large NHNs the recognition accuracies of both XPS and CWT-XPS models are comparable.

Another difference can be identified by calculating the total recognition accuracy (TRA). The TRA is simply the sum of the recognition accuracies determined at all thresholds for a given NHN. The TRAs computed for the recognition accuracies shown in Table 2 and 4 are compared in the bar graph in Fig. 4.

Table 3. Training and testing errors of BPNN models constructed with CWT-XPS patterns

NHN	Training Error	Testing Error
3	0.242	0.249
4	0.169	0.169
5	0.078	0.079
6	0.038	0.039
7	0.038	0.028
8	0.038	0.037
9	0.038	0.039

Table 4. Recognition accuracy of faults with respect to fault threshold

NHN	0.5	0.4	0.3	0.2	0.1
3	8	8	8	7	4
4	13	12	11	10	8
5	16	16	16	15	12
6	16	16	16	16	15
7	16	16	16	16	15
8	16	16	16	16	16
9	16	16	16	16	15

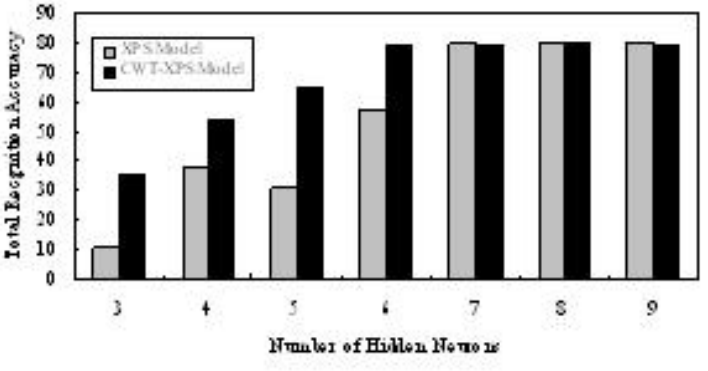


Fig. 4. Comparison of TRAs as a function of NHN

For the NHNs between 7 and 9, both models exhibit comparable TRAs regardless of NHNs. However, at relatively smaller NHNs less than 6, the CWT-XPS models yield a significant improvement in TRA over XPS models. Moreover, as certified from Table 1 and 3, the CWT-XPS model is superior to XPS model in the corresponding training and prediction accuracies. This indicates that preprocessing XPS with CWT is effective in constructing a BPNN recognizer particularly at smaller NHNs.

4 Conclusion

A new technique for recognizing complex ex-situ measurements of film surface was presented. This was accomplished by combining neural network, continuous wavelet transformation (CWT), and X-ray photoelectron spectroscopy (XPS). The wavelet was used to preprocess XPS patterns. The experimental XPS patterns were collected during the plasma etching of silicon carbide films. Both XPS and CWT-XPS models were compared in terms of training error, prediction error, and recognition accuracy. Depending on the number of hidden neurons, the contributions of XPS to the network recognition performance were quite different. Preprocessing XPS with

CWT considerably improved the model prediction accuracy. The advantages of CWT preprocessing became pronounced at smaller networks.

Acknowledgements

This study was supported by "System IC 2010" project of Korea Ministry of Commerce, Industry, and Energy, and partly by the University IT Research Center Project.

References

1. Stevenson, J.O., Ward, P.P., Smith, M.L., Markle R.J.: A Plasma Process Monitor/Control System. Surf. Interf. Anal. 26 (1998) 124-133
2. Bose, F., Patrick, R., Baltes H.: Measurement of Discharge Impedance for Dry Etch Process Control. SPIE 2336 (1994) 101-110
3. Kim, B., Lee, C.: Monitoring Plasma Impedance Match Characteristics in a Multipole Inductively Coupled Plasma for Process Control. J. Vac. Sci. Technol. A 18(1) (2000) 58-62
4. Hong, S.J., May G.S., Park D.C.: Neural Network Modeling of Reactive Ion Etching Using Optical Emission Spectroscopy Data. IEEE Trans. Semicond. Manufact. 16(4) (2003) 598-608
5. Kim, B., Kim S.: Partial Diagnostic Data to Plasma Etch Modeling Using Neural Network. Microelectron. Eng. 75(4) (2004) 397-404
6. Kim, B., Kim, S.: Diagnosis of Plasma Processing Equipment Using Neural Metwork Recognition of Wavelet-Filtered Impedance Matching. Microelectron. Eng. 82(1) (2005) 44-52
7. Kim, B., May G.S.: Real-Time Diagnosis of Semiconductor Manufacturing Equipment Using a Hybrid Neural Network Expert System. IEEE Trans. Comp. Packag. Manufact. Technol. 20(1) (1997) 39-47
8. Rummelhart, D.E., McClelland, J.L.: Parallel Distributed Processing. M.I.T. Press, Cambridge (1986)
9. Kim, B., Lee, S.Y., Lee B.T.: Etching of 4-H SiC in a NF_3/CH_4 Incudctively Coupled Plasma. J. Vac. Sci. Technol. B 21(6) (2003) 2455-2460
10. Kim, B., Park S.: An Optimal Neural Network Plasma Model: a Case Study. Chemometr. Intell. Lab. Syst. 56(1) (2001) 39-50

Polynomial Neural Network Modeling of Reactive Ion Etching Process Using GMDH Method

Seung-Soo Han[1] and Sang Jeen Hong[2]

[1] Department of Information Engineering & NPT Center, Myongji University, Yongin, Gyeonggi, Korea
shan@mju.ac.kr
[2] Department of Electronic Engineering & Nano-Bio Research Center, Myongji University Yongin, Gyeonggi, Korea
samhong@mju.ac.kr

Abstract. The construction of models for prediction and control of initially unknown, potentially nonlinear system is a difficult, fundamental problem in machine learning and engineering control. Since the neural network as a tool for machine learning was introduced, significant progress has been made on data handling and learning algorithms. Currently, the most popular learning algorithm in neural network training is feed forward error back-propagation (FFEBP) algorithm. Aside from the success of the FFEBP algorithm, a polynomial neural networks (PNN) learning has been proposed as a new learning method. The PNN learning is a self-organizing process designed to determine an appropriate set of Ivakhnenko polynomials that allow the activation of many neurons to achieve a desired state of activation that mimics a given set of sampled patterns. These neurons are interconnected in such a way that the knowledge is stored in Ivakhnenko coefficients. In this paper, PNN model has been developed using the nonlinear reactive ion etching (RIE) experimental data utilizing Group Method of Data Handling (GMDH). To characterize the RIE process using PNN, a low-k dielectric polymer benzocyclobutene (BCB) is etched in an SF_6 and O_2 plasma in parallel plate system. Data from 2^4 factorial experimental design to characterize etch process variation with controllable input factors consisting of the two gas flows, RF power and chamber pressure are used to build PNN models of etch rate, uniformity, selectivity and anisotropy. The modeling and prediction performance of PNN is compared with those of FFEBP. The results show that the prediction capability of the PNN models is at least 16.9% better than that of the conventional neural network models.

1 Introduction

In the age of industry, the process engineer must often rely on his/her own experience or on books containing recommended operating parameters for given conditions to operate the highly complex equipments. In order to facilitate more reliable results, robust process modeling methodologies are highly desirable. Accurate process models can enable prediction of process variations over a wide range of input conditions.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1043 – 1052, 2006.

Efficient and tighter process model often offers a competitive edge in today's complex and competitive semiconductor manufacturing.

The neural network based models have recently been shown by several researchers to exhibit superior performance in both accuracy and prediction capability [1-4]. The basic components of a neural network are processing elements (neurons) and weights (connections). The adjustable weights correspond to biological synapses. The weighted inputs to a neuron are accumulated and then passed through an activation function which determines the neuron's response. How to adjust these weights, called learning or training, is the most important in neural network modeling. Since the neural network was introduced, significant process has been made on learning algorithms. Currently, the most popular learning algorithm in neural network training is feed forward error back-propagation (FFEBP) algorithm.

Aside from the success of the FFEBP algorithm, a polynomial neural network (PNN) learning has been proposed as a new learning method. The PNN learning is a self-organizing process designed to determine an appropriate set of Ivakhnenko polynomials that allow the activation of many neurons to achieve a desired state of activation that mimics a given set of sampled patterns. These neurons are intercon nected in such a way that the knowledge is stored in Ivakhnenko coefficients [5-7]. The activation level of a node is determined by a nonlinear activation function with optimal complexity called a Ivakhnenko polynomial. The PNN utilizes the Group Method of Data Handling (GMDH) for data management, which is introduced by Ivakhnenko [5].

In this paper, the PNN model has been developed using the reactive ion etching (RIE) experimental data. For semiconductor manufacturing in this era of sub-micron technology, the importance of reactive ion etching (RIE) cannot be over-emphasized. Due to its capability of producing extremely minute features, RIE is a critical process for the fabrication of both current and next-generation integrated circuits. Much research has been performed to understand this complex process in various ways. Fundamental physical methods, the statistical approaches involving experimental design and response surface modeling have been used to gain a better understanding of process dynamics [8]. More recently, empirical RIE models derived from neural networks have been shown to provide advantages in both accuracy and robustness over traditional statistical approaches [1]. The etching properties of Benzocyclobutene (BCB), a low-k inter-level dielectric material (k=2.65) are determined by the nature and composition of the plasma, which in turn controlled by the deposition variables involved in the RIE process. However, due to the complex nature of particle dynamics within plasma, it is difficult to qualify the exact causal relationship between controllable input factors (such as chamber pressure, RF power, and gas flow rates) and critical output parameters.

To characterize the RIE process, BCB etched under varying conditions were analyzed using 2^4 factorial experimental design [9]. Parameters varied in these experiments included pressure, RF power, CHF_3 flow and oxide flow. 20 data from these experiments were used to train polynomial neural networks to model the following properties: etch rate, selectivity, anisotropy, and non-uniformity.

2 Experiment

The etching tool being used in this study was a Plasma Therm 700 series dual chamber reactive ion etch (RIE). A schematic diagram of this etching tool is shown in Fig. 1.

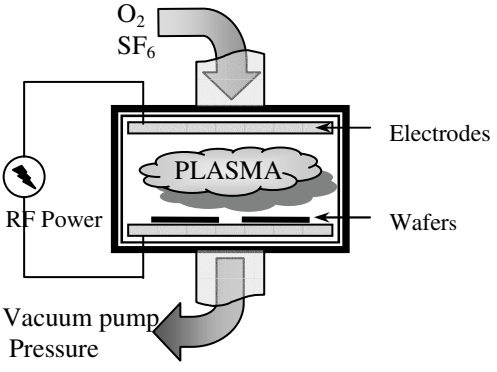


Fig. 1. A schematic of reactive ion etching system operating 13.56 MHz RF power

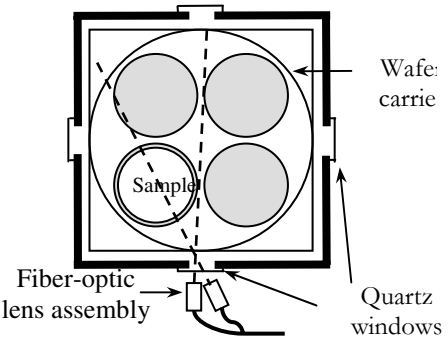


Fig. 2. Optical emission sensors configuration

A Chromex 3020 OES system was used for monitoring the RIE chamber during etching. The OES system consists of three fiber-optic cables connected to three sensors. Two fiber-optic cables were used to collect the plasma emission intensity readings independently through a small quartz window in the front of the RIE chamber. The data collected from a more sensitive sensor (the top sensor) was used for the acquisition of the actual neural network training data, and the other sensor (the bottom sensor) was to check the validity of the top sensor. The top sensor was aimed at the middle of the sample, and the bottom sensor pointed toward the middle of the etch chamber to achieve overall intensity monitoring. Fig. 2. shows etch chamber sensor configuration, and Fig. 3. depicts a typical OES signal in this experiment.

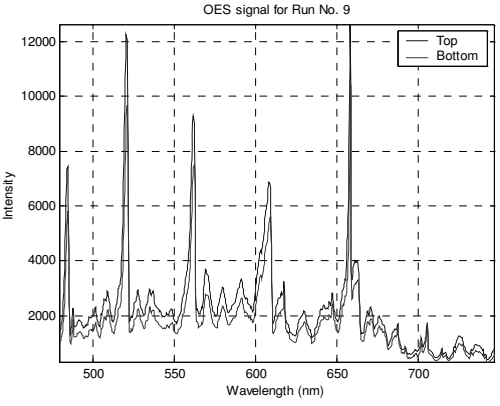


Fig. 3. A typical OES signal in this experiment (upper wavelength)

The material being etched was benzocyclobutene (BCB), marketed under the commercial name of *Cyclotene 3022-46*, with a thick photoresist (AZP 4620) mask. BCB has been proposed as a low dielectric constant (low-*k*) inter-level dielectric (ILD) material. Low-*k* materials allow faster signal propagation speeds through IC interconnect by reducing the capacitance between lines. BCB is a silicon-containing spin-on polymer formed from the divinylsiloxane-bisbenzocyclobutene (DVS-BCB) monomer. Due to the silicon content, it retains many of adhesive and patterning advantages of SiO_2. BCB also offers other advantages, such as excellent planarization, low moisture absorption properties, good compatibility with copper and the ability to be cured by rapid thermal techniques. Metal or organic masks have been used for BCB patterning, but the soft mask patterning technique proposed by Garrow [10] greatly simplifies processing. The patterns to be etched were fabricated on 4-inch <100> silicon wafers. Approximately 5 microns of BCB film was deposited by spin coating and cured at 250 ºC for 1 hour. Twelve microns of AZP 4620 photoresist was subsequently coated on each wafer. This thick photoresist layer was used as a soft mask for 10 to 40μm on-a-side square patterns with approximately 15% open area.

Four process conditions were selected as input factors in the RIE characterization experiment. The experiment consisted of a 2^4 factorial design with three center points. This design was initially performed to ascertain the statistical significance of the input factors. The initial design was further augmented by a central composite circumscribes (CCC) design. The input factors and their respective ranges are shown in the Table 1.

Table 1. Input factors and their ranges of 2^4 factorial designed experiment

INPUT	RANGE	UNITS
SF_6	2 - 6	Sccm
O_2	20 - 60	Sccm
Pressure	100 - 300	Mtorr
RF Power	200 - 400	Watts

BCB film thickness was measured by ellipsometry, and photoresist thickness was measured by a profilometer after pattern transfer. Vertical etch rates were calculated by dividing the difference between pre-etch and post-etch depth by etch time. Lateral etch rates for BCB were calculated by dividing the half of difference between pre-etch and post-etch width by etch time. For the measurement of lateral etch rate, profilometer measurements were linearly scaled after six samples were measured and calibrated by scanning electron microscope (SEM) photos. SEM calibration of the profilometer measurements was less time-consuming and tedious than multiple SEM measurements. A sample SEM photo is shown in Fig. 4.

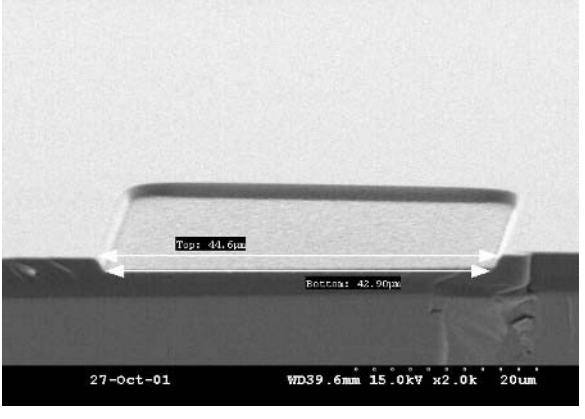


Fig. 4. A sample SEM photo for a 40 um of square pattern

Expressions of interest for responses such as lateral etch-rate *(L_{BCB})*, selectivity *(S_{PR})*, percent uniformity *(U)* and percent anisotropy *(A)* are as follows:

$$L_{BCB} = \frac{Top - Bottom}{2t} \tag{1}$$

$$S_{PR} = \frac{R_{BCB}}{R_{PR}} \tag{2}$$

$$U = \left\{ \frac{\left| R_{BCB,C} - R_{BCB,E} \right|}{R_{BCB,C}} \right\} \times 100\,(\%) \tag{3}$$

$$A = (1 - \frac{L_{BCB}}{R_{BCB}}) \times 100 \tag{4}$$

where L_{BCB} is the lateral etch rate, *Top* is the top-width of etched pattern, *Bottom* is the bottom-width of etched pattern, S_{PR} is the selectivity of BCB against photoresist, R_{BCB} is the etch rate of BCB, R_{PR} is the etch rate of photoresist, $R_{BCB,C}$ is the etch rate

of BCB measured in the center of a wafer, and $R_{BCB,E}$ is the etch rate of BCB averaged from the edge of a wafer.

3 Polynomial Neural Network Process Modeling

The PNN is a network transformation of $Rn \rightarrow R$, as shown in Fig. 5. The PNN learning is a self-organizing process designed to determine an appropriate set of Ivakhnenko polynomials that allow the activation of many neurons to achieve a desired state of activation that mimics a given set of sampled patterns [5]. These neurons are interconnected in such a way that the knowledge is stored in Ivakhnenko coefficients. The activation level of a node is determined by a nonlinear activation function with optimal complexity called a Ivakhnenko polynomial. This function usually has a form such as;

$$y = A + \sum_{i=1}^{n} B_i x_i + \sum_{i=1}^{n}\sum_{j=1}^{n} C_{ij} x_i x_j + \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} D_{ijk} x_i x_j x_k + \ldots \quad (5)$$

where x_i, x_j and x_k are nodal input variables, and y is the output of an individual neuron. A, B_i, C_{ij} and D_{ijk} are the coefficients of the Ivakhnenko polynomial. This activation function endows the PNN the ability to generalize with an added degree of freedom not available in statistical regression techniques.

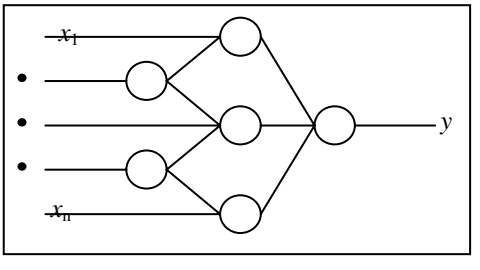


Fig. 5. The basic PNN configuration

The PNN utilizes the Group Method of Data Handling (GMDH) for data management. GMDH was first introduced by the Russian cyberneticist Ivakhnenko [5], and has been used to synthesize the building blocks of modeling methodology [7]. In GMDH, the data sample is divided into training set and validation set. The PNN methodology is implemented on the random input-output sets of measured training and validation data obtained from the experiments. An input matrix, consisted of measured (first layer) or calculated (second layer and more) input data, is presented to the each layer, and the output at each node is calculated by combining the inputs of each layer as pairs (or singles or triples) and filtering this, using the least square method. At each layer, new generations of complex equations are constructed which is evolved from simple form. Survival of the fittest principle (appropriate thresholds) determines the equations that can be passed to the next layer.

The calculated output vector is then compared to the measured output data vector, and the mean squared difference between these two vectors determines the error of this model. The number of layers is increased until the newer model begins to have poorer performance than their predecessors. This provides the self-organizing feature of the algorithm, leading to models of optimal complexity.

4 PNN Modeling of RIE Characteristics and Results

A PNN structure may be effectively employed to model the input-output behavior of the RIE experiments. Since there were four controllable input parameters and four measured output characteristics in this RIE experiments, four different models were constructed. Output of each model represents one characteristic of the RIE process. The number of inputs of each model was set to four; one controllable RIE parameter per one input node. The Predicted Squared Error (*PSE*), an estimate of the Mean Squared Error (*MSE*) of unseen data, is used as the performance metric. Initially, PNN RIE models were obtained using a set of input pairs and default structures. Since these rough models include all the polynomial type regression functions of input variables generated on the PNN structure, they were refined by varying the number of hidden nodes (neurons) and layers by analyzing all the *PSE*s at each layer, and select the polynomial coefficients to minimize the error. In evaluating learning (training) error, 20 RIE experimental data were used to build the models. The prediction error of this model was determined using 7 other experimental data which were not used during training process.

To search for the PNN architecture and coefficients of polynomial that minimized both the training error and prediction error, the following *PSE* was defined as a performance index:

$$PSE = MSE + KP \tag{6}$$

where *MSE* is the training mean squared error and *KP* is the complexity penalty (over fit penalty). The *KP* is defined by:

$$KP = PF \times \frac{2K}{N} \times S_p^2 \tag{7}$$

where K, N, and S_p^2 are determined by the data used to synthesize the network and *PF* is the penalty factor which is set to 1 in this model. N is the number of training data, S_p^2 is a prior estimate of the true error variance, and K denotes the total number of coefficients. The penalty factor allows the designer to accommodate the computational complexity in the calculation of the error term.

The *PSE* takes into account the complexity of the RIE system, and it attempts to reduce the predicted squared error. Initially, the *MSE* had a significant effect, but as the model increases in the size of training data, *KP* becomes more important. A minimum value of *PSE* will always exist, because *MSE* decreases with each additional trial but always remains nonnegative, whereas *KP* linearly increases as the number of trials increases. Therefore, *PSE* is minimized in the point of compromise of both *KP* and *MSE*. The smallest values of *PSE*s are used to determine the PNN

architecture and coefficient set that simultaneously minimize both training and prediction errors. Hence, the presence of both *MSE* and *KP* ensures that *PSE* favors simple models with low error.

In performing this analysis, it has been assumed that the optimal polynomials fitted by linear regression equations were sufficient to capture the dependence of PNN performance on the optimal coefficients. The fitted final polynomials were subsequently used to generate a PNN that optimized the training and prediction errors. Figures 6-9 show the synthesized PNN structure for the four output characteristics. Fig. 6 is a synthesized optimal PNN models for the anisotropy. The optimal model was determined to find input nodes from the satisfied node, after iteratively calculating the *PSE* until the *PSE* was smaller than the previous value. The equation shown in the figure represents the "Triple" in the network structure (shown in thick black square). The $x1$, $x2$, $x3$ in the equation represent the first input, the second input, and the third input respectively.

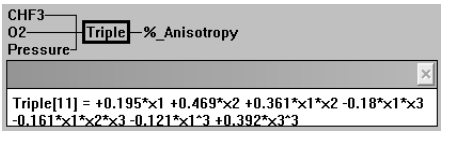


Fig. 6. Synthesized PNN for anisotropy

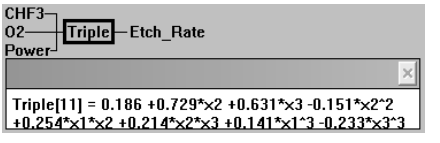


Fig. 7. Synthesized PNN for etch rate

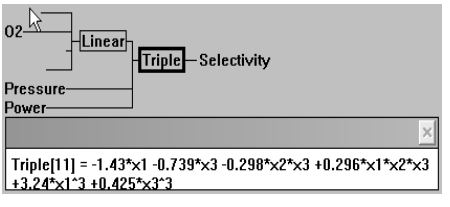


Fig. 8. Synthesized PNN for selectivity

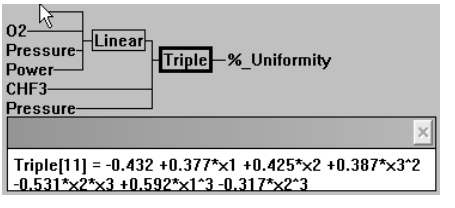


Fig. 9. Synthesized PNN for uniformity

The outputs obtained from each of these nodes are used to construct a high-degree polynomial model with automatic elimination of undesirable variable interactions. We can see the automatic elimination result of the synthesized network in the Figures 6-8. Although all the four inputs are used to synthesize network for the uniformity (Fig. 9), only three inputs are used to synthesize network for anisotropy (Fig. 6), etch rate (Fig. 7), and selectivity (Fig. 8). We can also see the self-organizing capability in the uniformity model (Fig. 9). Fig. 9 shows that the pressure was used as an input both for the first layer and for the second layer.

In order to establish the model prediction accuracy, the usual root-mean-squared (RMS) error (ε) metric was used:

$$\varepsilon = \sqrt{\frac{1}{(n-1)} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{8}$$

where n is the size of the data set, y is the measured value of the etch response and $\hat{y}$ is the response predicted response by the neural network model.

To compare the performance of prediction capability of the PNN with feed forward error back-propagation (FFEBP) neural network, FFEBP neural network models were trained using the same 20 experimental data. The remaining 7 runs were used to measure the prediction error of the FFEBP neural network models. Various FFEBP neural networks with different numbers of hidden layer neurons were trained until the RMS error was minimized. For all four responses, the neural networks with two hidden layers converged much more readily than those with a single hidden layer. Table 2 contains the structure and parameters of the FFEBP neural networks modeling four responses of interest. The RMS error of PNN models and FFEBP neural network models are compared and summarized in Table 3. The result shows that the prediction error for etch rate is remarkably improved. Although the prediction error for the uniformity is not so much as etch rate, the prediction error improvement is at least 16.9% compared to traditional FFEBP neural network modeling.

Table 2. FFEBP neural network training parameters

	Hidden 1	Hidden 2	η	Momentum
Etch Rate	4	3	0.03	0.03
Selectivity	4	5	0.05	0
Uniformity (%)	4	6	0.05	0
Anisotropy (%)	8	10	0.03	0.04

Table 3. Prediction RMS error comparison between FFEBP neural networks and PNNs

	FFEBP (RMS)	PNN (RMS)	% Improvement
Etch Rate	0.288	0.121	138.02
Selectivity	0.465	0.298	56.04
Uniformity (%)	25.899	22.154	16.90
Anisotropy (%)	15.950	8.185	94.87

5 Conclusions

Polynomial neural network modeling using GMDH method has been applied to the highly complex and nonlinear reactive ion etching (RIE) process. In contrast to the conventional regression technique, PNN has several distinct advantages: a smaller number of data set is required, less computational time, and the final structure of the PNN does not need to be specified. Although, high-order regression often leads to severely ill-conditioned system equations, the PNN avoids this by constantly eliminating variables and variable interactions at each layer, and helps to reduce linear dependencies. Therefore, complex systems can be modeled without specific knowledge of the system or massive amounts of data. This optimal network synthesis capability was shown in this paper using RIE experimental data. The prediction performance of the PNN models is superior to conventional FFEBP neural network models. The results show that at least 16.9% of prediction error is reduced using PNN models. This excellent PNN models can be used not only in analyzing the impact of

each input conditions on the output characteristics, but also in synthesizing the recipes for the etching with special characteristics.

Acknowledgement

This work was supported by the ERC program of MOST/KOSEF (Next-Generation Power Technology Center).

References

1. Himmel, C., May, G.: Advantages of Plasma Etch Modeling Using Neural Networks Over Statistical Techniques. IEEE Trans. Semi. Manufac. 6 (1993) 103-111
2. Mocella, M., Bondur, J., Turner, T.: Etch Process Characterization Using Neural Network Methodology: A Case Study. SPIE Proceedings on Module Metrology, Control and Clustering 1594, (1991) 232-242
3. Kim, B., Han, S., Kim, T., Kim, B., Shim, I.: Modeling Refraction Characteristics of Silicon Nitride Film Deposited in a SiH4-NH3-N2 Plasma Using Neural Network. IEEE Transactions on Plasma Science 31(3) (2003) 317-323
4. Han, S., Bidstrup, S., Kohl, P., May, G.: Modeling the Properties of PECVD Silicon Dioxide Films Using Optimized Back-Propagation Neural Networks. IEEE Trans. Comp. Pack. and Manufac. Part A 17(2) (1994) 174-182
5. Ivakhnenko, A.: The Group Method of Data Handling - A Rival to Stochastic Approximation. Soviet Automat. Contr. 13(3) (1968)
6. Farlow, J.: Self-Organizing Methods in Modeling GMDH Type Algorithms. New York, Marcel Dekker. (1984)
7. Madala, H., Ivakhnenko, A.: Inductive Learning Algorithms for Complex Systems Modeling. Roca Raton, CRC Press (1944) 1-42
8. May, G., Huang, J., Spanos, C.: Statistical experimental design in plasma etching modeling. IEEE Trans. Semi. Manufac. 4(2) (1991)
9. Box, G., Hunter, W., Hunter, J.: Statistics for Experiments. New York, Wiley (1978)
10. Roger, B., Berry, M., Turlik, I., Garrow, P., Castillo, D.: Soft Mask for Via Patterning in Benzocyclobutene. Int. J. Micro. and Elect. Packaging 17(3) (1994)

Wafer Yield Estimation Using Support Vector Machines

Lei-Ting Chen[1], David Lin[2], Dan Muuniz[2], and Chia-Jiu Wang[2]

[1] School of Computer Science and Engineering,
University of Electronic Science and Technology of China, Chengdu, China
richardchen@uestc.edu.cn
[2] Department of Electrical and Computer Engineering,
University of Colorado at Colorado Springs, Colorado Springs, CO, USA
cwang@eas.uccs.edu

Abstract. Wafer yield estimation is a very complicated nonlinear problem due to many variations in fabrication processes at different silicon foundries. The purpose of this paper is to use Support Vector Machines (SVMs) to analyze and predict electrical test data, which are traditionally captured by probing each chip on the wafer. The predicted data produced by the support vector machines is then compared with the known measured data to determine the accuracy. Once the SVM has captured nonlinear relationship between fabrication processes and wafer yields, it can be used to predict wafer yield in other lots fabricated by the same silicon foundry. The advantage of using this approach is to save time due to probing hardware constraints, predict wafer yield across the same fabrication process and give an alternative method of device simulation. Our experiments show that the SVMs predict more accurate than classical device physics equations and in some cases SPICE simulation software in comparison with the actual measured electrical data. Electrical data used for this research include threshold voltages, saturation currents, and leakage currents.

1 Introduction

Traditional methods of testing wafers can be very time consuming and costly. This is due to the fact that each wafer and every die on the wafer must be thoroughly tested using a lot of hardware and man resources. It is not uncommon to take more than one working day to measure electrical test (e-test) data from a single wafer. A wafer lot usually contains up to 25 wafers. The time required to measure all wafers in a lot exceeds two weeks easily even with modern testing equipments. These individual wafers may be tested in parallel, but additional hardware is needed. Measured results may also vary depending on variations within different testing hardware. For example, testing the same die on two different wafer probe stations with identical hardware may show slightly different results. Although it is very time consuming to fully characterize wafers, it is a necessity for process flow development, model extraction, and design. In this paper, we explore the possibility of reducing the wafer testing time by using the support vector machine learning algorithm. A Support Vector Machine (SVM) can be trained on a database and later used to predict new outputs based on the training data and new inputs [1 – 6]. Instead of measuring all

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1053 – 1058, 2006.

wafers using probing stations, the SVM could predict the same desired electrical data in a much reduced amount of time.

2 Wafer Yield and Electrical Test Data

Electrical test data commonly refer to characterized devices taken from wafers after completion of fabrication processes. The characteristics of a device are measured and used to determine the pass or fail of a device. The more the dices on a wafer pass the specifications and the higher the yield the wafer has. In general, several characteristics from multiple dies are analyzed to determine the wafer yield. Device characterization refers to measuring an electrical device (transistor, capacitor, diode, resistor, etc.) to determine its current-voltage characteristics.

It is important in determining whether a fabrication process meets specifications and design rules. If requirements are not met, physical parameters can be extracted from the measured data to provide possible solutions for failure issues. If all specifications and design rules pass, then the characterized data will be used to extract device models for designers. In most cases, a device is characterized by using manual or automated probe stations in which specific voltage conditions are applied to measure the desired current.

3 Regression SVM for Wafer Yield Prediction

A Support Vector Machine was first developed by Vladimir Vapnik and others at AT&T Bell Laboratories in the early to mid 1990s. Support Vector Machines can map the input space into a very high dimensional feature space to find an optimal solution in a high dimensional space. The major functions of SVM can be classified into two areas: classification and regression. Two SVM kernel functions i.e., radial base function (RBF) and dot product are used in this work. In our application we use SVM for regression applications. In this paper measured data was taken from wafers processed by the 0.18μm technology with the operating voltage = 1.8 Volts. Three measurement parameters such as linear threshold voltage, saturation current, and leakage current are the focus of all the experiments.

3.1 Wafer Yield (Threshold Voltage, Saturation Current and Leakage Current)

In general, threshold voltages, saturation currents, and leakage currents are the most important parameters in determining yield and passing/failing specification limits for transistors. The following measurement conditions are used for threshold voltages, saturation currents and leakage currents.

Linear Threshold Voltage: VD = 0.1V, VS = VWELL = 0.0V, VG is swept from 0V to 1.8V to calculate maximum slope of ID, or maximum GmMax. The threshold voltage is calculated using the following equation.

$$VT = - (Intercept/Slope) - VD/2 \quad (1)$$

Saturation Current: Drain Current at VD = VG = 1.8V, VS = VWELL = 0.0V
Leakage Current: Drain Current at VD = 1.8V, VG = VS = VWELL = 0.0V

3.2 Wafer Experiments

In our experiments, each tested wafer has 149 dies and each die has about 40 CMOS transistors with different channel widths and lengths. Table 1, Figure 1 and Figure 2 present results obtained by SVM, SPICE simulations, and analytical calculations [7] for threshold voltages, saturation currents, and leakage currents. The radial basis function (FBF) is selected as the kernel function for the SVM. For the first experiment, the SVM is essentially used for interpolation because the training data set covers the most part of the wafer and the testing data is selected from part of the same wafer even though the testing data set is different from the training data set.

Table 1. Threshold Voltage Using RBF Kernel SVM, SPICE, and Calculation

Width	Length	Measured	SVM	SPICE	Calculated	%diffSVM	%diffSPICE	%diffCALC
10	10	0.591	0.591	0.591	0.682	0.0	0.0	15.4
10	5	0.604	0.604	0.595	0.691	0.0	-1.4	14.5
10	2	0.633	0.633	0.607	0.724	0.0	-4.1	14.4
10	1	0.652	0.656	0.628	0.753	0.6	-3.8	15.4
10	0.26	0.664	0.658	0.680	0.833	-0.9	2.3	25.3
10	0.24	0.659	0.658	0.673	0.833	-0.1	2.2	26.4
10	0.18	0.598	0.657	0.630	0.773	9.9	5.3	29.3
10	0.16	0.573	0.657	0.607	0.727	14.6	5.8	26.9
2	0.18	0.633	0.633	0.623	0.768	0.0	-1.5	21.4
0.8	0.18	0.633	0.633	0.611	0.761	0.0	-3.5	20.1
0.6	0.18	0.617	0.629	0.604	0.746	2.0	-2.0	20.9
0.4	0.2	0.630	0.620	0.609	0.763	-1.6	-3.4	21.2
0.4	0.18	0.648	0.624	0.590	0.732	-3.8	-9.1	12.8
0.3	0.18	0.600	0.620	0.574	0.696	3.3	-4.3	16.0
0.24	0.2	0.587	0.614	0.578	0.671	4.6	-1.6	14.3
0.24	0.18	0.635	0.618	0.558	0.685	-2.7	-12.1	7.9
0.22	0.2	0.613	0.613	0.570	0.674	0.0	-7.0	9.9
0.22	0.18	0.562	0.617	0.551	0.605	9.8	-1.9	7.7

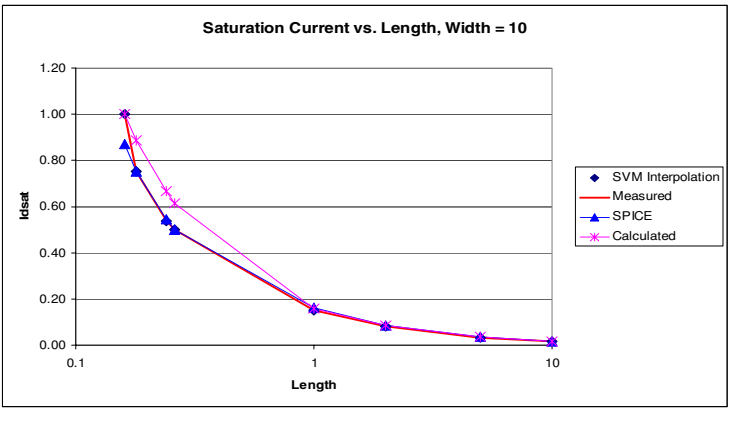


Fig. 1. Saturation Current Length Scaling (RBF SVM, SPICE, & Calculation)

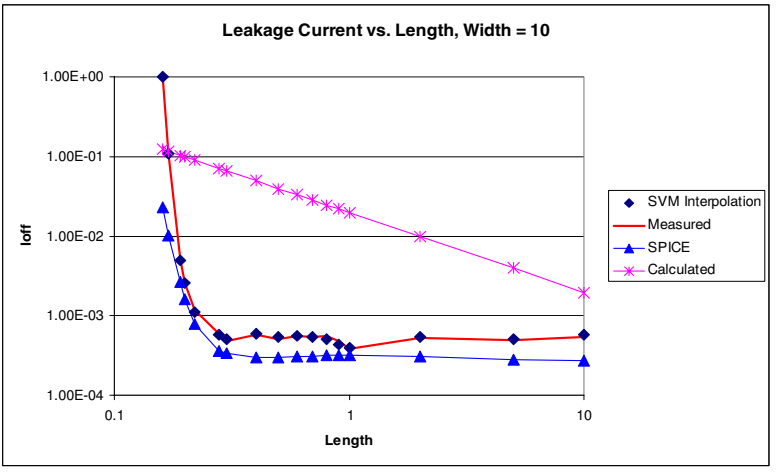


Fig. 2. Leakage Current Length Scaling (RBF SVM, SPICE, & Calculation)

For all three parameters, SVMs produce more accurate results than SPICE and analytical equations calculations in comparison with the measured data. The SPICE model has limitations in capturing exact accuracies across numerous widths and lengths. Classical analytical equations do not capture the varying effects across dimensions i.e. short-channel effect, narrow-channel effect).

For the second experiment, a single die with varying channel widths and lengths was used to predict threshold voltages, leakage currents and saturation currents. The importance of prediction across varying widths and lengths would be useful for device model extraction. It is extremely time consuming to measure all the dimensions needed for model extraction, and even more difficult to fit them during the model development process. It would be ideal if only half the measurements were needed and use SVM to fit the other dimensions through prediction. The measured data used in this experiment was identical to that used for interpolation. Each training file was identical to what was used for interpolation, with the exception of a few dimensions that were intentionally left out. The idea behind this was to have SVM predict the correct values for those missing dimensions during test.

Figure 3 and Figure 4 represent length scaling curves for constant width of 10 showing the SVM prediction vs. measured values. As seen from the prediction tables and plots, SVM predicts accurately for threshold voltage and saturation current. This is most likely due to values not having much variation across the varying dimensions.

The next sets of experiments focus on wafer prediction using SVM. The first wafer experiment focused on measured threshold voltage and saturation current from 149 die for a W/L = 10/10 device. The SVM predicts threshold voltages and saturation currents with 97% accuracy comparing to the actual measured data. Due to page limitation, the detail results are not presented in this paper about this set of experiments.

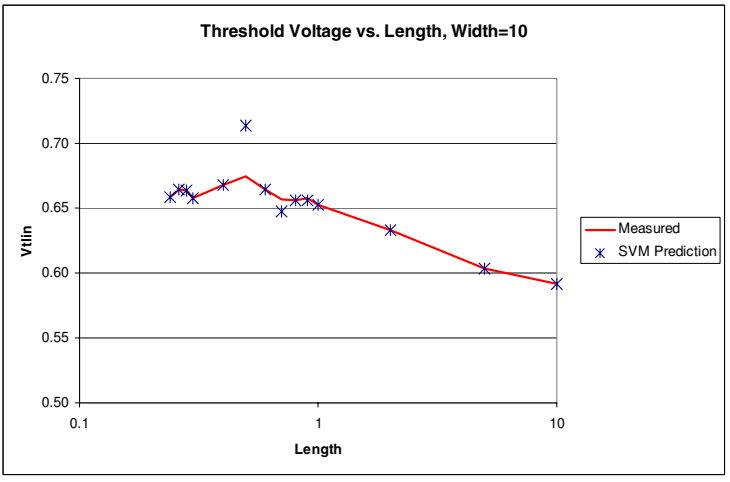


Fig. 3. Threshold Voltage Length Scaling (SVM Prediction vs. Measured)

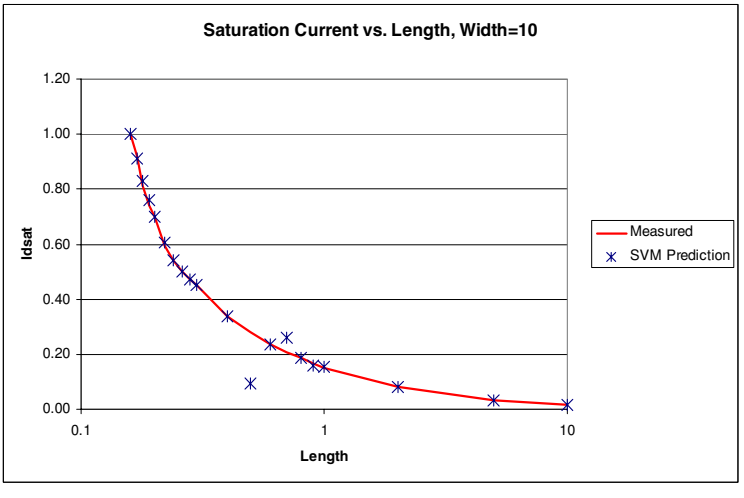


Fig. 4. Saturation Current Length Scaling (SVM Prediction vs. Measured)

The final set of wafer experiments was based on the idea of using one wafer as training data to predict data for another wafer. Because it is unrealistic to completely predict electrical data for an unknown wafer, training data was taken from one entire wafer and 30% of the second wafer. The goal now is to predict the other 70% of the second wafer. Again, threshold voltage and saturation current were used as the key test parameters, while leakage current data were limited. The input values for the training files now contain a column for wafer number as well as channel width and channel length. In this experiment, it was decided to use dimensions of W/L = 10/10, 10/0.18, 0.22/10, and 0.22/0.18 to understand how SVM would handle prediction

across different dimensions. Multiple training data files were needed for each dimension. In all cases, the SVM predicts values within 10% of the actual measured data. Due to page limitation, the detail results are not able to be presented in this paper about this set of experiments.

4 Discussions

From all experiments in this work, it has been observed that the SVM can predict electrical data very accurately within a single die. Within a wafer, the SVM can produce electrical data with 90% accuracy. Using all data from the first wafer and 30% data of the second wafer, the SVM can predict the electrical data, i.e., threshold voltages and saturation currents of the remaining 70% of the second wafer with 90% accuracy. This means that there is a potential to reduce the wafer testing time by 35%.

References

1. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machine (and other Lernel-Based Learning Methods). Cambridge University Press (2000)
2. Schoelkopf, B., Smola, A.J., Williamson, R.C., Bartlett, P.L.: New Support Vector Algorithms. Neural Computation 12 (2000) 1207-1245
3. Schölkopf, B., Williamson, R.C., Smola, A.J., Shawe-Taylor, J.: SV Estimation of a Distribution's Support. In: Solla, S.A. and Leen, T.K. and Müller, K.-R. (eds.): Neural Information Processing Systems, 12 MIT Press.
4. Joachims, T.: Making Large-scale SVM Learning Practical. In: Advances in Kernel Methods - Support Vector Learning, Chapter 11. MIT Press (1999)
5. Tobias, S., Thorsten, J.: Expected Error Analysis for Model Selection. In International Conference on Machine Learning (ICML) (1999)
6. Vapnik, V.: Statistical Learning Theory. Wiley (1998)
7. John, P. U.: Circuit Design for CMOS VLSI. Kluwer Academic Publishers (1992)

Dynamic Soft-Sensing Model by Combining Diagonal Recurrent Neural Network with Levinson Predictor

Hui Geng, Zhihua Xiong, Shuai Mao, and Yongmao Xu

Department of Automation, Tsinghua University, Beijng 100084, P.R. China
zhxiong@tsinghua.edu.cn

Abstract. Dynamic soft-sensing model of diesel oil solidifying point (DOSP) in crude distillation unit (CDU) is proposed based on diagonal recurrent neural network (DRNN). Because of long time-delay of the DOSP measurements, multi-step-ahead predictions are obtained recursively by Levinson predictor and then used as input of DRNN. Simulation results on the actual industrial process data show that the proposed dynamic soft-sensing model took good effects practically and significantly diminished the time-delay of output value.

1 Introduction

In recent years, soft-sensing models are developed and implemented widely in the control of product quality in industrial processes. For most industrial processes, there exists the lack of suitable on-line quality measurements. Most laboratory analyzers suffer from large measurement delays and high investment and maintenance cost. Even in some cases, online analysis devices, e.g. near-infrared instrument, are used to measure the product quality, but time-delay of such online device always exists. However, soft-sensing model utilizes the on-line measurable variables (secondary variable) to predict the product quality variable (primary variable) through certain modeling approaches, such as mechanistic model, statistical model, and artificial intelligent model, etc [1]. Due to the uncertainty, complexity and nonlinearity of industrial processes, mechanistic models are often unavailable. Therefore, data-driven empirical models, such as neural network, are useful alternatives. Neural networks (NN) have been shown to be able to approximate any continuous nonlinear functions and have been applied to nonlinear process modeling and control [2][3].

Most researches in the soft-sensing modeling have been essentially concentrated on static soft-sensing models. Since in practical industrial applications, soft sensors based on static models often lead to low accuracy and poor robustness, dynamic soft-sensing model should be developed. Feed-forward neural network (FNN) is usually used to build the soft sensor. However, the FNN is a static mapping and can only represent a dynamic system mapping by combining tapped delays of network inputs and the back-propagation (BP) training algorithm. On the other hand, recurrent neural networks (RNN) [4][5] have important capabilities not found in FNN, such as attractor dynamics and the ability to store information for later use. Of particular interest is their ability to deal with time-varying input or output through their own natural temporal operation [5]. Thus RNN is a dynamic mapping and is better suited for dynamical systems than the feed-forward network.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1059 – 1064, 2006.

In this study, the focus is on coping with the time-delay of product quality measurement and building a dynamic soft-sensing model for product quality (DOSP) in a crude distillation unit. The model is based on diagonal recurrent neural network (DRNN) and combined with Levinson predictor. Multi-step-ahead estimation of its measurement is given by Levinson predictor and then used as input of DRNN.

2 Dynamic Soft-Sensing Model of DOSP

In this study, a crude distillation unit (CDU) in an industrial petroleum refining enterprise is considered. Diesel oil is one of the main products in the CDU and the diesel oil solidifying point (DOSP) is the most important index to scale the grade of the product. Of particular interest is the control of consistent product quality. However, due to unknown disturbances during the operation of CDU, the DOSP is affected by many operation conditions, such as the property of the refining crude oil, oil feed rate, distillation pressure, and so on. In practice, an on-line analysis device is implemented to measure the DOSP and it can provide the value every 4 minutes, but the measurement of the DOSP is delayed by almost 20 minutes. So it is difficult to use the measured value to control the product quality on-line.

The aim of the soft-sensing model is to predict the DOSP at the current time, and then it is possible to control the product quality on-line. Fig. 1 shows the whole structure of the novel dynamic soft-sensing model.

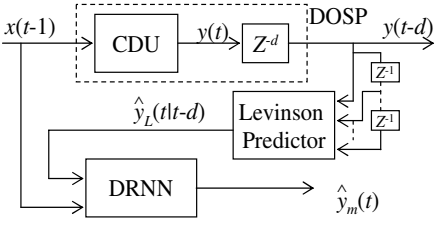


Fig. 1. The whole structure of dynamic soft-sensing model of DOSP in CDU

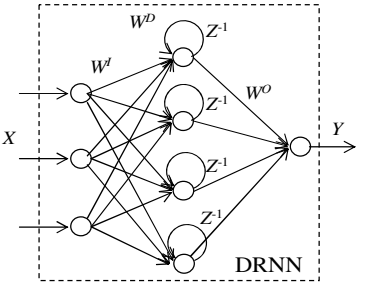


Fig. 2. A structure of DRNN

2.1 Multi-step-ahead Prediction by Levinson Predictor

Levinson predictor [6] is used here to provide the product estimation at current time based on the delayed measurement of product quality. In the Levinson predictor, the output historical operation data is only used, and the prediction is obtained by computing the moving average value of output, which can be described as [6]

$$\hat{y}_L(k) = -\sum_{i=1}^{q} a_i y(k-i) \tag{1}$$

where k represents the current sampling time, $\hat{y}_L(k)$ is the predicted output, $a_i(i=1,\ldots,q)$ are the model parameters, and q is the model order of the Levinson predictor,

respectively. It can be observed that $\hat{y}_L(k)$ is calculated only using the former outputs $y(t\text{-}1)$ to $y(t\text{-}q)$. On the assumption that the future output be replaced by the predicted output $\hat{y}_L(k|k-d)$, the d-step-ahead output can be obtained at current time k by using Eq(1) recursively, which can be written as

$$\hat{y}_L(k\,|\,k-d) = -a_1\hat{y}_L(k-1\,|\,k-d) - \cdots - a_{d-1}\hat{y}_L(k-d+1\,|\,k-d) - a_d y(k-d) - \cdots - a_q y(k-q) \tag{2}$$

The parameters $\{a_i\}$ can be identified by least squares method using the historical operation data set. The model order q can be determined by examining sum of squared errors (SSE) of output prediction. After the Levinson predictor is developed, its prediction can be considered as the "measurement" of product quality at the current time and then used as the input of the DRNN, as shown in Fig. 1.

2.2 Diagonal Recurrent Neural Network

A structure of DRNN is shown in Fig. 2. It consists of three layers, but particularly it has one hidden layer with sigmoid-type recurrent neurons. Unlike the fully-connected recurrent neural network (FRNN), DRNN has no interlinks among neurons in the hidden layer [5]. Since a recurrent neuron already has an internal feedback loop in the hidden layer, it can capture the dynamic behavior of a system without external feedback through tapped delays in the input layer. Considering the DRNN shown in Fig. 2, it consists of m inputs, n recurrent neurons in the hidden layer, and one linear output. The mathematical model of DRNN can be expressed as [5]

$$y_m(k) = \sum_{i=1}^{n} w_i^O h_i(k),\ h_i(k) = f(s_i(k)),\ \ s_i(k) = w_i^D h_i(k-1) + \sum_{j=1}^{m} w_{ij}^I x_j(k) \tag{3}$$

where $w^I(n\times m)$, $w^O(1\times n)$ and $w^D(1\times n)$ are weight vectors of the input, output, and hidden layers, $x_j(k)$ is the jth input of the network, $s_i(k)$ is the sum of inputs to the ith recurrent neuron, $h_i(k)$ is the output of the ith recurrent neuron, $f(\cdot)$ is a usual sigmoid function of the recurrent neuron, and $y_m(k)$ is the output of the network, respectively. From Eq(3), we can obtain

$$\begin{aligned} y_m(k) &= \sum_{i=1}^{n} w_i^O f\left(\sum_{j=1}^{m} w_{ij}^I x_j(k) + w_i^D h_i(k-1)\right) \\ &= \sum_{i=1}^{n} w_i^O f\left(\sum_{j=1}^{m} w_{ij}^I x_j(k) + w_i^D f\left(\sum_{j=1}^{m} w_{ij}^I x_j(k-1) + w_i^D h_i(k-2)\right)\right) \\ &= \cdots = \sum_{i=1}^{q} g_i(x_j(l)) = F(x_j(l)) \ , \qquad 1 \le l \le k, j = 1,2,\ldots,m \end{aligned} \tag{4}$$

where $F(\cdot)$ is a nonlinear function. It can be seen that for each discrete time k, the output of DRNN is not only the function of the current input $X(k)$, but also the function of all inputs $X(l)$ up to time k. That is, DRNN is a dynamic mapping which is very useful for associative memory purpose, and can be utilized in simulating dynamic systems. DRNN can only be trained by dynamic back-propagation (DBP) algorithm, because the output of DRNN is related to all inputs before current time, so during the training the output error of DRNN is also related all output errors before. The error function for DRNN can be defined as

$$E = \frac{1}{2}\sum_{k=}^{N} e^2(k) = \frac{1}{2}\sum_{k=}^{N} \left(T(k) - y_m(k)\right)^2 \tag{5}$$

where T is the desired output, and N is the amount of training sample.

Let W be the weight vector composed of w^I, w^O and w^D. From Eq(3) and (5), we can obtain the gradients of the output $y_m(k)$ with respect to W, and then the weights of DRNN can be adjusted as follows

$$W(n+1) = W(n) + \eta\left(-\frac{\partial E}{\partial W}\right) = W(n) + \eta e(k)\frac{\partial y_m(k)}{\partial W} \tag{6}$$

where η is a learning rate, and

$$\frac{\partial y_m(k)}{\partial w_i^O} = h_i(k) \tag{7}$$

$$\frac{\partial y_m(k)}{\partial w_i^D} = w_i^O P_i(k),\ P_i(k) = f'(s_i(k))(h_i(k-1) + w_i^D P_i(k-1)),\ P_i(0) = 0 \tag{8}$$

$$\frac{\partial y_m(k)}{\partial w_{ij}^I} = w_i^O Q_{ij}(k),\ Q_{ij}(k) = f'(s_i(k))(x_j(k-1) + w_i^D Q_{ij}(k-1)),\ Q_{ij}(0) = 0 \tag{9}$$

According to above dynamic training algorithm Eq(6)-Eq(9), the weights W are usually updated each time for each training sample. When the number of training data set is large, the procedure of training is very time-consuming. In this study, we applied a batch training scheme to compute the errors of all training samples at each iteration and then update the weights. This scheme can avoid overmuch weight revision and lead to great decrease of training time.

In the above dynamic soft-sensing model based on DRNN, both the prediction $\hat{y}_L(k|k-d)$ obtained by Levinson predictor and other on-line measured variables are used as inputs of the DRNN, and the output of DRNN is the prediction $\hat{y}_m(k)$ of DOSP at the current time k.

3 Application to Actual Process Data

From the industrial crude distillation unit, over 5000 data sets of samples, including the delayed analyzed product values DOSP, were collected from the plant and used to build and test the dynamic soft-sensing model. These samples almost covered all normal operation conditions of CDU. To pre-process data set, any outliers or bad samples were first verified and deleted, and then all data were scaled to zero mean and unit variance.

The measurement of the DOSP is delayed by almost 20 minutes, the sampling time of the model is set to be 6 minutes, and therefore the multi-step-ahead prediction of the Levinson predictor is set to be d=3. After examining sum of squared errors (SSE) of predictions with different model order q, it was found that the value of q=7 obtained the best performance. Then the parameters $\{a_i\}$ were identified by least squares method, and the Levinson predictor was described as follows:

$$\hat{y}(t \mid t-3) = 0.932\,\hat{y}(t-1 \mid t-3) + 0.159\,\hat{y}(t-2 \mid t-3) - 0.0241\,y(t-3) + 0.0872\,y(t-4) + 0.0006\,y(t-5) - 0.0566\,y(t-6) + 0.0409\,y(t-7) \quad (10)$$

where $\hat{y}(t-1|t-3)$ and $\hat{y}(t-2|t-3)$ are calculated recursively by Eq(2), and $y(t\text{-}3),\ldots, y(t\text{-}7)$ are the historical values by the analysis device.

As well known in the soft-sensing model, the model accuracy is affected strongly by the input variables or secondary variables. From the historical operation data, the product quality DOSP is very complicated with over 20 correlated variables that can be measured on-line. After the statistical correlation analysis of all data sets, 7 related variables were selected, including the oil feed rate, the distillation pressure, the temperature of top distillation, the temperature and flow rate of the third outlet, the reflux rate of the second inlet, and the flow rate of the second outlet. The prediction $\hat{y}(t|t-3)$ is also used as another input of the DRNN. Almost 4000 sets of data were used to train the DRNN model, 1200 sets of data were used as testing data and 100 sets of data were used as unseen data to assess the generalization performance. As described above, the number of input nodes was set to 8, but the number of hidden layer neurons of DRNN was determined through cross-validation. Networks were trained using the batch dynamic training algorithm. Networks were trained on the training data, and the network with the least sum of squared prediction errors on the validation data was chosen and 15 hidden neurons were utilized.

Fig. 3 shows the predictions of the trained DRNN models on the testing data set. It is clear that the DRNN model has captured the dynamic trends of the product quaility in the data quite well. The prediction errors are also shown in Fig. 4, and the root-mean-square-error (RMSE) is 0.25 and almost 95% prediction errors are less than 0.5. Fig. 5 shows the comparison between model predictions and actual values in the unseen data set. It can be seen that the model predictions almost lead 3 sampling interval than the actual DOSP value. The time-delay of the analyzed DOSP was diminished quite well. Thus it is possible to implement this dynamic soft-sensing model for the closed-loop control of the product quality.

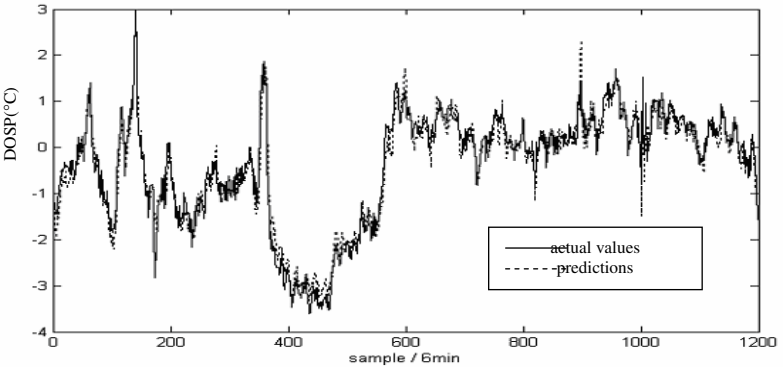


Fig. 3. Predictions of dynamic soft-sensing model on the testing data

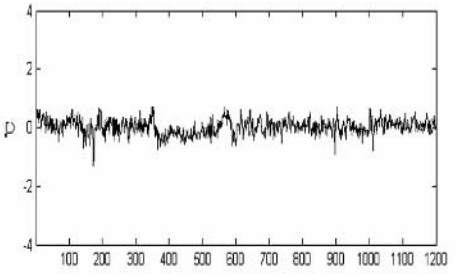


Fig. 4. Prediction errors on the testing data

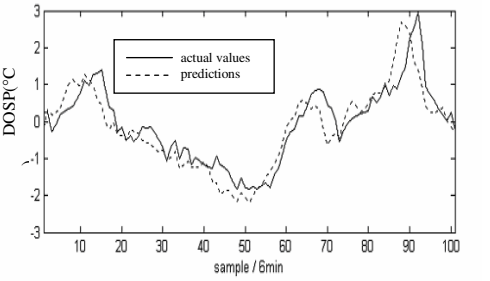


Fig. 5. Predictions on the unseen data

4 Conclusions

A novel dynamic soft-sensing model of DOSP in CDU based on the DRNN and Levinson predictor is proposed in this paper. Levinson predictor gives an optimal estimation of the DOSP measurement and then it is used as the input signal of DRNN. By the multi-step-ahead predictions, the effect of long time-delay is degraded. Simulation results on the actual industrial process data show that the predictions of the dynamic model are accurate and make it possible to implement the product control.

Acknowledgements

The authors greatly appreciate the financial support of the NSFC (No. 60404012) and SRF for ROCS, SEM of China.

References

1. Luo, J.X., Shao, H.H.: Developing Soft Sensors Using Hybrid Soft Computing Methodology: a Neurofuzzy System Based on Rough Set Theory and Genetic Algorithms. Soft Computing 10(2006) 54-60
2. Sjoberg, J., Zhang, Q., Ljung, L., Benveniste, A. *et al.*: Nonlinear Black-box Modeling in System Identification: a Unified Overview. Automatica 31 (1995) 1691-1724
3. Bhat, N., McAvoy, T.: Use of Neural Nets for Dynamic Modeling and Control of Chemical Process Systems. Comput. Chem. Eng. 14 (1990) 573-583
4. Xiong, Z.H., Zhang, J.: A Batch-to-Batch Iterative Optimal Control Strategy Based on Recurrent Neural Network Models. Journal of Process Control 15(2005) 11-21
5. Chao, C.K.: Diagonal Recurrent Neural Networks for Dynamic Systems Control. IEEE. Trans. on Neural Networks 6(1995) 144-155
6. Goodwin G. C., Sun G.S.: Adaptive Filtering, Predict and Control. Science Press, Beijing (1992)

Thermal Properties Reduced Models by ANN in Process Simulation

Xia Yang, Rongshan Bi, Yugang Li, and Shiqing Zheng

Qingdao University of Science and Technology, Qingdao 266042
yangxia@qust.edu.cn

Abstract. It's very time-consuming to evaluate thermal properties by rigorous methods in process real-time simulation, especially when the simulated project relates to multi-units and multi-components, which takes about 70 to 80 percent of the total simulation time. We developed a new reduced method for thermal properties evaluation based on the artificial neural net(ANN), in which we established several reduced evaluation models using ANN, such as models of vapor-liquid equilibrium, models of vapor-liquid enthalpy and models of temperature calculated from given enthalpy. We used the reduced models in a dynamic distillation simulation. Compared with rigorous thermal properties models, the ANN-reduced models could save 10 to 20 times simulation time with a satisfied accuracy. The results show it's an efficient and effective method.

1 Introduction

Process simulation has been a major tool for chemical engineer,there have been developed several general commercial simulation softwares. Thermal properties evaluation models is very important for it determines the accuracy and efficiency of the simulation. We cannot imagine an inaccurate vapor-liquid equilibrium could give an accurate distillation simulation results. While thermal properties evaluation is very tedious and time-consuming for amounts of iterations and test-and-errors, it will be calculated for thousands times in a simulation and take more than 70 to 80 percents in the total simulation time[1]. So rigorous models of thermal properties evaluation are hard to satisfy dynamic simulation which needs real-time.

To meet the need of real-time, several local thermodynamic models are proposed[2]. The basic theory of local thermal properties models is to substitute a simple function for the rigorous models in local range , the simple function could describe the complex thermodynamic behavior of temperature, pressure or composition. However, the local models have limits, firstly these methods are local, which are hard to adapt to a large range. secondly, the parameters in these model must be updated, so there must have an update-strategy, and which must call the rigorous model to calculate the updated parameters.

With the development of the computer and calculation technique, artificial intelligence (AI) has made a great progress, artificial neural network (ANN) as one of the important branch is used widely. ANN has a great advantage in dealing with

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1065–1070, 2006.

complex system model establishment, it combines simple nonlinear functions together to describe complex system and has a remarkable ability to derive meaning from complicated or imprecise data. In addition, ANN has the other advantages including adaptive learning, self-organization,and auto-associative. ANN only needs the input and output data to find the relations between them, and because its information is stored distributed, it has great stability.

The use of feed forward neural networks for process modeling has proven very successful for steady-state applications[3], Qu et al.[4]predict vapor-liquid equilibrium constant by three layer BP neural network, and the results show the predicted data are in good agreement with the experimental data. Wei et al[5]predicted the density of saturated liquid. So, ANN has the advantage of dealing with huge amounts of data in chemical engineering.

Based on the advantages of ANN, we established several reduced models of thermal properties evaluation in process real-time simulation.

2 Structure of ANN Used in Thermal Properties Models

We select the feed forward artificial neural networks with back propagation algorithm, the structure we used is as Fig.1, it has one input layer, one hidden layer and one output layer[6].So we use the number of every layer nodes to describe the structure of ANN used in thermal properties models, for example if we say the ANN structure is 8:10:6, that means input layer nodes are 8, hidden layer nodes are 10,output layer nodes are 6.

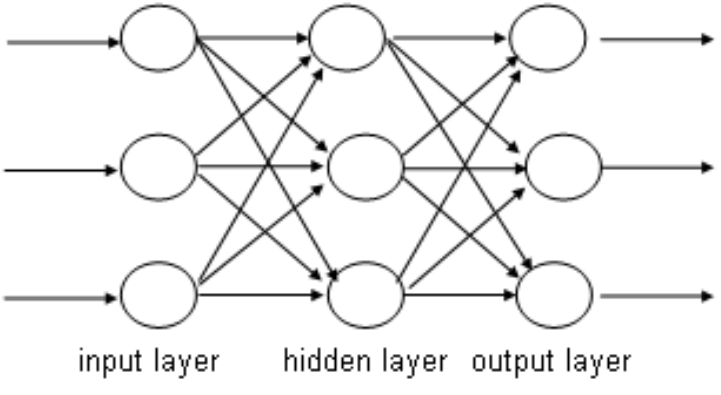


Fig. 1. the structure of ANN used in thermal properties reduced models

3 Steps to Establish ANN-Reduced Model

The steps to establish ANN-reduced thermal properties models are as followed.

1. Analyze the thermal property rigorous model and define the input and output layer of ANN model;

2. Select the rigorous method to calculate enough data as training sample and test sample;
3. Define the hidden layer and parameters of ANN, such as the training rate and training times and so on;
4. Test the training data using the test sample, if the allowance is satisfied, goto step 5; else goto 3;
5. Pick the weights of the network, the ANN-reduced thermal properties model is established;

4 Characters of ANN Used in Thermal Properties Models

Before we establish ANN-reduced thermal properties model used in process real-time simulation, we study some relative characters of ANN used in thermal properties models, including how they affect the accuracy and efficiency of the reduced models,such as the number of ANN nodes, the learning rate, the training times. We select y=tanh(x) as the transfer function.

4.1 The Effect of ANN Nodes Number

We take a three-component non-ideal mixture to study how the number of the ANN nodes affect the accuracy of the reduced models.

Vapor-liquid enthalpy is determined by the pressure (P), temperature (T) and compositions (X) of the mixture, that means giving the P,T,X of a mixture, the vapor-liquid enthalpy can be calculated. So we establish the vapor-liquid enthalpy ANN-reduced model, for this three-component mixture, the input layer has 5 nodes(P,T,X1,X2,X3) and the output layer has 2 nodes (vapor enthalpy, liquid enthalpy), the learning rate is 0.3, and the training times set as 100,000 times. we change the number nodes of the hidden layer to see its effect. The results show that the number of hidden layer nodes affect the accuracy of ANN-reduced model greatly. If nodes is few (fewer than 4),the calculation error compared with rigorous models is very big, but increasing the number of hidden layer nodes, the accuracy is not always improved, it has an optimum number. The optimum number of hidden layer nodes are usually 2 5 more than input layer nodes.

4.2 The Effect of ANN Learning Rate

We also take the above example to study how the learning rate of the ANN affect the accuracy of the reduced models, just set the nodes of the hidden layer as 8, and change the learning rate to see its effect. The results show that the learning rate also affect the accuracy of ANN-reduced model greatly, and it also has an optimum value, too lower(lower than 1.0) or too higher(higher than 5.0) learning rate may decreased the accuracy of ANN-models,generally we set it between 1.5 to 3.0.

4.3 The Effect of ANN Training Times

We take a six-component ideal mixture to study how training times of the ANN affect the accuracy of the reduced models. The mixture includes 1-3-butadiene, 1-butylene, n-pentane, 1-amylene, 1-butylethylene and benzene.

We establish a Vapor-liquid equilibrium (VLE) reduced model by ANN. VLE is also determined by the pressure (P), temperature (T) and compositions (X) of the mixture. So we establish VLE ANN-reduced model, for this six-component mixture, the input layer has 8 nodes(P,T,X1,...,X6) and the output layer has 6 nodes (six equilibrium constants), the nodes of the hidden layer are set as 10, and the learning rate is 2.3. we change the training times to see its effect. The results show that the accuracy of ANN-reduced model is improved by increasing the training times, but when the training times are increased to 50,000, the improvement accuracy becomes little. Usually 100000 training times are enough to get the satisfied accuracy for most mixture and thermal properties ANN-reduced model.

5 Example

5.1 Establish Ann-Reduced Thermal Properties Models

We take the above six-component mixture as example and establish thermal properties models as shown in Table 1. That the structure is 8:10:6 means the number of nodes of input, hidden and output layer. The learning rate is 2.3 and the training times is 100000. Compared with rigorous thermal models, the average relative error of the established ANN-models are no more than 0.5%.

Table 1. The thermal properties ANN-reduced models

No.	Thermal Properties Model	Structure of ANN
1	Given T, P and X, calculate K	8:10:6
2	Given T, P and X, calculate liquid enthalpy	8:10:1
3	Given T, P and Y, calculate vapor enthalpy	8:10:1

5.2 Application in Process Real-Time Simulation

We use these models above to evaluate vapor-liquid equilibrium and vapor-liquid enthalpy in a dynamic distillation, which has 40 theory stages, a condenser and a reboiler, one feed and no side-stream. We compared the simulation time and accuracy of the dynamic distillation by rigorous thermal properties model (Peng-Bob method) with by ANN-reduced models, Table 2 lists the simulation time(unit: second) compared between rigorous and reduced models, and Fig.2 illustrates the accuracy compared between rigorous and reduced models.

From Table 2 we see that the simulation time can be greatly saved by ANN-reduced models, the efficiency (rigorous model/ANN-reduced models) can be

Table 2. Simulation time comparison(unit:second)

integral time	rigorous models	ANN-reduced models	Rigorous/ANN-reduced
50.0	447.97	22.46	19.94
80.0	726.99	35.75	20.31
100.0	913.46	44.65	20.46
120.0	1099.99	53.60	20.52
150.0	1379.67	66.79	20.66
180.0	1659.35	80.13	20.71
200.0	1845.88	88.98	20.74
250.0	2312.03	111.17	20.80
300.0	2778.18	133.30	20.84

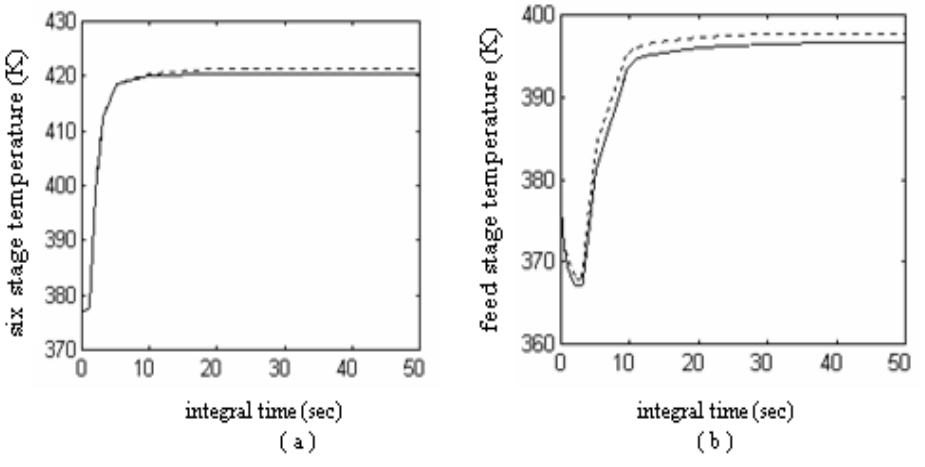


Fig. 2. the accuracy comparison between ANN-reduced and rigorous thermal properties models:(a)six stage temperature of the column (b)feed stage temperature of the column

raised by more than 20 times,which can meet the real-time demand. From Fig.2 we can see the simulation error of ANN-reduced models below 2%, this accuracy can be meet the practical demand.

6 Conclusion

We take the advantage of artificial neural network to establish reduced thermal properties models in this work. And studied the structure of ANN deeply to make the reduced models have the best performances like efficiency and accuracy. We use a three-layer structure of ANN, and study the effect of the structure to the reduced models. We find 10 nodes of the hidden layer is generally enough for the accuracy of most reduced models, the learning rate sets between 1.5 to 3.0 preferably, and 100,000 training times are enough.

From the example used in process real-time simulation, we can see the ANN-reduced thermal properties models avoid lots of iterations and test-and-errors compared with rigorous models, so which can save huge simulation time with allowable accuracy, and can be used in the practice.

Acknowledgments

We gratefully acknowledge Prof. Xiang for his guidance during the work.

References

1. Yang, Y.Q.: Practical Chemical Process Sysytem. Chemical Industry Press (1989)
2. Storen, S., Hertzberg, T.: Local Thermodynamic Models Used In Sensitivity Estimation Of Dynamic Systems. Comp. Chem. Eng. 21 (1997) S709-S714
3. Shaw, A., Doyle III, F.J., Schwaber, J.S.: A Dynamic Neural Network Approach to Nonlinear Process Modeling. Comp. Chem. Eng. 21 (1997) 371-385
4. Qu, Q., Luo, J.S., Yu, G.S., Yu, Z.H.: Application of Artificial Neural Network to Vapor Liquid Equilibrium Constant Prediction. Comp. Appl. Chem. 18(4) (2001) 383-387
5. Wei, T.Y., Huang, R.H.: Using Neural Network to Predict the Density of Saturated Liquid. Gx. Sci. 7(3) (2000) 201-205
6. Yang, X.: Distillation Dynamic Simulation. Master Paper. Qingdao University of Science and Technology (1998)

Nonlinear Identification of a PEM Fuel Cell Humidifier Using Wavelet Networks

Xian-Rui Deng

Laboratory of Complex Systems and Intelligence Science,
Institute of Automation Chinese Academy of Sciences, Beijing, China
Computer Department, Tangshan Teacher's College, Tangshan, China

Abstract. Humidification is a key factor influencing the performance of a Proton Exchange Membrane (PEM) Fuel Cell system. It is important to obtain an accurate temperature model of the humidifier to achieve the optimal humidity by means of some control strategy. Analysis shows that humidification process is nonlinear. To avoid the curse-of-dimensionality problem, a class of wavelet networks proposed by Billings was employed for the identification in this work. An efficient model term selection approach was applied to solve the high dimensional problem. The model was identified based on the experimental data acquired from our test bench. The one-step-ahead predictions and the five-step-ahead predictions were compared with the real measurements respectively. It shows that the identified model can effectively describe the real system.

1 Introduction

Proton Exchange Membrane fuel Cells, which convert chemical energy into electricity directly, have drawn much attention in the past decades. They offer a highly efficient and environmentally friendly option for energy conversion [1]. They are actively studied by scientists around the world. Recently, great progress has been made in developing PEM Fuel Cells technology. But many challenges affecting the performance of PEM Fuel Cells still exist. Humidification is one of the key issues to be extensively studied.

The polymer membranes of PEM fuel cells need to be properly hydrated in order to achieve maximum performance and extend life. Partial dehydration of the membrane decreases the net power and caused some local hot spots to emerge, which may dramatically reduce the life of the membrane. On the other hand, the performance of fuel cells will also be adversely affected by excessive water in the membranes due to the water blockage of the flow channels. In addition, the membrane humidity has to be properly controlled during transient process [2].

A great deal of effort has been taken in developing models of the humidification system. Some mathematical models have been proposed to optimize the fuel cell design to maintain high humidity [3-5]. However, the main purpose of most of these works is either to understand the system or to size the components. Few dynamic models are suitable for control purposes. Ref. [2, 6] analyzed a humidifier system and developed a theoretical model under some assumptions for simplification. This model aims to be used in humidification control and is proved effective in simulation. Yet,

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1071 – 1077, 2006.

no experiment result has been presented to demonstrate the effectiveness of the model in real test bench.

In this paper, a nonlinear MISO model using a class of wavelet networks was identified based on the experimental measurements of our fuel cell test bench. The paper begins with the description of the humidification system of a PEM Fuel Cell in Section 2. In Section 3, an overview of a class of Wavelet Networks is presented. The identification result is shown in Section 4 and a conclusion is given in Section 5.

2 PEM Fuel Cell Humidification System Description

A simplified description of a PEM Fuel Cell membrane humidifier is shown in Fig. 1. The deionized water passes on one side of a membrane. The fuel and oxidant gases are humidified by passing on the other side prior to introduction to the fuel cell stack, where the proton exchange membranes of the fuel cells will be humidified by the gases. In such arrangements, water transfers across the membrane to the fuel and oxidant gases.

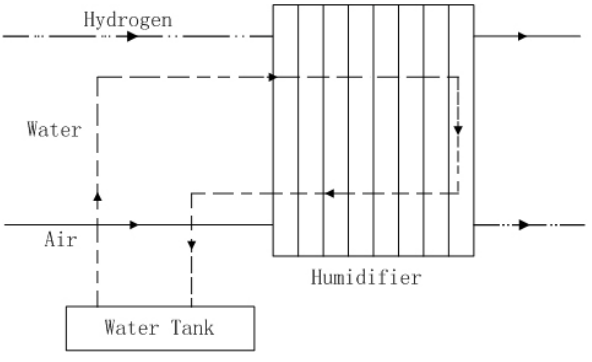


Fig. 1. The humidifier of a PEMFC

The water uptake by the gases from the humidifier is affected by the flow rates of the gases and the temperature of the humidifier. It was found that the humidity of the gas streams increases with the temperature at a constant flow rate. On the other hand, on increasing the gas flow rates at constant temperature, the absolute quantity of water picked up by the gas per hour increases [7].

The inputs of the humidifier system are chosen to be

u_1: the mass flow rate of the inlet air, l/min
u_2: the mass flow rate of the inlet Hydrogen, l/min
u_3: the temperature of the inlet water, ℃

The output of the system is chosen to be

y: the temperature of the outlet water, ℃.

A nonlinear autoregressive with exogenous inputs (NARX) model [8] can be used to describe the humidification process, which has strong nonlinearity. The identification

of it is a high dimensional problem and could lead to the curse-of-dimensionality. Therefore, a class of wavelet networks was employed to bypass the difficulty of curse-of-dimensionality.

3 A Class of Wavelet Networks [8]

Wavelet networks are inspired by both feedforward neural networks and wavelet decompositions. In recent years, they have been introduced for the identification of nonlinear static systems and nonlinear dynamical systems [9].

Let ψ be a mother wavelet, any function $g \in L^2(R)$ can be expanded as:

$$g(x) = \sum_{t \in \mathbf{\Gamma}} c_t \psi_t(x) = \sum_{t \in \mathbf{\Gamma}} c_t \psi_{(a_t, b_t)}(x) = \sum_{t \in \mathbf{\Gamma}} c_t \frac{1}{\sqrt{a_t}} \psi(\frac{x - b_t}{a_t}) \tag{1}$$

where a_t and b_t are the scale and translation parameters, and c_t are the decomposition coefficients or weights. Equation (1) is called the wavelet frame decomposition.

The decomposition (1) is often discretized by restricting the dilation and translation parameters to a dyadic lattice as $a_t = 2^{-j}$ and $b_t = k2^{-j}$ with $j, k \in Z$ (Z is the set of all integers). For the dyadic lattice case, (1) becomes

$$g(x) = \sum_j \sum_k c_{j,k} \psi_{j,k}(x) \tag{2}$$

where $\psi_{j,k}(x) = 2^{j/2} \psi(2^j x - k)$ and $j, k \in Z$.

The decompositions (1) and (2) are often truncated at an appropriate accuracy for either static function learning or dynamical system modeling. Taking the decomposition (2) as an example, an approximation to a function $g \in L^2(R)$ using the truncated wavelet decomposition with the coarsest resolution j_0 and the finest resolution $j_{\max}$ can be expressed in the following:

$$g(x) = \sum_{j=j_0}^{j_{\max}} \sum_{k \in K_j} c_{j,k} \psi_{j,k}(x) \tag{3}$$

where K_j ($j = j_0, j_0+1, ..., j_{\max}$) are subsets of Z. The truncated wavelet decomposition (3) is referred to as fixed grid wavelet networks.

Consider a nonlinear single-input-single-output (SISO) system expressed by a NARX model:

$$y(t) = f(y(t-1), \ldots, y(t-n_y), u(t-1), \ldots, u(t-n_u)) + e(t) \tag{4}$$

where f is an unknown nonlinear mapping, $u(t)$ and $y(t)$ are the sampled input and output sequences, n_u and n_y are the maximum input and output lags, respectively. The noise variable $e(t)$ is immeasurable but is assumed to be bounded and uncorrelated with the inputs.

The nonlinear function f in the model (4) can be decomposed into a number of functional components via the well-known functional ANOVA expansions [10].

$$\begin{aligned} y(t) &= f(x_1(t), x_2(t), \ldots, x_n(t)) \\ &= f_0 + \sum_{i=1}^{n} f_i(x_i(t)) + \sum_{1 \le i < j \le n} f_{ij}(x_i(t), x_j(t)) + \cdots \\ &\quad + \sum_{1 \le i_1 < \cdots < i_m \le n} f_{i_1 i_2 \ldots i_m}(x_{i_1}(t), x_{i_2}(t), \ldots, x_{i_m}(t)) \\ &\quad + \cdots + f_{12\cdots n}(x_1(t), x_2(t), \ldots, x_n(t)) + e(t) \end{aligned} \tag{5}$$

where $x(t) = [x_1(t), x_2(t), \ldots, x_n(t)]^T$ and

$$x_k(t) = \begin{cases} y(t-k), & 1 \le k \le n_y \\ u(t-k+n_y), & n_y + 1 \le k \le n = n_y + n_u \end{cases} \tag{6}$$

The first functional component f_0 is a constant to indicate the intrinsic varying trend; f_i, f_{ij},..., are univariate, bivariate, etc., functional components.

From (1), $f_{i_1 i_2 \ldots i_m}(x_{i_1}(t), x_{i_2}(t), \ldots, x_{i_m}(t))$ in the ANOVA expansion (5) can be expressed using an m-dimensional Wavelet Networks as

$$f_{i_1 i_2 \ldots i_m}(x_{i_1}(t), \ldots, x_{i_m}(t)) = \sum_{j=j_m}^{J_m} \sum_{k_1 \in K_j} \cdots \sum_{k_m \in K_j} c_{j:k_1,\ldots,k_m} \psi^{[m]}_{j:k_1,\ldots,k_m}(x_{i_1}(t), \ldots, x_{i_m}(t)) \tag{7}$$

where the m-dimensional wavelet function $\psi^{[m]}_{j:k_1,\ldots,k_m}(x_{i_1}(t), \ldots, x_{i_m}(t))$ can be generated from a scalar wavelet.

Expanding each functional component in (5) by using a radial wavelet networks, a nonlinear wavelet networks can be obtained. Assume that M candidate wavelet terms are involved in a wavelet networks. The wavelet networks can then be converted into a linear-in-the parameters form

$$y(t) = \sum_{m=1}^{M} \theta_m p_m(t) + e(t) \tag{8}$$

where $p_m(t)$ (m = 1, 2, ..., M) are regressors (model terms) produced by the dilated and translated versions of some mother wavelets. For a High-dimensional system, the model (8) may involve a great number of model terms. Experience shows that often many of the model terms are redundant and can be removed from the model with a given accuracy. Therefore, there exists an integer M_0 (generally $M_0 << M$), such that the model

$$y(t) = \sum_{k=1}^{M_0} \theta_{i_k} p_{i_k}(t) + e(t) \tag{9}$$

provides a satisfactory representation over the range considered for the measured input-output data. A fast and efficient model structure determination approach has been implemented using the forward OLS algorithm [8, 11].

4 Identification Result

The identification scheme is demonstrated in Fig. 2, where U(t) = [$u_1(t)$, $u_2(t)$, $u_3(t)$]' is an input vector; $y(t)$ is the output of the system; $y_e(t)$ is the output of the wavelet networks model.

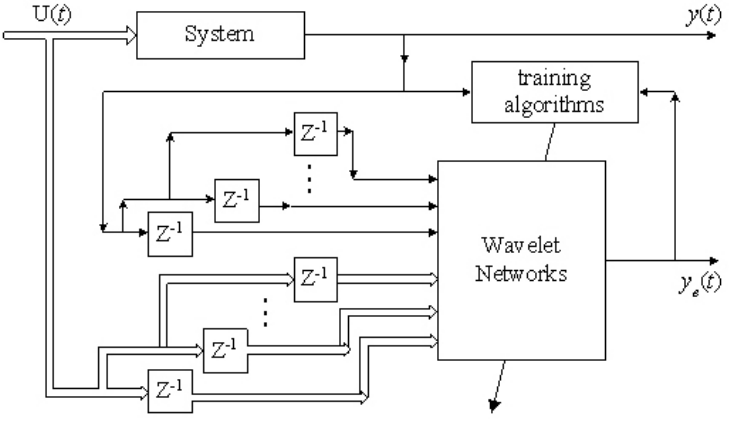


Fig. 2. The Identification scheme using wavelet networks

The data used for identification were obtained from our test bench. The sample time is 0.1s, and 2500 samples were acquired. They were divided into two parts. The first 1000 data points were used for wavelet networks training and the remaining 1500 data points were used for model testing.

The model order was determined first. Then, significant variables were selected from all candidate lagged output and input variables. Mexican hat radial wavelet networks were used for identification. The coarsest resolution was set to be $j_0 = 0$ and finest resolution was set to be $j_{\max} = 3$.

The one-step-ahead predictions and five-step-ahead predictions of the identified wavelet networks model were compared with the measurements respectively. The comparisons are shown in Fig. 3 and Fig. 4.

The following index, the mean-square-error on the test set, was used to measure the performance of the identified wavelet networks

$$\overline{E} = \frac{\sum_{k=1}^{N_{test}} \left| x_k - \hat{x}_k \right|^2}{\sum_{k=1}^{N_{test}} \left| x_k - \overline{x}_k \right|^2} \tag{10}$$

where N_{test} is the length of the test set, x_k and $\hat{x}_k$ are the measurements over the data set and associated predictions, and $\overline{x} = (1/N_{test}) \sum_{k=1}^{N_{test}} x_k$. In one-step-ahead predictions, it was calculated that $\overline{E} = 0.0019039$. In five-step-ahead predictions, it was calculated that $\overline{E} = 0.0072915$.

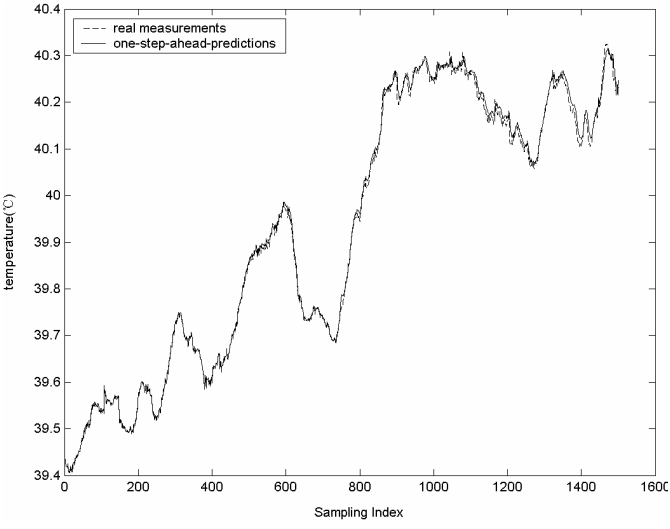


Fig. 3. Comparison between one-step-ahead predictions and the real measurements

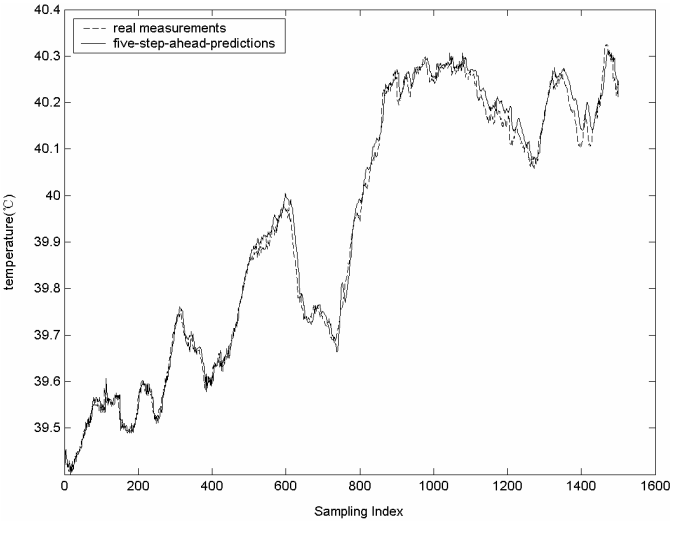


Fig. 4. Comparison between five-step-ahead predictions and the real measurements

5 Conclusion

In this paper, a nonlinear MISO model of a PEM Fuel Cell humidifier was identified by using a class of Wavelet Networks, which can bypass the difficulty of the curse-of-dimensionality. The identification is based on the experimental measurements obtained from our fuel cell test bench. One-step-ahead predictions and five-step-ahead

predictions were compared with the real measurements respectively. The comparisons demonstrate the effectiveness of the identified model.

References

1. Farooque, M., Maru, H.C.: Fuel Cells-the Clean and Efficient Power Generators. In: Proceedings of the IEEE, Vol. 89. (2001) 1819–1829
2. Chen, D.-M., Peng, H.: Modeling and Simulation of a PEM Fuel Cell Humidification System. In: Proceedings of 2004 American Control Conference, Vol. 1, Boston, Massachusetts (2004) 822 – 827
3. Ceraolo, M., Miuli, C. Pozio, A.: Modelling Static and Dynamic Behavior of Proton Exchange Membrane Fuel Cells on the Basis of Electro-chemical Description. J. Power Sources 113 (2003) 131-144
4. Janssen, G.J.M., Overvelde, M.L.J.: Water Transport in the Proton-Exchange-Membrane Fuel Cell: Measurements of the Effective Drag Coefficient. J. Power Sources 101 (2001) 117–125
5. Nguyen, T.V., White, R.E.: A Water and Heat Management Model for Proton-Exchange-Membrane Fuel Cells. J. Electrochem. Soc. 140 (1993) 2178–2186
6. Chen, D.-M., Peng, H.: Analysis of Non-Minimum Phase Behavior of PEM Fuel Cell Membrane Humidification Systems. In: Proceedings of 2005 American Control Conference, Vol. 6, (2005) 3853–3858
7. Sridhar, P., Perumal, R., Rajalakshmi, N., Raja, M., Dhathathreyan, K.S.: Humidification Studies on Polymer Electrolyte Membrane Fuel Cell. J. Power Sources 101 (2001) 72–78
8. Billings, S. A., Wei, H.-L.: A New Class of Wavelet Networks for Nonlinear System Identification. IEEE Trans. Neural Netw. 16 (2005) 862–874
9. Zhang, Q. H., Benveniste, A.: Wavelet Networks. IEEE Trans. Neural Netw. 3 (1992), 889–898
10. Friedman, J.H., Multivariate Adaptive Regression Splines, Ann. Statist. 19 (1991), 1–67
11. Wei, H.-L., Billings, S.A., and Liu, J.: Term and Variable Selection for Nonlinear System Identification. Int. J. Control 77 (2004) 86–110
12. Billings, S. A., Chen, S., and Korenberg, M. J.: Identification of MIMO Nonlinear Systems Using a Forward Regression Orthogonal Estimator. Int. J. Control. 49 (1989) 2157–2189
13. Chen, S., Billings, S.A., Luo, W.: Orthogonal Least Squares Methods and Their Application to Nonlinear System Identification. Int. J. Control 50 (1989) 1873–1896

Application of RBF Neural Networks Based on a New Hybrid Optimization Algorithm in Flotation Process

Yong Zhang and Jie-Sheng Wang

Anshan University of Science & Technology, Anshan,
Liaoning 114044, P.R. China
{zy9091, wjs_1977720}@163.com

Abstract. An inferential estimation strategy of quality indexes of flotation process based on principal component analysis (PCA) and radial basis function neural network (RBFNN) is proposed. Firstly, the process prior knowledge and PCA method are used to simplify the networks' input dimension and to choose the secondary variables. Then a new hybrid optimization algorithm of RBFNN is developed. The algorithm includes simplified rival penalized competitive learning method (SRPCL) to make an adaptive clustering of networks' input pattern and recursive least squares method (LSM) with forgetting factor to update networks' weights. The simulation results show that this inference estimation strategy has high predictive accuracy in flotation process.

1 Introduction

Flotation is a method to choose and separate minerals by the buoyancy of bubble from mine slurry according to different surface characters of minerals [1]. The quality indexes (extractive ore grade and flotation callback ratio) are important in flotation process. But they can only be controlled based on the off-line analysis results in reality. Its long delay and low precision make it difficult to realize direct quality closed-loop control of flotation process. In addition, to build up the soft-sensing model of flotation process based on mechanism analysis or experimental method isn't feasible in industry. The inferential estimation of extractive ore grade and flotation callback ratio in flotation process is in urgent demand and actual research in this field is few.

Therefore the paper proposes an inferential estimation strategy of flotation quality indexes based on PCA method and RBFNN. Firstly, primary process variables are obtained by technique mechanism and experience knowledge. Then the algorithm selects secondary variables to reduce the input dimension of RBFNN using PCA method. At the same time, in order to enhance precision and real-time characteristic of RBFNN recursive training, the paper proposes a new hybrid recursive algorithm, which includes SRPCL to make an adaptive clustering of networks' input pattern and the recursive LSM with forgetting factor to update the networks' weights. Finally, the simulation results obtained from the actual prediction demonstrate that the performance and capability of the proposed strategy are superior.

The paper is organized as follows. In section 2, the PCA of flotation process is introduced. The new hybrid algorithm of RBFNN is presented in section 3. In section 4, simulation outcomes are summarized. The conclusion illustrates the last part.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1078 – 1084, 2006.

2 Dimensionality Reduction Using PCA

PCA method was originally developed in the 1900s and has now re-emerged as an important technique in data analysis [2]. Multiple regression and discrimination analysis using variable selection procedures to reduce the dimension result in the loss of one or more important dimensions. The PCA approach uses all of the original variables to obtain a smaller set of new variables (principal components-PCs), which can be used to approximate the original variables. PCs are uncorrelated and are ordered so that the first few retain most of the variation present in the original set.

A data matrix X represents n observations of each of m variables. PCA is scale dependent, and so the data matrix X must be scaled in some meaningful way before PCA analysis in order to avoid the influence caused by different dimensions of process variables to predictive outcome. The mean vector and standard difference vector of X are μ and σ, receptively. Then the scaling of process variable $\tilde{X}$ is given as:

$$x_{ij} = (x_{ij} - \mu_j)/\sigma_j \quad (\mathrm{i} = 1,\cdots,\mathrm{n},\ \mathrm{j} = 1,\cdots,\mathrm{m}) \ . \tag{1}$$

The covariance matrix of $\tilde{X}$ is $\sum$. The PCA procedure is:

1. Calculate the m eigenvalues $\lambda_j (\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m)$ and corresponding unit orthogonal eigenvector P_j of covariance matrix $\sum$.
2. Calculate the j th principal component t_j:

$$t_j = \tilde{X} P_j \ . \tag{2}$$

3. Calculate principal component model:

$$\hat{X} = t_1 P_1^T + t_2 P_2^T + \cdots + t_j P_j^T \ . \tag{3}$$

where $t_j \in R^n$ is scoring vector or principal component, $P_j \in R^m$ is loading vector.

The first few eigenvectors are the PCs that can capture most of the variance of the original data while the remaining PCs mainly represent noise in the data. $T = [t_1,\cdots,t_m]$ is scoring matrix and $P = [P_1,\cdots,P_m]$ is loading matrix. Then the formula 3 may be described as matrix form:

$$\hat{X} = TP^T \ . \tag{4}$$

4. Calculate variance contributed ratio η_k of the forward $k(1 \leq k \leq m)$ PCs:

$$\eta_k = \sum_{i=1}^{k} \lambda_i \Big/ \sum_{i=1}^{m} \lambda_i \ . \tag{5}$$

Extractive ore grade and callback ratio of flotation process is directly influenced by mineral slurry thickness, mineral properties and additive medicament. Based on technique mechanism and experience knowledge the paper selects five process variables as the input of PCA model, which are feed grade, feed thickness, feed granularity, medicament flow rate and mineral slurry PH value. Using these variables for RBFNN modeling will make network topology and training complex. Therefore, the paper

uses PCA method to analysis input historical data matrix. The variance represented by each PC is given in Table 1. Three principal components can express data change above 85%. After the PCA disposal, the PCs variables as the RBFNN model input not only retain the original system characteristic information, but also simplify the RBFNN structure. Therefore many merits of the RBFNN can be fully displayed.

Table 1. Summary of the PCs

PCs	Eigenvalues	Percentage of eigenvalues (%)	Cumulated percentage (%)
1	1.1109	37.73	37.73
2	1.0430	35.42	73015
3	0.4108	13.95	87.10
4	0.2480	8.42	95.53
5	0.1318	4.47	100

3 New Hybrid Recursive Algorithm of RBFNN

In Fig.1. a three-layer RBFNN structure is illustrated:

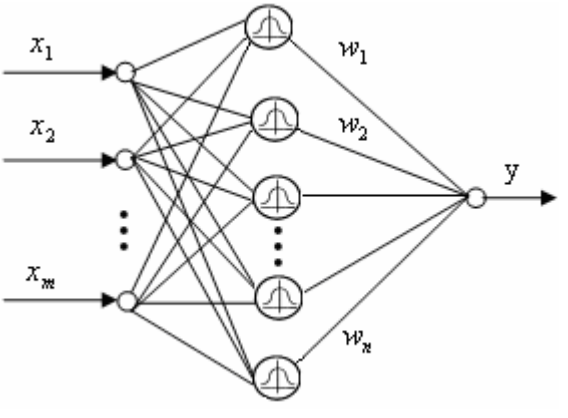


Fig. 1. Architecture of RBFNN

Structurally, RBFNN is composed of receptive units (neurons) which act as the operators providing the information about the class to which the input signal belongs to [3]. Each neuron in the hidden layer provides a degree of membership value for the input pattern with respect to the basis vector of the receptive unit itself. The output layer is comprised of linear combiners. Suppose there are m input nodes, n hidden nodes and one output node. Then RBFNN may be represented as:

$$y(x) = \sum_{i=1}^{n} w_i \cdot g\left(\left\| x - c_i \right\|_B \right) . \tag{6}$$

$$\|x-c_i\|_B^{\ 2} = (x-c_i)^T B^T B(x-c_i) \ . \tag{7}$$

where $x \in R^m$ is network input, g(·) is a radial basis function, $c_i \in R^m (1 \le i \le n)$ is the RBF center of, w_i is the network connective value and B is the distance addition matrix of the Euclidean norm. The common RBF is Gauss function:

$$g(x) = \exp[-\frac{\|x-c_i\|^2}{2\sigma_i^{\ 2}}], i = 1,2,\cdots,n \ . \tag{8}$$

The new hybrid recursive algorithm of RBFNN proposed in the paper includes:

1. Revise network center using a new self-tuning clustering algorithm-SRPCL.
2. Revise network connective values using the recursion LSM with forgetting factor.

3.1 Simplified Rival Penalized Competitive Learning Method (SRPCL)

The K-means clustering is a classical algorithm that determines the RBF centers. The main drawback is that network performance may be seriously reduced when the number of hidden nodes doesn't match with the number of input patterns. Therefore, the paper adopts SRPCL to train the RBFNN centers [4]. For each input vector, SRPCL adjusts the center of the winner node to make it near the input vector and carries out counter-study with small learning rate at the center of inferior winner node to make it deviate from the input vector. The algorithm is showed as follows:

1. Given the stochastic initial center $c_i(0)(1 \le i \le n)$, initial forward learning rate α and reverse learning rate $\alpha_r (0 < \alpha_r \le \alpha < 1)$.
2. Calculate the distance of the step k:

$$d_i(k) = \|x(k) - c_i(k-1)\|_B, 1 \le i \le n \ . \tag{9}$$

where $d_i(k)$ is the Euclidean norm.

3. Calculate the minimum distance and the inferior small distance of the step k:

$$\begin{aligned} d_v(k) &= \min_{1 \le i \le n} |d_i(k)| \\ d_l(k) &= \min_{1 \le i \le n} |d_i(k)|, l \ne v \end{aligned} \ . \tag{10}$$

4. Adjust the center:

$$c_v(k) = c_v(k-1) + \alpha[x(k) - c_v(k-1)], i = v \ . \tag{11}$$

$$c_l(k) = c_l(k-1) - \alpha_r[x(k) - c_l(k-1)], i = l \ . \tag{12}$$

$$c_i(k) = c_i(k-1), 1 \le i \le n, i \ne v, i \ne l \ . \tag{13}$$

5. Calculate the distance between the node v and node l over again:

$$d_v(k) = \|x(k) - c_v(k)\|_B \ . \tag{14}$$

$$d_l(k) = \left\| x(k) - c_l(k) \right\|_B \ . \tag{15}$$

6. $k = k + 1$, Return 2.

3.2 Recursive Least Squares Method with Forgetting Factor

Transitional recursion least squares method will appear the phenomenon of losing the revision ability gradually with the increase of data quantity. Therefore, the paper adopts the recursion LSM with forgetting factor to train the network connective weights. The method can effectively overcome this drawback and has quicker training speed and stronger time-vary tracking capacity. Given the relation from the hidden layer to the output level can be expressed as follows:

$$y(k) = \sum_{i=1}^{n} w_i(k) \times Z_i(k) \ . \tag{16}$$

where $Z_i(k)$ is the i th node output of hidden level and $w_i(k)$ is the connective weight of the i th node of hidden level and output layer. The algorithm procedure can be represented as:

1. Given the initial weight vector $w(0)$, the initial training sample converse correlation matrix $P(0)$ and forgetting factor.
2. Calculate the gain matrix $K(k)$ and $P(k)$

$$K(k) = P(k-1)Z(k)[Z^T(k)P(k-1)Z(k) + \mu]^{-1} \ . \tag{17}$$

$$P(k) = \frac{1}{\mu}[I - K(k)Z^T(k)]p(k-1) \ . \tag{18}$$

3. Adjust the weight vector $w(k)$:

$$w(k) = w(k-1) + K(k)[Y_d(k) - Z^T(k)w(k-1)] \ . \tag{19}$$

where $Y_d(k)$ is the k th output target of RBFNN.

4 Simulations

We collect 700 representative historical data of flotation process in 2004. Partial data is selected stochastically as training data to optimize RBFNN centers and weights utilizing the proposed algorithm. The rest data are used for testing the performance of soft-sensor model. Simulation parameters are initialized as follows: The node number of input level, hidden level and output level are 3, 8 and 2, respectively. Distance weighted matrix $B = diag(1.0,1.1,1.0)$. Learning speed α is 0.1 and α_τ is 0.01. Initial center $c_i(0)$ and vector $w(0)$ are chosen as random numbers in range [0, 1]. $P(0) = 10I$ (I is 9×9 unit matrix). Forgetting factor μ is 0.75. Fig. 2-5 is the predictive output and error of extractive ore grade and callback ratio, receptively.

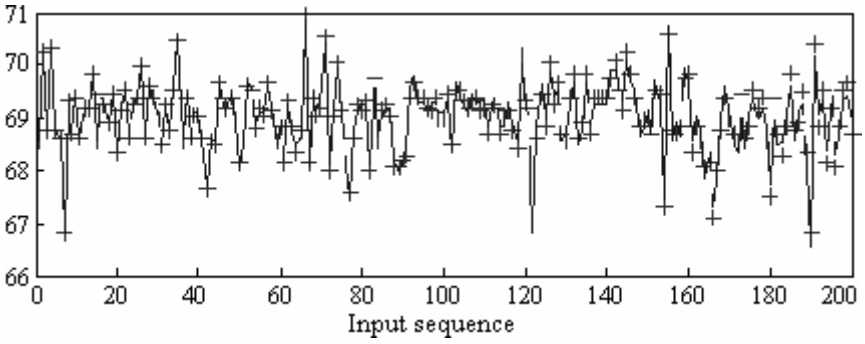


Fig. 2. Prediction output of extractive ore grade (%)

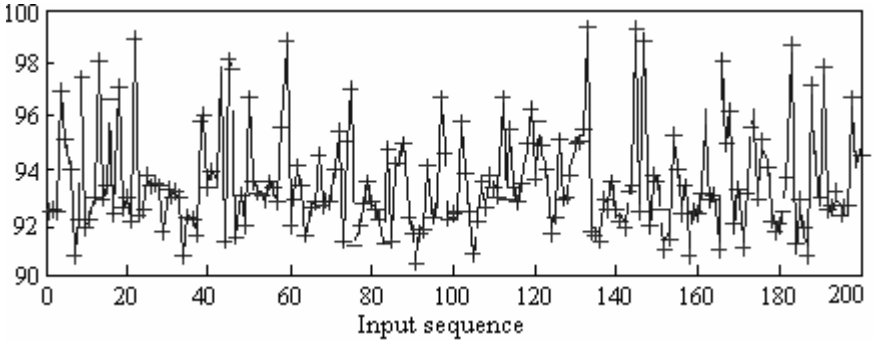


Fig. 3. Prediction output of flotation callback ratio (%)

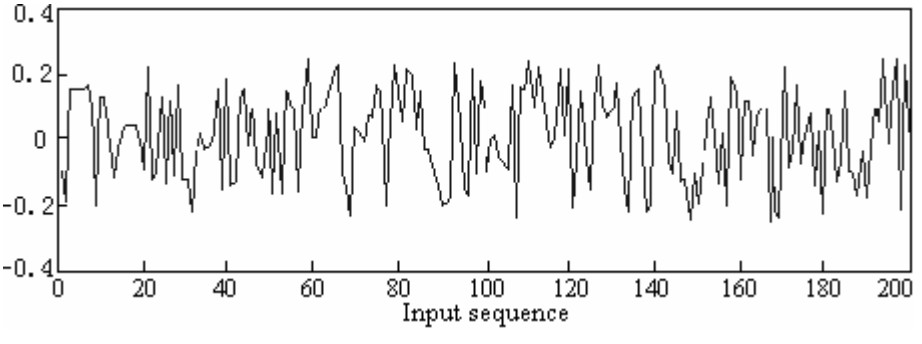


Fig. 4. Prediction error of extractive ore grade (%)

From upper figures, the predictive error of extractive ore grade and flotation callback ratio is in the range [-0.3% 0.3%] and [-0.1% 0.1%], respectively. Simulation result indicates that the predictive precision of the model is high. The medicament additive system based on the predictive model can effectively stabilize flotation process and enhance flotation callback ratio.

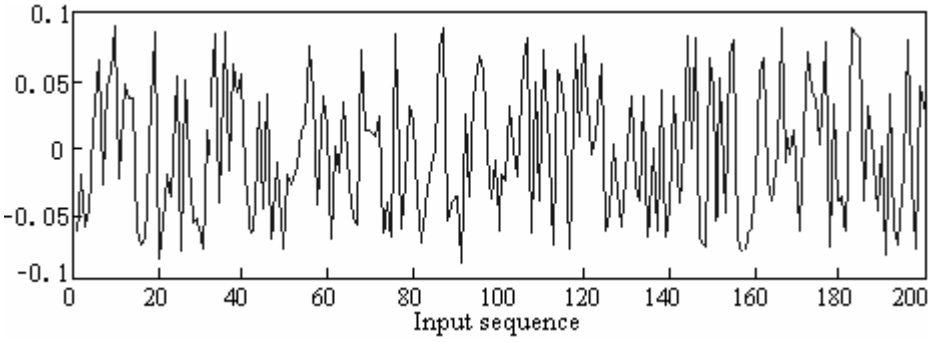


Fig. 5. Prediction error of flotation callback ratio (%)

5 Conclusions

This paper proposes an inferential estimation strategy of quality indexes of flotation process based on the principal component analysis method and the RBF neural network. A new hybrid optimization algorithm combining SRPCL method and the recursion least squares method with forgetting factor is described in this paper. The simulation indicates that this method has a high estimation precision and completely satisfies the request of the optimized control of flotation process. This conclusion is not only suitable for this system, but also suitable for other production processes, such as chemical reaction process, petroleum refinement process and papermaking process.

References

1. Zhang, Y., He, H.J., Wang, W.: Research on Technology for Iron Increasing and Silicon Reduction. Mining and Metallurgical Engineering 23(1) (2003) 34-37
2. Ipek, H., Ankara, H.: The Application of Statistical Process Control. Minerals Engineering 12(7) (1999) 827-835
3. Wang, X.D., Shao, H.H.: The Theory of RBF Neural Network and Its Application in Control. Information and Control 26(4) (1997) 272-284
4. Wang, X.F., Wan, Z.Q., Song, W.Z.: A New Hybrid Recursive Learning Algorithm for Radial Basis Function Neural Networks. Control Theory and Application 15(2) (1998) 272-276

Estimation of Some Crucial Variables in Erythromycin Fermentation Process Based on ANN Left-Inversion

Xianzhong Dai, Wancheng Wang, and Yuhan Ding

Department of Automation, Southeast University, Nanjing, 210096, China
xzdai@seu.edu.cn

Abstract. For the on-line estimation of some directly immeasurable crucial variables in erythromycin fermentation process, this paper presents an Artificial Neural Network (ANN) left-inversion based on the "assumed inherent sensor" and its left-inversion concepts. The ANN left-inversion is composed of two relatively independent parts — a static ANN used to approximate the complex nonlinear function and several differentiators used to represent its dynamic behaviors, so that the ANN left-inversion is a special kind of dynamic ANN in essence. Different from common dynamic ANNs, such a separate structure makes the ANN left-inversion easier to use, hence facilitating its application. The ANN left-inversion has been used to estimate such immeasurable variables as mycelia concentration, sugar concentration and chemical potency in erythromycin fermentation process. The experimental results show its validity.

1 Introduction

The use of artificial neural network (ANN) as a soft sensor to estimate immeasurable variables in biochemical process has received a great deal of attention [1],[2],[3] owing to its excellent ability to approximate a nonlinear function with satisfactory accuracy. However, most of the existing ANN soft sensors are empirical models [2],[3], which are generally obtained by using a static or dynamic ANN to approximate the relation between the inputs and outputs on the basis of empirical rules rather than strict theory. In general, it remains difficult to obtain a satisfactory ANN soft sensor only using empirical methods.

Motivated by this, this paper proposes a strict ANN soft sensor with the help of inversion theory to estimate the crucial immeasurable variables in erythromycin fermentation process. To achieve this, we first present the "assumed inherent sensor" and its left-inversion concepts for the erythromycin fermentation process, and then a static ANN is used to overcome the difficulty in constructing the left-inversion in an analytic manner, thus leading to an ANN left-inversion (i.e. the desired ANN soft sensor). Besides a static ANN, the ANN left-inversion also feathers a set of differentiators that describe its dynamic behaviors so that it is a special kind of dynamic ANN in essence.

Compared with the common dynamic ANN like Hopfield Neural Network, the ANN left-inversion has a simpler structure and its dynamic characteristics are easy to capture. In addition, owing to the fact that the research object under consideration in this paper is a partially known system rather than a complete black-box, the ANN left-inversion is a preferred option to implement the desired ANN soft sensor.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1085 – 1090, 2006.

2 Erythromycin Fermentation Process and Estimating Principle Based on Left-Inversion

For such a complex biochemical process as erythromycin fermentation, it is very difficult to establish its accurate mathematical model. As a compromise, the following gray-box model based on its partial knowledge [4], [5] is considered in this paper.

$$\begin{cases} \dot{x}_1 = \mu x_1 - \dfrac{x_1}{x_6}\left(\sum\limits_{i=1}^{5} u_i\right) = f_1(\boldsymbol{x},\boldsymbol{u}) \\ \dot{x}_2 = \sigma x_1 + \dfrac{s_1}{x_6}u_2 - \dfrac{x_2}{x_6}\left(\sum\limits_{i=1}^{5} u_i\right) = f_2(\boldsymbol{x},\boldsymbol{u}) \\ \dot{x}_3 = \pi x_1 - s_2 x_3 + \dfrac{s_3}{x_6}u_3 - \dfrac{x_3}{x_6}\left(\sum\limits_{i=1}^{5} u_i\right) = f_3(\boldsymbol{x},\boldsymbol{u}) \\ \dot{x}_4 = \eta x_1 - s_4 x_4 - \dfrac{x_4}{x_6}\left(\sum\limits_{i=1}^{5} u_i\right) = f_4(\boldsymbol{x},\boldsymbol{u}) \\ \dot{x}_5 = \psi x_1 + \dfrac{s_5 u_4 - s_6 u_1 - s_7 u_2 - s_8 u_3}{x_6} - \dfrac{x_5}{x_6}\left(\sum\limits_{i=1}^{5} u_i\right) = f_5(\boldsymbol{x},\boldsymbol{u}) \\ \dot{x}_6 = u_1 + u_2 + u_3 + u_4 + u_5 = f_6(\boldsymbol{x},\boldsymbol{u}) \end{cases}, \tag{1}$$

where $\boldsymbol{u} = (u_1, u_2, u_3, u_4, u_5)^T$ are inputs (oil u_1, dextrin u_2, propanol u_3, aqua ammonia u_4, and water u_5), and $\boldsymbol{x} = (x_1, x_2, x_3, x_4, x_5, x_6)^T$ are states which can be divided into two groups: the measurable group (dissolved oxygen concentration x_4, pH value x_5 and zymotic fluid volume x_6) and the immeasurable group (mycelia concentration x_1, sugar concentration x_2 and chemical potency x_3). In addition, $\mu(\boldsymbol{x}), \sigma(\boldsymbol{x}), \pi(\boldsymbol{x}), \eta(\boldsymbol{x}), \psi(\boldsymbol{x})$ are all analytic functions and s_i, $i = 1, \cdots, 8$ are all non-zero constants, which imply that all the functions $f_i(\boldsymbol{x},\boldsymbol{u})$, $i = 1, \cdots, 6$ are still analytic.

For the erythromycin fermentation process (1), to estimate x_1, x_2, x_3, one may

① first, assume that in the erythromycin fermentation process (1), there exists a subsystem whose inputs just are the to-be-estimated x_1, x_2, x_3 while whose outputs are the measurable variables x_4, x_5, x_6. Such a subsystem would be viewed as an "assumed inherent sensor" (see Fig. 1), with the $u_1 \sim u_5$ as parameter variables;

② then, under the condition that the left-inversion of the "assumed inherent sensor" exists, cascading the left-inversion with the "assumed inherent sensor" can lead to a composite identity system (see Fig. 2). Thus, the outputs of the left-inversion can completely reproduce the inputs to the "assumed inherent sensor" so that the problem of estimating the to-be-estimated variables x_1, x_2, x_3 can be solved.

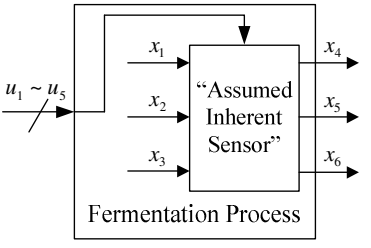


Fig. 1. The "assumed inherent sensor" in erythromycin fermentation process

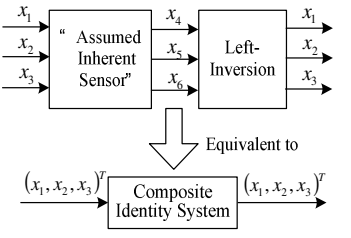


Fig. 2. Estimating principle based on left-inversion of "assumed inherent sensor"

3 The "Assumed Inherent Sensor" and Its ANN Left-Inversion

Denote the to-be-estimated variables as $\hat{\boldsymbol{x}} = (x_1, x_2, x_3)^T$ and the measurable ones as $\boldsymbol{z} = (z_1, z_2, z_3)^T = (x_4, x_5, x_6)^T$. Then, the "assumed inherent sensor" can be achieved by coupling the second order derivative of z_1 with the forth, fifth equations in (1), i.e.

$$\begin{cases} \ddot{z}_1 = \ddot{z}_1(\boldsymbol{x}, \boldsymbol{u}, \dot{\boldsymbol{u}}) \\ \dot{z}_1 = \dot{x}_4 = \eta x_1 - s_4 x_4 - \dfrac{x_4}{x_6}(\sum_{i=1}^{5} u_i) \\ \dot{z}_2 = \dot{x}_5 = \psi x_1 + \dfrac{s_5 u_4 - s_6 u_1 - s_7 u_2 - s_8 u_3}{x_6} - \dfrac{x_5}{x_6}(\sum_{i=1}^{5} u_i) \end{cases}, \tag{2}$$

According to the inversion theory [6],[7],[8], if $det\left(\partial(\ddot{z}_1, \dot{z}_1, \dot{z}_2)^T / \partial(x_1, x_2, x_3)\right) \neq 0$ holds in the operating range, the "assumed inherent sensor" (2) will be left-invertible and its left-inversion can be expressed by the following inverse function of (2)

$$\hat{\boldsymbol{x}} = \boldsymbol{\varphi}(z_1, z_2, z_3, \dot{z}_1, \ddot{z}_1, \dot{z}_2, \boldsymbol{u}, \dot{\boldsymbol{u}}) \tag{3}$$

In general, it is very difficult to verify such a left invertibility condition. As an alternative, in the practical use we can first suppose the "assumed inherent sensor" is left-invertible in its operating range, and then examine its validity and credibility via the experimental results. In section 5, we will find that the field experimental results show that such a supposition is reasonable and credible.

Obviously, it is hard to implement the left-inversion (3) by analytic means due to high inaccuracy and nonlinearity of the erythromycin fermentation process. To overcome this problem, a static ANN is adopted to approximate the static nonlinear function $\boldsymbol{\varphi}(\square)$ appearing in (3) by taking advantage of its strong potential to approximate a nonlinear function, thus resulting in an ANN left-inversion.

The static ANN adopts Back Propagation (BP) network due to its simple structure. It has a structure of 16-22-15-1 with "tan sigmoid" transfer function on the nodes of

hidden layers, and "linear" transfer function on that of output layer. Besides a static ANN, the ANN left-inversion also feathers a set of differentiators that describe its dynamic behaviors (see Fig.3) so that it is a kind of dynamic ANN in essence.

As opposed to the common dynamic ANNs like the well-know Hopfield Neural Network which is composed of dynamic neurons, the ANN left-inversion features two relatively independent parts — a set of differentiators and a static ANN. This separate structure makes the ANN left-inversion easy to use and enables it to approximate a complex dynamic system with satisfactory accuracy [9].

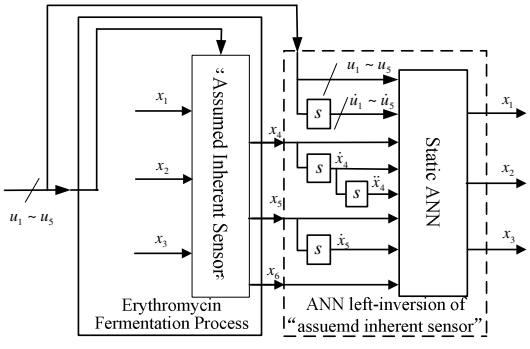


Fig. 3. ANN left-inversion of the "assumed inherent sensor" (*s* denotes a differentiator)

4 ANN's Training

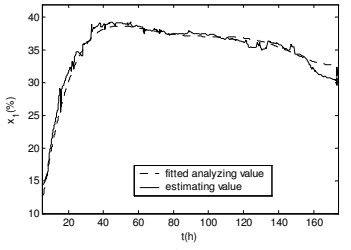
Before training the ANN left-inversion, we first need to collect the raw data to form the training data sets. For the erythromycin fermentation process, the raw data x_4, x_5, x_6, $u_1 \sim u_5$ are acquired by using actual chemical and physical sensors every 5 minutes while the x_1, x_2, x_3 are obtained by using off-line analyzers every 6 hours. The first- and second-order derivatives of x_4, x_5, x_6 appearing in ANN left-inversion are obtained by using 5-point derivative method that can guarantee high accuracy. Thus, we obtain the ultimate ANN training data sets.

The ANN was trained with Levenberg-Marquardt training algorithm due to its faster convergence than common BP algorithms. In addition, the trained ANN left-inversion was tested with the data not used for training. The test results showed that the generalization of the ANN left-inversion is appropriate for actual application.

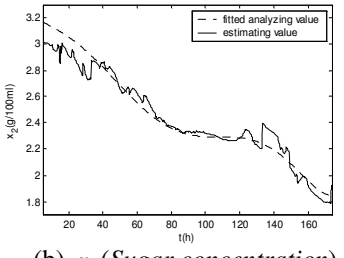
5 Experimental Results

A typical batch of estimating results are shown in Fig. 4 where the real lines represent the real-time on-line estimating values denoted by x_1^s, x_2^s, x_3^s and the broken lines represent the fitted off-line analyzing values denoted by x_1^a, x_2^a, x_3^a. For making a clear

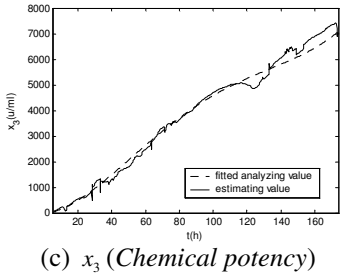
comparison, view the off-line analyzing values as the true ones and then define the relative errors as $\Delta_r x_i = (x_i^s - x_i^a)/x_i^a, i = 1,2,3$ that are shown in Fig. 4.



(a) x_1 (*Mycelia concentration*)

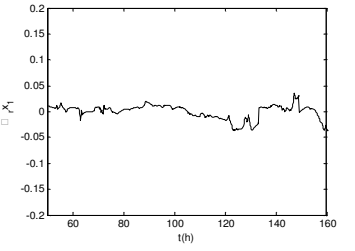


(b) x_2 (*Sugar concentration*)

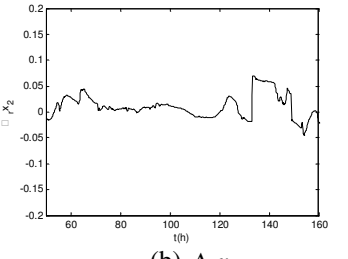


(c) x_3 (*Chemical potency*)

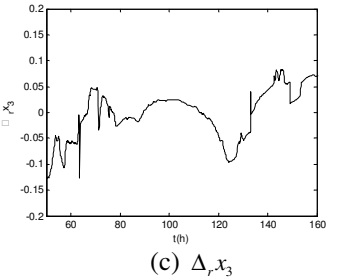
Fig. 4. The estimating results



(a) $\Delta_r x_1$



(b) $\Delta_r x_2$



(c) $\Delta_r x_3$

Fig. 5. The relative errors

Fig. 5 shows that the relative errors $\Delta_r x_1, \Delta_r x_2, \Delta_r x_3$ are all less than 12% during the main stage of the fermentation process (from 50th to 160th hour). To be precise, the maximum relative error of mycelia concentration is 8% and those of sugar concentration and chemical potency are 10% and 12% respectively.

6 Conclusion

In this paper, an ANN left-inversion of the so-called "assumed inherent sensor" is presented for on-line estimation of some crucial immeasurable variables in erythromycin fermentation process. The proposed ANN left-inversion has been used to

estimate such immeasurable variables as the mycelia concentration, sugar concentration, and chemical potency and the field experimental results show its validity.

Compared with the off-line laboratory analyses every 6 hours, the ANN left-inversion is capable of providing more information timely, hence making it possible to achieve the real-time control of the erythromycin fermentation process. This is an open topic for further research.

Acknowledgments

This work is supported by both the National Natural Science Foundation of China (60574097, 60174004) and the Specialized Research Fund for the Doctoral Program of Higher Education (20050286029).

References

1. Shi, D., Zhang, H., Yang, L.: Time-Delay Neural Network for the Prediction of Carbonation Tower's Temperature. IEEE Trans. Instrumentation and Measurement 52(4) (2003) 1125-1128
2. Wang, W., Ren, M.: Soft-Sensing Method for Wastewater Treatment Based on BP Neural Network. In: Xi, Y., Cao, X., Guo, L. (eds.): Proceedings of the World Congress on Intelligent Control and Automation, Vol. 3. Shanghai, P.R. China (2002) 2330-2332
3. Su, H., Fan, L., Schlup, J.R.: Monitoring the Process of Curing of Epoxy/Graphite Fiber Composites with a Recurrent Neural Network as a Soft Sensor. Engineering Applications of Artificial Intelligence 11(2) (1998) 293-306
4. Paul, G.C., Syddall, M.T., Kent, C.A., Thomas, C.R.: A Structured Model for Penicillin Production on Mixed Substrates. Biochemical Engineering Journal 2(1) (1998) 11-21
5. Birol, G., Undey, C., Cinar, A.: A Modular Simulation Package for Fed-Batch Fermentation: Penicillin Production. Computers and Chemical Engineering 26(11) (2002) 1553-1565
6. Singh, S.N.: A Modified Algorithm for Invertibility in Nonlinear Systems. IEEE Trans. Automatic Control 26(2) (1981) 595-598
7. Dai, X., He, D., Zhang, X.: MIMO System Invertibility and Decoupling Control Strategies Based on ANN α th-Order Inversion. IEE Proceedings-Control Theory and Applications 148(2) (2001) 125-136
8. Benedetto, M.D.D., Glumineau, A., Moog, C.H.: The Nonlinear Interactor and Its Application to Input-Output Decoupling. IEEE Trans. Automatic Control 39(6) (1994) 1246-1250
9. Dai, X., Liu, J., Feng, C., He, D.: Neural Network α -th Order Inverse System Method for the Control of Nonlinear Continuous Systems. IEE Proceedings-Control Theory and Applications 145(6) (1998) 519-522

The Control of Membrane Thickness in PECVD Process Utilizing a Rule Extraction Technique of Neural Networks

Ming Chang[1], Jen-Cheng Chen[2], and Jia-Sheng Heh[3]

[1] Department of Mechanical Engineering, Chung Yuan Christian University
ming@cycu.edu.tw
[2] Department of Electronic Engineering, Chung Yuan Christian University
clement1972@gmail.comw
[3] Department of Information and Computer Engineering, Chung Yuan Christian University
jsheh@ice.cycu.edu.tw

Abstract. The principal object of this paper is to develop a neural network model, which can simulate the plasma enhanced chemical vapor deposition (PECVD) process in TFT-Array procedure. Then the Boolean logic rules are extracted from the trained neural network in order to establish a knowledge base of expert system. The input data of neural network was collected form the process parameters of PECVD machines in the TFT-Array department, included the flow rate of all gases, pressure and temperature of the chamber, etc. After checking, explaining and integrating the extraction rules into knowledge base, the rules can be the basics of membrane thickness prediction and alarm diagnosis in PECVD system.

1 Introduction

The semi-conductor and TFT-LCD are the two most important high-technology industrials currently. The manufacturing process in the array stage of TFT-LCD and the front end of semi-conductor are similar, with only the substrate material different. The major process includes deposition, lithography, etching, stripping and inspection. The R&D results in these two products are compatible and can be applied each other. Because these processes are usually operated with complex and nonlinear reactions, and the process parameters always drifting and variating [1], advance process control (APC) of parameters is becoming increasingly important in these two industrials.

In TFT array process, film deposition caused by CVD is very fundamental and important. A hybrid intelligence system for the purpose of APC of CVD system by combining the expert system with neural network [2, 3] techniques is presented. The main structure is the expert system, but the expert knowledge library and the initiative learning mechanism is achieved by a neural network system with the on-line manufacturing process data. An artificial neural network (ANN) has excellent learning ability via grouping input data into clusters. The knowledge of ANN is stored as the weights between the neurons [2, 3]. Unlike an expert system needs a human to build its knowledge base, ANN can learn itself from training data. But it's hard to explain the decision process of a trained neural network [2-9]. And at some applications, like credit approval and medical diagnosis, explaining the reasoning is important. If the

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1091 – 1098, 2006.

merits of these two systems can be combined, a powerful and efficient system is then established. Rule extraction is just a technique to combine expert systems with neural networks. It can extract logic rules from neural networks and these rules can be used into the knowledge base of expert systems. Besides, it can learn itself from the process parameters just as a neural network system. Experimental results show that the prediction accuracy of film deposition caused by CVD with this technique is the same as using a technique of neural network. Therefore, an intelligent decision making tool for any production process can be expected in the future with this hybrid system.

2 Rule Extraction Method

In the section, a rule extraction algorithm (bound decomposition tree algorithm, BDT algorithm) [10-12] is presented as following:
Assume a Boolean neural network is a network with *N* binary inputs, i.e.

$$\mathbf{x} = \{x_1, x_2, ..., x_N\} \quad x_i \in \{0,1\} \tag{1}$$

It is constituted of a series of neurons of the form as

$$y = f(u(x)) \tag{2}$$

$$u(x) = \sum_{i=1}^{N} w_i x_i + w_0 = \mathbf{w}^T \mathbf{x} + w_0 \tag{3}$$

where $\mathbf{w} = \{w_1, w_2, ..., w_N\}$ is the weight vector and $\boldsymbol{w_0}$ is bias. The threshold function $f(\cdot)$ possesses the properties is

$$f(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases} \tag{4}$$

With the above representation, to extract rules from the neuron, it means to design an effective algorithm to search all **x**s which satisfy $u(\mathbf{x}) \geq 0$.

Definition 1 (cube)
A ***cube*(w*)** is a set of all input vectors **x** of a neuron.

Definition 2 (bound of cube)
Because the input term $\boldsymbol{x_i}$ of neuron is Boolean logic type (either 0 or 1), $u(\mathbf{x})$ can be just simply showed as the sum of $\boldsymbol{w_i}$ when $\boldsymbol{x_i = 1}$

$$u(\mathbf{x}) = \sum_{i, x_i = 1} w_i + w_0 \tag{5}$$

Only when $\boldsymbol{x_i}$ is 1, the corresponding $\boldsymbol{w_i}$ would give contribution to $u(\mathbf{x})$. The maximal $u(\mathbf{x})$ will be constructed when all $\boldsymbol{x_i}$ with positive $\boldsymbol{w_i}$ are 1 and others are 0. On the other hand, The minimal $u(\mathbf{x})$ will be constructed when all $\boldsymbol{x_i}$ with negative $\boldsymbol{w_i}$ are 1 and others are 0. Each input vector **x** of cube satisfies difference $u(\mathbf{x})$. The ***bound*** of a cube is the maximal and minimal $u(\mathbf{x})$ which be constructed by input vector **x** of the cube. It is showed as

$$\boldsymbol{bound}(\boldsymbol{cube}(\mathbf{w}^*)) = [\boldsymbol{lbound}, \boldsymbol{ubound}] \tag{6}$$

where the *lower bound* (***lbound***) and the *upper bound* (***ubound***) are

$$lbound = \sum_{i, w_i < 0} w_i + w_0 \quad \text{and} \quad ubound = \sum_{i, w_i > 0} w_i + w_0 \tag{7}$$

In order to find out the rules of a neuron, it needs to find the subsets of cube (input vectors) which can make the neuron activated ($u(\mathbf{x}) \geq 0$). Therefore, it needs to divide the cube into smaller subsets of input vectors. At first, it can assign x_1 to be 0 and 1 separately, and then the original cube can be divided into two parts. One sub-cube is the set of input vectors which x_1 is 0; and the other is the set of input vectors which x_1 is 1. With the same method, each sub-cube can be divided into smaller sub-cubes with x_2 be assigned again. Then with more x_i terms are assigned in a sub-cube, the sub-cube could be smaller subset of input vectors.

Definition 3(sub-cube)
A *sub-cube* of a cube is a set of input vectors **x** of a neuron which first m x_i terms ($x_1 \sim x_m$) be assigned and is showed as:

***cube*(w*,** $x_1x_2 \ldots x_m$**)**

where **w*** is weights and bias of the neuron
and $x_1x_2 \ldots x_m$ is the value of first x_i terms which be assigned

Definition 4 (bound of sub-cube)
Each sub-cube is a sub-set of input vectors. Input vectors in each sub-cube provide different $u(\mathbf{x})$. Thus, we can find out the extreme value of a sub-cube. Just like definition 2, the ***bound of a sub-cube*** is the maximal and minimal $u(\mathbf{x})$ which can be constructed by input vectors of the sub-cube and is showed as

$$\textbf{\textit{bound}}(\textbf{\textit{cube}}(\mathbf{w}^*, x_1x_2 \ldots x_m)) = [\textbf{\textit{lbound}}, \textbf{\textit{ubound}}] \tag{8}$$

$$\begin{aligned} \textbf{\textit{lbound}}(\textbf{\textit{cube}}(\mathbf{w}^*, x_1x_2 \ldots x_m)) &= min\{u(\boldsymbol{x}) \mid \boldsymbol{x} \in \textbf{\textit{cube}}(\mathbf{w}^*, x_1x_2 \ldots x_m)\} \\ &= \sum_{i=1, x_i=1}^{m} w_i + \sum_{j=m+1, w_j<0}^{n} w_j + w_0 \end{aligned} \tag{9}$$

$$\begin{aligned} \textbf{\textit{ubound}}(\textbf{\textit{cube}}(\mathbf{w}^*, x_1x_2 \ldots x_m)) &= max\{u(\boldsymbol{x}) \mid \boldsymbol{x} \in \textbf{\textit{cube}}(\mathbf{w}^*, x_1x_2 \ldots x_m)\} \\ &= \sum_{i=1, x_i=1}^{m} w_i + \sum_{j=m+1, w_j>0}^{n} w_j + w_0 \end{aligned} \tag{10}$$

Definition 5 (positive cube)
If the lower bound of a sub-cube is positive, it means all the input vectors of the sub-cube would activate the neuron. Such a sub-cube is called ***positive cube***.

Definition 6 (negative cube)
If the upper bound of a sub-cube is negative, it means all the input vectors of the sub-cube would not activate the neuron anymore. Such a sub-cube is called ***negative cube***.

Definition 7 (uncertain cube)
If the ubound of a sub-cube is positive and the lbound is negative, it means only some input vectors would activate the neuron. Thus it needs to be decomposed again. Such a sub-cube is called ***uncertain cube***.

If a ***cube*(w*,** $x_1x_2 \ldots x_m$**)** is a positive cube, then the corresponding rule is

$$\prod_{i=1}^{m} e(x_i) \quad where \quad e(x_i) = \begin{cases} x_i & , x_i = 1 \\ \overline{x_i} & , x_i = 0 \end{cases} \tag{11}$$

To extract rules from a neuron is finding the all sub-sets of input vectors which could make the neuron activated. And the positive cubes are such kinds of sub-sets that all input vectors in them would make neuron activated. Therefore the principle of the Bound Decomposition Tree algorithm (BDT) is to divide the cube of all input vectors of a neuron into sub-cubes again and again until all positive cubes had been found. The process of the algorithm can be seen as a hierarchical search.

Algorithm 1. Bound Decomposition Tree (BDT)

Input: A neuron's weights and bias
Output: Extract rules of the neuron

Step 1. Set ***positive cube set, negative cube set, positive rule set,*** and ***negative rule set*** as empty
Set ***uncertain cube set*** as ***cube*****(w*)**
Step 2. Select a ***cube x*** form *uncertain cube set*
Step 3. Divide cube x into two sub-cubes: ***cube m***, ***cube n***
Step 4. Compute the bound of *cube m and n* with *Eqs.(9)&(10)*
Step 5. Check cube m and cube n separately,
If *the cube* is positive cube, add it to *positive cube set*
Else if *the cube* is negative cube, add it to *negative cube set*
Else add it to *uncertain cube set*
Step 6. If the uncertain cube set is not empty go to *Step 2*.
Step 7. Transform each cube in the *positive cube set* into a rule with *Eq.(11)* into *positive rule set*.
Step 8. Then, *positive rule set* contains the rules which make the neuron active; and *negative rule set* contains the rules which make the neuron inactive.
(end)

In order to reduce the extracted rule, the BDT Algorithm has a pruning option. It's just need to redefine Definition 5&6 as the follow:

Definition 8(positive cube and negative cube with threshold Δ)

A sub-cube is a *positive cube* if lbound $\geqq -\Delta$.

A sub-cube is a *negative cube* if ubound $\leqq \Delta$.

3 Results and Discussion

3.1 Training Data

The neural network's training data (process parameters and membrane thickness) were collected from the CVDs' Quality control (QC) data of Chungwa Picture Tubes, LTD. (CPT). The training data has 122 sets of records from several CVD machines. The raw records contains 34 kinds of parameters, include pressures of Loadlock In (Out) Chamber (LL\IN and LL\OUT) and Transfer Chamber(TC), pressures and flow rates of 11 gases in Process Chamber(PC), cavity pressure, RF output power, etc. Because the time interval of each record in CPT is not fixed, therefore it is hard to be seen as a

time sequence data. As the result, we choice 4 values of maximum, minimum, average, and standard deviation for each parameter to be feature values. With above 136 parameter's feature values, we also choice working time and pre-mix pressure of gas to be the inputs of the training data. Five values of membrane thickness including U value (Uniformity), average, maximum, minimum and standard deviation are chosen to be the outputs of the training data.

The training data has 138 inputs, and it may be too large for neural network and rule extraction. Thus, the correlation values between these inputs and outputs are determined, and only the inputs which have high correlation (> 0.5) with outputs are finally chosen as the neural network's inputs.

From the above correlation analysis, we find that almost all inputs have lower correlation with U value and standard deviation of membrane thickness. For this reason, U value and standard deviation were also removed from the outputs.
And the chosen inputs are:

- x_1. Average flow rate of GAS02
- x_2. Average flow rate of GAS04
- x_3. Minimum flow rate of GAS04
- x_4. Average flow rate of GAS05
- x_5. Minimum flow rate of GAS05
- x_6. Maximum flow rate of GAS05
- x_7. Maximum flow rate of GAS09
- x_8. Average flow rate of GAS10
- x_9. Average flow rate of GAS12
- x_{10}. Minimum flow rate of GAS12
- x_{11}. Average pressure of PC
- x_{12}. Minimum pressure of PC
- x_{13}. Maximum pressure of PC
- x_{14}. Standard deviation of PC pressure
- x_{15}. Minimum pressure of LL2\out
- x_{16}. Average power of RF

3.2 Training with the Neural Network

From the above analysis, a neural network with 16 inputs and 3 outputs is built to learn from the training data. With 122 sets of collected records, 92 sets are used to be the training data of the neural network and the other 30 sets are used to be test data.

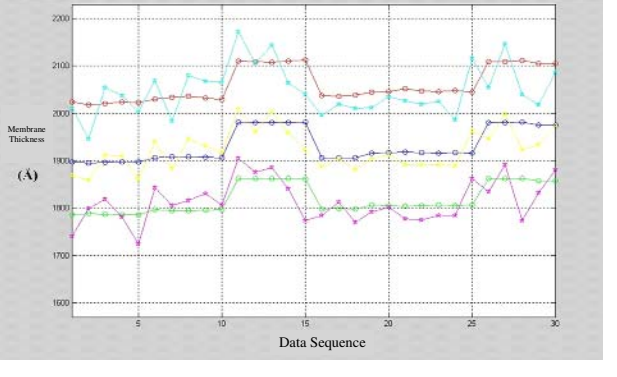


Fig. 1. The prediction results of the trained neural network with the test data

After training with training data, we use test data to test the trained neural network. As shown in Fig.1, the obtained average errors of predicted maximum, average, and minimum of membrane thickness are ±1.92%, ±1.34%, and ±1.61%, respectively. Corresponding to the membrane thickness are ±39.19 Å, ±25.83 Å, and ±29.06 Å. Since the prediction accuracy is roughly 98%, it is possible to develop a control tool of membrane thickness in the deposition process by using these 16 inputs.

3.3 Extracted Rules from the Trained Network

After training and extracting the rules from the trained network, 962, 1258, 1107 rules for these outputs of maximum, average, and minimum of membrane thickness were obtained, respectively. When all of the rules are used to control the deposition process, the precision will be equivalent to the prediction precision obtained from the trained neural network [12]. Since these rules are too complex and not easy to be surveyed with human, the extracted rules can be simplified and reduced by using the pruning method via a choice of suitable threshold value. For example, after a pruning, the numbers of the extracted rules for maximum, average, and minimum of membrane thickness are reduced to 24, 37, and 28, respectively. The extracted rules with pruning of maximum membrane thickness are shown in Table 1.

The rule 1 means if inputs $\boldsymbol{x_{14}}$ and $\boldsymbol{x_{16}}$ are all true and $\boldsymbol{x_{13}}$ is false, then outputs is true. But in this research, the inputs and outputs are both continues values, so that it should be explained as: when the input values of $\boldsymbol{x_{14}}$ and $\boldsymbol{x_{16}}$ get rise and input value of $\boldsymbol{x_{13}}$ is lower, then the output value will be larger, i.e., the maximum thickness of membrane will be thicker. Similarly, the rule 11 shows if input values of $\boldsymbol{x_3}$ and $\boldsymbol{x_{13}}$ get lower and input values of $\boldsymbol{x_9}$ and $\boldsymbol{x_{10}}$ get rise, the output value also will be larger. As a result, the rules on the above table can be integrated and shown as:

$$\begin{aligned} Rule = {} & x_{13}'x_{14}x_{16} + x_3'x_{14}x_{16} + x_3'x_{13}'x_{16} + x_9x_{14}x_{16} + x_9x_{13}'x_{16} + x_3'x_9x_{16} + \\ & x_3'x_9x_{13}'x_{14} + x_9x_{10}x_{16} + x_9x_{10}x_{13}'x_{14} + x_3'x_9x_{10}x_{14} + x_3'x_9x_{10}x_{13}' + \\ & x_{10}x_{14}x_{16} + x_{10}x_{13}'x_{16} + x_3'x_{10}'x_{16} + x_3'x_{10}x_{13}'x_{14} + x_8'x_9x_{10}x_{14} + \\ & x_8'x_9x_{10}x_{13}' + x_8'x_{10}x_{13}'x_{14} + x_3'x_8'x_{10}x_{14} + x_3'x_8'x_{10}\,x_{13}' + x_8'x_9x_{13}'x_{14} + \\ & x_3'x_8'x_9x_{14} + x_3'x_8'x_9x_{13}' + x_3'x_8'\,x_{13}'x_{14} \end{aligned} \tag{12}$$

From Table 1, it can find that input x_{14} and the maximum membrane thickness are in direct proportion. The input x_{14} is the standard deviation of PC pressure during the depositing process. Large value of x_{14} means the pressure is unstable, the uniformity of the membrane thickness will be bad, and the distance between maximum and minimum of membrane will be large. Therefore, it's a reasonable inference that if x_{14} gets rise then the maximum membrane thickness will be thicker; And input x_{16} is the average power of RF. For the experience, lager power of RF will get thicker membrane.

Approximately, these extracted rules fit the experience. The 92 sets of records of training data are used to be the standard, and are compared to 30 sets of testing data. The obtained accuracies of maximum, minimum and average of membrane thickness were 75.9%, 77.7%, and 82.02%, respectively. The accuracy is not bad for the case with only 24~37 rules. In case a higher accuracy is needed, the pruning threshold could be adjusted to a smaller value, and the numbers of the extracted rules are also increased.

Table 1. The extracted rules of maximum membrane thickness from trained neural network with pruning

	x_1	x_2	x_3	x_4	x_5	x_6	X_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
Rule 1	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	0	1
Rule 2	0	0	-1	0	0	0	0	0	0	0	0	0	0	1	0	1
Rule 3	0	0	-1	0	0	0	0	0	0	0	0	0	-1	0	0	1
Rule 4	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1
Rule 5	0	0	0	0	0	0	0	0	1	0	0	0	-1	0	0	1
Rule 6	0	0	-1	0	0	0	0	0	1	0	0	0	0	0	0	1
Rule 7	0	0	-1	0	0	0	0	0	1	0	0	0	-1	1	0	0
Rule 8	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1
Rule 9	0	0	0	0	0	0	0	0	1	1	0	0	-1	1	0	0
Rule 10	0	0	-1	0	0	0	0	0	1	1	0	0	0	1	0	0
Rule 11	0	0	-1	0	0	0	0	0	1	1	0	0	-1	0	0	0
Rule 12	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1
Rule 13	0	0	0	0	0	0	0	0	0	1	0	0	-1	0	0	1
Rule 14	0	0	-1	0	0	0	0	0	0	1	0	0	0	0	0	1
Rule 15	0	0	-1	0	0	0	0	0	0	1	0	0	-1	1	0	0
Rule 16	0	0	0	0	0	0	0	-1	1	1	0	0	0	1	0	0
Rule 17	0	0	0	0	0	0	0	-1	1	1	0	0	-1	0	0	0
Rule 18	0	0	0	0	0	0	0	-1	0	1	0	0	-1	1	0	0
Rule 19	0	0	-1	0	0	0	0	-1	0	1	0	0	0	1	0	0
Rule 20	0	0	-1	0	0	0	0	-1	0	1	0	0	-1	0	0	0
Rule 21	0	0	0	0	0	0	0	-1	1	0	0	0	-1	1	0	0
Rule 22	0	0	-1	0	0	0	0	-1	1	0	0	0	0	1	0	0
Rule 23	0	0	-1	0	0	0	0	-1	1	0	0	0	-1	0	0	0
Rule 24	0	0	-1	0	0	0	0	-1	0	0	0	0	-1	1	0	0

4 Conclusions

A rule extraction algorithm of neural networks for the prediction of the membrane thickness in PECVD process is presented. Because these processes are usually operated with complex and nonlinear reactions, the process parameters are always drifting and variating. The extracted rules can explain the influence between parameters and membrane thickness. After checking, explaining and integrating the extraction rules into knowledge base, the rules can be the basics of membrane thickness prediction and alarm diagnosis in PECVD system. When the parameters are abnormal, to make the membrane thickness out of control limit, an alert message can be offered and also the knowledge base can provide a reference to the engineers for the work of recipe adjustments.

Acknowledgement

The authors gratefully acknowledge the support of the National Science Council of Taiwan, the Republic of China, under project number NSC93-2218-E-033-006.

References

1. Jansen, F.: AVS Short Course:PECVD. American Vacuum Society (1990)
2. Haykin, S.: Neural Netwroks. Prentice Hall International, New Jersey (1999)
3. Freeman, J.A., Skapura, D.M.: Neural Networks. Addison-Wesley, New York (1992)
4. Garcez, A.S. d'Avila, Broda, K., Gabbay, D.M.: Symbolic Knowledge Extraction From Trained Neural Networks: A Sound Approach. Artificial Intelligence 125 (2001) 155-207
5. Buchnan, B. G., Shortliffe, E. H.: Rule-based expert systems. Addison-Wesley, New York (1984)
6. Setiono, R.: A Penalty Function Approach for Pruning Feedforward Neural Networks. Neural Computation 9(1) (1997) 185-204
7. Towell, G.G., Shavlik, J.W.: The Extraction of Refined Rules from Knowledge-Based Neural Networks. Machine Learning 13 (1993) 71-101
8. Fu, L.M.: Rule Generation from Neural Networks. IEEE Trans. System, Man, and Cybernetics 28(8) (1994) 1114-1124
9. Andrews, R., Diederich, J., Tickle, A. B.: Survey and Critique of Techniques for Extracting rules from Trained Artificial Neural Networks. Knowledge Base Systems 8(6) (1995) 373-389
10. Chen, J.C., Liu, T.S., Weng, C.S., Heh, J.S.: An Expert System of Coronary Artery Disease in Chinese and Western Medicine. In: 6th Asian-Pacific Conference on Medical and Biological Engineering, Tsukuba, Japan (2005)
11. Chang, M., Chen, J.C., Chang, J.W., Heh, J. S.: Advanced Process Control Expert System of CVD Membrane Thickness Based on Neural Network. Material Science Forum 505-507 (2006) 313-318
12. Chen, J.C., Heh, J.S., Chang, M., Designing a Decompositional Rule Extraction Algorithm for Neural Networks. In: Third international Symposium on Neural Networks, Chengdu, China (2006) (accepted)

PCA-Based Neural Network Modeling Using the Photoluminescence Data for Growth Rate of ZnO Thin Films Fabricated by Pulsed Laser Deposition

Jung Hwan Lee[1], Young-Don Ko[1], Min-Chang Jeong[2], Jae-Min Myoung[2], and Ilgu Yun[1]

[1] Semiconductor Engineering Laboratory, Department of Electrical and Electronics Engineering, Yonsei University, 134 Shinchon-Dong, Seodaemun-Gu, Seoul 120 - 749, Korea
{backta, ksimpson, iyun}@yonsei.ac.kr
[2] Information and Electronic Materials Research Laboratory, Department of Materials Science and Engineering, Yonsei University, 134 Shinchon-Dong, Seodaemun-Gu, Seoul 120 - 749, Korea

Abstract. The process modeling for the growth rate of pulsed laser deposition (PLD)-grown ZnO thin films was investigated using neural networks (NNets) based on the back-propagation (BP) algorithm and PCA-based NNets using photoluminescence (PL) data. D-optimal experimental design was performed and the growth rate was characterized by NNets. PCA-based NNets were then carried out in order to build the model by PL data. The statistical analysis for those results was then used to verify the fitness of the nonlinear process model. Based on the results, this modeling methodology can explain the characteristics of the thin film growth mechanism varying with process conditions and the model can be analyzed and predicted by the multivariate data.

1 Introduction

Recently, with the development of optoelectronic devices, ZnO becomes distinguished as one of the II-VI compound semiconductors that has the indirect band gap, which can be used for the optical device applications such as light emitting diodes, laser diodes as well as it is possible to manufacture invisible field effect transistors towards a flat panel display due to exciting optical properties [1-3]. Considering the electrical and optical characteristics of this material, it can be widely used in the useful applications that are antireflection coatings, transparent electrodes in solar cells, gas sensors, and surface acoustic wave devices [4]. However, it is difficult to characterize the manufacturing process for its nonlinear characteristic in general due to the nature and the unavoidable random variations of the process. Therefore, the response factor must be characterized with respect to the varying process conditions. Additionally, the

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1099 – 1104, 2006.

accurate explanation by the process model can significantly impact the values and the range of process input factors. The statistical variations of process input factors must be carefully examined to build the process model by considering the ranges and the values of process variables.

There have been researched that the process modeling to control the conditions and predict the response by several scientists as well as the various modeling methodology is investigated to improve the yield of the process and the accuracy of the model. G. Triplett *et al*. and B. Kim *et al*. applied NNets to the semiconductor manufacturing processes [5][6]. S. J. Hong *et al*. investigated NNets of the reactive ion etching using optical emission spectroscopy data which are transformed by PCA to compress the multivariate data set [7].

This paper presents a modeling methodology technique applied to the thin film fabrication process by using NNets. Principal component analysis (PCA) was used to reduce the multivariate data set related to the measurement data and PCA-based NNets can provide better explanation compared to simple NNets by using comprehensive PL data. D-optimal design was used to make design matrix in this experiment and then carry out the modeling. The statistical analysis with the assumption of the model is also verified under the significance level.

2 Experiments

The initial substrate of n-type InP has $3x10^{18}$ cm^{-3} doping concentrations. The PLD technique was used for the deposition of n-type ZnO layer. The chamber was evacuated by a turbomolecular pump to base pressure at $1x10^{-6}$ Torr. Pulsed Nd:YAG laser was operated at a 355-nm wavelength with 2 Hz repetition rate and 2.5 J/cm^2 energy density. The ZnO films were deposited with varying, substrate temperature in the range of 350-450 °C and oxygen pressure in the range of 250-450 mTorr. A substrate holder was placed at 5 cm from the target. After ZnO thin films were deposited by PLD process, the thicknesses of the ZnO thin films on InP substrate were measured by using SEM (scanning electron microscopy) and the intensities of PL were measured, respectively.

3 Modeling Scheme

3.1 Design of Experiments

Substrate temperature (T) and oxygen pressure (P) were selected as input factors to investigate the characterization of growth rate of ZnO thin films. These input factors were explored via D-optimal experimental design with 17 runs to minimize the volume of the joint confidence region on the vector of regression coefficients [8]. Table 1 shows the experimental design matrix of input factors used in each run. The run order has been randomized to avoid statistically the effect of irrelevant factors.

Table 1. D-optimal design matrix

Run	T [℃]	P [mTorr]	Run	T [℃]	P [mTorr]
1	400	450	10	375	400
2	400	300	11	400	350
3	375	350	12	450	450
4	350	250	13	400	250
5	425	400	14	350	450
6	425	300	15	350	350
7	400	400	16	375	300
8	450	350	17	450	250
9	425	350			

3.2 Neural Networks

Neural networks are utilized to model the nonlinear relationship between inputs and outputs in semiconductor process modeling. The networks consist of the three layers that are the input layer, the hidden layer and the output layer. That is comprised of simple processing units called neurons, interconnection, and weights that are assigned to the interconnection between neurons [9]. Each neuron contains the weighted sum of its inputs filtered by a nonlinear sigmoid transfer function. Neural network structure and parameters used in this study are shown in Table 2.

In this study, the simple random sampling and the Latin Hypercube Sampling (LHS) were used to select randomized value for the initial weights and biases. The BP algorithm was used in NNets.

Table 2. Neural networks structures and parameters

	NNets	PCA-based NNets
Structure	2-9-9-1	2-5-3-1
Learning rate	0.0025	0.0275
Momentum	0.95	0.06

3.3 Principal Component Analysis

PCA is used to reduce the dimensionality of a data set which consists of a large number of interrelated variables. Consider a vector x, which consist of p random variables. Let $\sum$ be the covariance matrix of x. then, for k=1, 2, …, r, the k th PC is given by [9]

$$Z_k = a_k' x \tag{1}$$

where a is an eigenvector of $\sum$ corresponding to its k th largest eigenvalue λ_k.

In this study, 166 data points of PL were reduced to 3 PCs using PCA. Those provide a reasonable summary of the PL data, accounting for about 90.16% of the total variance. Figure 1 shows the cumulative proportion of number of PCs.

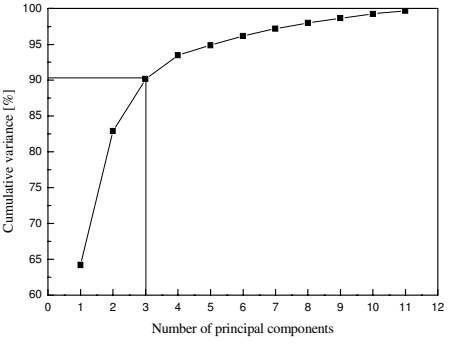


Fig. 1. Cumulative proportion of number of PC

4 Result

The neural network modeling results versus measured values are illustrated in Fig. 2 (a). It is verified that the linear relationship between the network output values and the experimental data. The '▪' and the '△' points designate the training and testing data, respectively. The root mean square errors (RMSEs) of this model for training and testing data are 0.051 and 0.085, respectively.

The 3-D surface plot of response model for the growth rate is shown in Fig. 2 (b). It is observed that the growth rate for PLD-grown ZnO thin film is near the maximum growth rate when the temperature is in the range of 330 ~ 350°C and the pressure is in the range of 390 ~ 410 mTorr. Zn and O_2 molecules do not have enough thermal energy to form stoichiometry ZnO thin film so that Zn and O_2 molecules need enough thermal energy in order to react up to the maximum growth rate. As the temperature and the pressure increase, the reaction between Zn and O_2 molecules actively occur, the growth rate increases with the improvement of the oxygen-deficient stoichiometry ZnO thin film simultaneously. Above the maximum growth rate, the increased temperature affects the number of Zn due to Zn evaporation leading to inferior stoichiometric ZnO as Zn has melting point at 693 K [10]. With the change of Zn increases, oxygen vacancies decrease under the increase of the oxygen pressure at the same time [11-12]. From above reasons, the oxygen composition and the change of temperature that affect the deposition of the thin film in the circumstance. The growth rate was decreased on the circumstance fully filled with oxygen to limit the crystallite size of the thin film on the higher substrate temperature.

The PCA-based neural network model results and the residual plots for the growth rate are illustrated in Fig. 3. It indicated that the measured values and the predicted value are well fitted. The residuals are randomly distributed without either biased or special patterns. The RMSEs of this model for training and testing data are 0.03 and 0.01, respectively.

The principal components, which were from compressed PL data set by PCA, can be applied to predict the growth rate of thin film.

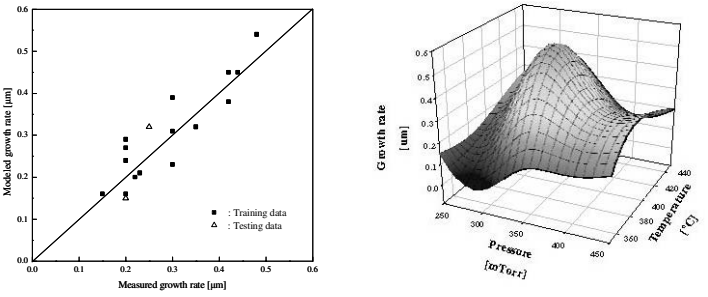


Fig. 2. Neural network modeling result for the growth rate : (a) The measured vs. the predicted values and (b) The surface plot of neural network model for the growth rate

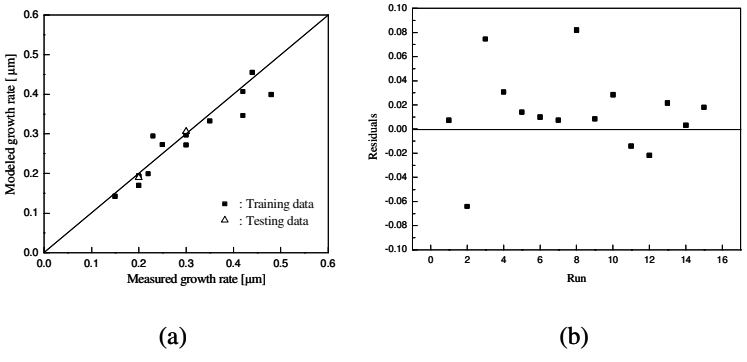


Fig. 3. The PCA-based neural network modeling results for the growth rate: (a) The measured vs. the predicted values and (b) The residual plot

5 Conclusion

The growth rate of thin films was investigated by NNets and the PCA-based NNets. The initial weights and biases were selected by the simple random sampling and the LHS. From these results, the neural network model can explain the comprehensive effects of the response on the varying process conditions in accordance with the physical mechanisms as well as the response model can be developed via the dimension reduced the PL data by PCA. The methodology can allow us to predict the growth rate of thin films with respect to process conditions as well as it can improve the manufacturability.

Acknowledgment

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support

program supervised by the IITA (Institute of Information Technology Assessment). The authors would like to thank Prof. S. Y. Lee for fabricating the test structures.

References

1. Aoki, T., Hatanaka, Y., Look, D. C.: ZnO Diode Fabricated by Excimer-Laser Doping. Appl. Phys. Lett. 76 (2000) 3257-3258
2. Ryu, Y.R., Kim, W.J., White, H.W.: Fabrication of Homostructural ZnO P-N Junction. J. Cryst. Growth 219 (2000) 419-422
3. Nakayama, T., Murayama, M.: Electronic Structures of Hexagonal ZnO/GaN Interfaces. J. Cryst. Growth 214 (2000) 299-303
4. Studenikin, S.A., Golego, N., Cocivera, M.: Fabrication of Green and Orange Photoluminescent, Undoped ZnO Films Using Spray Pyrolysis. J. Appl. Phys. 84 (4) (1998) 2287-2294
5. Triplett, G., May, G.S., Brown, A.: Modeling Electron Mobility in MBE-Grown InAs/AlSb Thin Films for HEMT Applications Using Neural Networks. Solid-State Electron (2002) 1519-1524
6. Kim, B., May, G.S.: An Optimal Neural Network Process Model for Plasma Etching. IEEE Trans. Semiconduct Manufact (1994) 12-21
7. Hong, S.J., May, G.S., Park, D.C.: Neural Network Modeling of Reactive Ion Etching Using Optical Emission Spectroscopy Data. IEEE Trans. Semiconduct Manufact. (2003) 598-608
8. de Aguiar, P.F., Bourguignon, B., Khots, M.S., Massart, D.L., Phan-Than-Luu, R.: D-optimal Designs. Chemomet. Intell. Lab. Syst. 30 (1995) 199–210
9. Triplett, G., May, G., Brown, A.: Modeling Electron Mobility in MBE-Grown InAs/AlSb Thin Films for HEMT Applications Using Neural Networks. Solid-State Electronics 46 (10) (2002) 1519-1524
10. Bae, S.H., Lee, S.Y., Jin, B.J., Im, S.: Growth and Characterization of ZnO Films Grown by Pulse Laser Deposition. Appl. Surf. Sci. 169 (2001) 525-528
11. Jin, B.J., Bae, S.H., Lee, S.Y., Im, S.: Effects of Native Defects on Optical and Electrical Properties of ZnO Prepared by Pulsed Laser Deposition. Mat. Sci. Eng. B 71 (2000) 301-305
12. Jin, B.J., Woo, H.S., Im, S., Base, S.H., Lee, S.Y.: Relationship Between Photoluminescence and Electrical Properties of ZnO Thin Films Grown by Pulsed Laser Deposition. Appl. Surf. Sci. 169 (2001) 521-524

Wood Defects Classification Using a SOM/FFP Approach with Minimum Dimension Feature Vector

Mario I. Chacon and Graciela Ramirez Alonso

Chihuahua Institute of Technology, DSP & Vision Lab.,
Ave. Tecnologico 2909,
31310 Chihuahua, Chih., Mexico
mchacon@itchihuahua.edu.mx

Abstract. This paper describes the design and implementation of a wood defect classifier. The defects are four different types of knots found in wood surfaces. Classification is based on features obtained from Gabor filters and supervised and non supervised artificial neural networks are used as classifiers. A Self-organizing neural network and a fuzzy Self-organizing neural network were designed as classifiers. The fuzzy SONN shows a reduction on the training time and had a better performance. A final classifier, a feedforward perceptron using the weights of the fuzzy SONN as initial weights turn to be the best classifier with a performance of 97.22% in training and 91.17% in testing. The perceptron classifier surpasses a human inspector task which has a maximum performance of 85%.

1 Introduction

The value of a piece of wood is directly related to the quality of the wood. The quality is determined considering the number of defects and their distribution. Knots are the most common defects found during wood inspection [1]-[5]. In this work four different types of knots are considered, *encased*, *leaf*, *edge*, and *sound*. Some examples of these types of knots are shown in Fig.1. This paper presents an experiment using supervised and non supervised artificial neural networks, ANNs, to perform the classification of four different types of knot defects. The features used in the experiments are texture features obtained from the defects with Gabor filters. The images used in this work were obtained from [6].

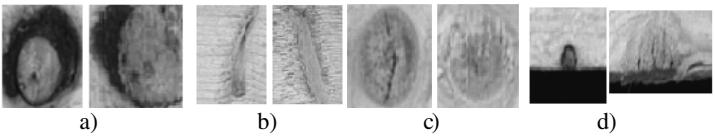


Fig. 1. Examples of knot defects, a) encased, b) leaf, c) sound, d) edge

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1105 – 1110, 2006.

2 Feature Extraction

2.1 Gabor Filters

Gabor filters are used for image analysis due to their biological relevance and computational properties [7]. These filters are frequency and orientation sensitive. The filters are 2D selective band pass filters with respect frequency and orientation. The filters constitute a family or a filter bank where each filter is dilated, translated and rotated with regard to each other. The mathematical definition of the Gabor filter is,

$$\psi(f,\theta,x,y) = exp\left\{ i(f_x x + f_y y) - \frac{f^2(x^2+y^2)}{2\sigma^2} \right\}. \quad (1)$$

where $f_x = f\cos\theta$, $f_y = f\sin\theta$ *and* $i = \sqrt{-1}$, x and y are pixel coordinates of the image in the interval $(-x/2, x/2)$ and $(-y/2, y/2)$. The central frequency of the band pass is f, θ is the orientation of the filter, and σ determines the bandwidth. The Gabor transform of an image, corresponds to the convolution of the image with the family of Gabor kernels.

$$\Im\{O_{f,\theta}(x,y)\} = \Im\{I(x,y)\}\Im\{\psi(f,\theta,x,y)\}. \quad (2)$$

thus the filtered images are

$$O_{f,\theta}(x,y) = \Im^{-1}\{\Im\{I(x,y)\}\Im\{\psi(f,\theta,x,y)\}\}. \quad (3)$$

where $I(x,y)$ is the image of the defect, $\psi(f,\theta,x,y)$ is the family of the Gabor filters and $O_{f,\theta}(x,y)$ corresponds to the filtered images. In this work only information of the magnitude of the filter is used. Defect features were obtained with the Gabor filters implemented in the frequency domain. All defect images were normalized to 64x64. The output images are filtered images, $O_{f,\theta}(x,y)$ for $f \in \{1,..4\}, \theta \in \{1,..8\}$. Each output image $O_{f,\theta}(x,y)$ is concatenated by rows to form a vector, $X^{f\theta}$, to represent that output image. Then, $X^{f\theta}$ is subsampled by a factor of 4 to reduce the size of the vector to 1X1024. Finally all the subsampled vectors $\left(X_{\downarrow 4}{}^{f\theta}\right)^T$, are used to form a feature matrix.

2.2 Dimension Reduction of the Feature Vector

Once the feature matrices that represent the knot defects are defined, they are processed by two methods to select the most discriminative features. The tested methods are incorporation and principal components. The incorporation method yielded the best results in this work. The best features found with the incorporation

method are: mean and standard deviation of the gray levels in the defect, and seven Gabor features that represent the standard deviation of the images generated by the Gabor filters The final feature vector is defined by

$$X=[\,\mu_{I(x,y)},\sigma_{I(x,y)},\sigma_{\psi_{2,8}},\sigma_{\psi_{3,5}},\sigma_{\psi_{4,1}},\sigma_{\psi_{4,2}},\sigma_{\psi_{4,5}},\sigma_{\psi_{4,7}},\sigma_{\psi_{4,8}}\,]. \tag{4}$$

where $\mu_{I(x,y)}$ is the mean value of the original defect image, $\sigma_{I(x,y)}$ is the standard deviation of the original defect image, and σ_{ψ} is the standard deviations of the images generated by the Gabor filters.

3 Neural Networks Classifier

The design of the classifier is based on two types of neural networks, Self-organizing neural network, SONN and a feedforward perceptron neural network, FFPNN. These two types of networks were used because they are two of the best networks for pattern recognition tasks [11]. In this work several SONNs and FFNNs were designed and are described next.

3.1 SONN Experiments

The first experiment in the design of the classifier was a SONN trained with the standard Kohonen algorithm [8]. A total of 140 defect samples were used in the design of the SONN, 72 samples were used for training and 68 for testing. Sample class distributions are shown on Table I. Several SONNs with different topologies and number of epochs were tested to obtain the best performance based on the number of samples correctly classified. The best SONN had a topology of 5x2 neurons with a performance of 91.66% in training and 85.28% in testing of correct classification in 550 epochs. Trying to improve this performance a new SONN was designed but now incorporating a fuzzy parameter in the training algorithm [9]. The fuzzy parameter is incorporated to update the weights of the network. The purpose of this variation is to incorporate another criterion, besides the crisp criterion obtained by the Euclidean distance, to designate the class of each input vector. This fuzzy parameter is obtained with the fuzzy C-means algorithm. The fuzzy parameter is included in the learning rule of the SONN as shown in equation (5).

$$\mathbf{w}_i = \mathbf{w}_{i(\,old\,)} + L^{*}\, d\,(\,\mathbf{x}_i, \mathbf{w}_i\,)^{*}\mu_{ix}\,. \tag{5}$$

where $\mathbf{w}_i$ is the updated weight vector, $\mathbf{w}_{i(\,old\,)}$ corresponds to the previous weight, L is the learning rate, $d\,(\mathbf{x}_i,\mathbf{w}_i\,)$ is the Euclidean distance between the input vector $\mathbf{x}_i$ and the weight vector $\mathbf{w}_i$ of a neuron, and μ_{ix} is the fuzzy parameter. In this equation the fuzzy parameter is a measurement of the membership of the input vector, $\mathbf{x}_i$, to the class represented by the neuron i defined by (6).

Table 1. Sample class distribution

Defect	No. of samples		Total
	Training	Test	
Sound	20	20	40
Encased	10	8	18
Edge	22	21	43
Leaf	20	19	39
Total	72	68	140

$$\mu_{ix} = \frac{1}{\left[\sum_{j=1}^{c}\left(\frac{d(\mathbf{x_i}, \mathbf{w}_i)}{d(\mathbf{x_i}, \mathbf{w}_j)}\right)^{\frac{2}{(m-1)}}\right]}. \tag{6}$$

where c is the number of neurons in the network, $d(\mathbf{x}_i, \mathbf{w}_i)$ is the Euclidean distance of the input $\mathbf{x}_i$ with respect the neuron weight, $\mathbf{w}_i$ under analysis, $d(\mathbf{x}_i, \mathbf{w}_j)$ is the Euclidean distance of the input $\mathbf{x}_i$ with respect to each one of the weight vectors of the other neurons, and m is the degree of fuzziness of the fuzzy C-means algorithm. The performance of correct classification of the fuzzy SONN was 91.66% in training and 88.23% in testing in 400 epochs. We can notice the reduction of the training of the SONN using the fuzzy parameter, from 550 to 400 epochs, and the best performance in testing. However, results of the classifier shown that the class *encased* is not well represented by the neural network. Because several SONNs were designed with no improvement in the performance, it was decided to use a perceptron neural network to verify if a better performance can be achieved.

3.2 FFPNN Experiments

Several FFPNN, one layer with ten neurons were trained with the Levenberg-Marquardt algorithm. The same sets of defect samples used in the SONN experiments were used in the FFPNN experiments. The FFPNN was trained using random weights. The parameter μ, of the LM algorithm, was initialized at 0.001, and its maximum value allowed was 10. Under these considerations the training stopped between 70 and 80 epochs. The best performance of the FFPNN achieved was 97.22% of correct classification for training and 85.29% for testing in 80 epochs. A significant reduction in the number of epochs 400/80 is observed in the FFPNN experiment compared with the fuzzy SONN experiment. However, the performance on testing is lower in the FFPNN than in the fuzzy SONN. Now, considering that the fuzzy SONN had a topology of 5x2, it was decided to use a FFPNN with 10 neurons and use the weights of the final state of the fuzzy SONN as the initial weights for the FFPN, to verify two points: if the training time can be reduced, and to improve the

performance without loss of generalization. The best network with these initial weights was obtained using 10 epochs. The performance of the FFPNN was 97.22% for training and 91.17% for testing. The training performance can be improved if the number of epochs increases, however loss of generalization was perceived in the testing performance. Defect classes are coded in the outputs of this FFPNN. Fig. 2 illustrates the final process used in the design of the best ANN classifier. The Gabor filters generate the original feature vector. Then the original feature dimension is reduced with the incorporation method. The fuzzy SONN is trained, and the weights of the best fuzzy SONN are used as the initial weights for the FFPNN. Finally the FFPNN is trained. The confusion matrices for training and testing of the best FFPNN are shown in Tables II.

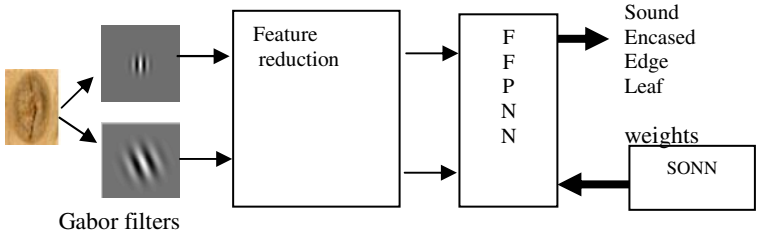


Fig. 2. General scheme of the wood defect classifier

4 Results and Conclusions

This paper described the design of a neural network classifier for four wood defects. The feature vector used for the classifier design was obtained from Gabor filters. The dimensionality of the feature vector was reduced from 66 to 9 features yielding a better control of the curse of dimensionally [10]. Several SONN were designed. Findings in the SONN designs showed that the inclusion of a fuzzy parameter reduced the training time. The best SONN turned out to be a fuzzy SONN with a performance

Table 2. FFPNN confusion matrix for training

Training						
Defect	Sound	Encased	Edge	Leaf	No. Samples	Errors
Sound	20	0	0	0	20	0
Encased	2	8	0	0	10	2
Edge	0	0	22	0	22	0
Leaf	0	0	0	20	20	0
Total errors						2
Testing						
Sound	20	0	0	0	20	0
Encased	2	6	0	0	8	2
Edge	0	0	21	0	21	0
Leaf	4	0	0	15	19	4
Total errors						6

Table 3. Performance of the best ANNS

ANN	Training	Test	No. Epochs
SONN	91.66%	85.28%	550
Fuzzy SONN	91.66&	88.23%	400
FFPNN	97.22%	91.17%	10

of 91.66% in training and 88.23% in testing in 400 epochs. A posterior experiment with a FFPNN showed that using the weights of the fuzzy SONN as initial weights of the FFPNN reduces also the training time and yields a better performance, 97.22% for training and 91.17% for testing in 10 epochs. Table III summarizes the performance of the best ANNs designed. These results are considered acceptable because the maximum performance achieved by a human expert inspector is 85%.

Acknowledgement

The authors greatly appreciate the support from COSNET under grant 445.05P to develop this research.

References

1. Lampinen, J., Smolander, S.: Self Organizing Feature Extraction in Recognition of Wood Surface Defects and Color Images. International Journal of Pattern Recognition and Artificial intelligence (10) (2) (1996.) 97-113
2. Silven, O., Niskanen, M., Kauppinen, H.: Wood Inspection with Non-supervised Clustering. Machine Vision and application (13) (2003) 275–285
3. Lampinen, J., Smolander, S., Korhonen, M.: Wood Surface Inspection System Based on Generic Visual Features. International Conference on Artificial Neural Networks ICANN'95 Paris (1995)
4. Lampinen, J., Smolander, S.: Wood Defect Recognition: A Comparative Study. Proc. of workshop on Machine Vision in Advanced Production Oulu, Finland (1994) 2-6
5. Kauppinen, H., Silven, O., Piirainen, T.: Self Organizing Map Based User Interface for Visual Surface Inspection. Proc. 11th Scandinavian Conference on Image Analysis, Kangerlussuaq, Greenland (1999) 801-808
6. G-SOM tool, University of Oulu Department of Electrical Engineering. URL: http:// www.oulu. fi /research/ imag/gsom/
7. Liu, C., Wechsler H.: Independent Component Analysis of Gabor Features for Face Recognition. IEEE Trans. Neural Networks, (14) (4) (2003) 919-928
8. Kohonen, T.: Self Organizing Maps. 2nd edn. Springer Series in Information Sciences (1995)
9. Kim, D. S., Huntsberger, T.L.: Self-organizing Neural Networks for Unsupervised Pattern Recognition. In: Proceedings of Tenth Annual International Phoenix Conference on Computers and Communications, March 27-30 (1991) 39 –45
10. Egmont, M., Ridder, D.,Handles, H.: Image Processing with Neural Networks- a review. Pattern Recognition (35) (10) (2002) 2279 – 2301

A Kernel Based Multi-resolution Time Series Analysis for Screening Deficiencies in Paper Production

Marcus Ejnarsson[1], Carl Magnus Nilsson[1], and Antanas Verikas[1,2]

[1] Intelligent Systems Laboratory, Halmstad University, Box 823, SE-301 18 Halmstad, Sweden
Marcus.Ejnarsson@ide.hh.se, Carl-Magnus.Nilsson@ide.hh.se
[2] Department of Applied Electronics, Kaunas University of Technology, Studentu 50, LT-513 68, Kaunas, Lithuania
Antanas.Verikas@ide.hh.se

Abstract. This paper is concerned with a multi-resolution tool for analysis of a time series aiming to detect abnormalities in various frequency regions. The task is treated as a kernel based novelty detection applied to a multi-level time series representation obtained from the discrete wavelet transform. Having *a priori* knowledge that the abnormalities manifest themselves in several frequency regions, a committee of detectors utilizing data dependent aggregation weights is build by combining outputs of detectors operating in those regions.

1 Introduction

Increasing demands from customers for uniformity of printing on newsprint mean that the uniformity of the paper structure must be improved. To be able to produce even more uniform paper, much more detailed measuring and analysis of the paper properties are required, if compared to those already available.

The high production speed of a modern paper machine means that most measurements are made on quite a coarse grid. The sensors are typically mounted on a head traversing the web. The typical transverse time for the head is about 30 seconds. Approximately 700 meters of paper will have passed through the paper machine during that time. Variability of the paper structure has been studied by a number of researchers [1], [2], [3], [4] and it was found that such a course sampling strategy works well for assuring the stability of the paper machine itself. However, it is not always adequate for assuring low variations in the paper structure.

The development of sensors and fast, high-capacity computers open up new opportunities in the future supervision strategy for a paper machine to include high-resolution paper- and print-related properties of the final product. The printability of the sheet of paper could be determined by monitoring the very local paper properties. A realistic development for a paper machine in the near future would be to extend the supervision of the paper production into the area

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1111–1116, 2006.

of on-line print quality control. This would require a tremendous increase in measurement capacity as well as a deepened understanding of "what to measure". However, increased measurement capacity means that a large amount of data must be continuously acquired and processed. Obviously, the interpretation of the data measured becomes more complex and time-critical.

In this study, to obtain an on-line fine characterization of the paper structure at a paper mill, a paper web running at of about 30 m/s speed is illuminated by a red diode laser, the reflected light is collected by a photo-detector and recorded as a very large time series of high resolution measurements. The objective of this work is then to develop a tool for on-line analysis of the time series aiming to detect abnormalities in various frequency regions ranging from millimeters to several meters. The abnormalities detected in different frequency regions give an indication for the paper maker about specific disturbances in the paper production process.

2 The Approach

To achieve the goal, the time series is first divided into consequent blocks of a predetermined length and a multi-resolution representation of each block is then obtained via the discrete wavelet transform (DWT). The task of abnormality detection in various frequency regions is then treated as a novelty detection problem using the obtained decompositions. A kernel based novelty detector is adopted in this work. Having *a priori* knowledge that the abnormalities manifest themselves in several frequency regions, a committee of detectors is build by combining outputs of detectors operating in those regions.

2.1 Multi-resolution Time Series Representation

Wavelets are a family of orthonormal basis functions generated by a translation and dilation of *the mother wavelet*, $\psi(t)$ [5]:

$$\psi_{(\nu,\beta)}(t) = \nu^{-1/2}\psi(\frac{t-\beta}{\nu}) \tag{1}$$

where ν and β are the dilation and translation parameter, respectively. The DWT operate in a dyadic discretized manner (power of two) with $\nu = 2^a$ and $\beta = \nu b$ yielding

$$\psi_{(a,b)}(t) = 2^{-a/2}\psi(2^{-a}t - b) \tag{2}$$

where a and b are integers. The dilation parameter a determines the location of the wavelet in the frequency domain while the translation parameter b determines the location of the wavelet in the time domain. Using the DWT, a time series $s(t)$ is represented as:

$$s(t) = \sum_a \sum_b \omega_{a,b}\psi_{(a,b)}(t) \tag{3}$$

where $w_{a,b}$ is a set of wavelet coefficients. The DWT decomposes a time series into several time-frequency regions—scales a, where the amplitude of the wavelet coefficients represents the frequency energy at time t.

We represent the frequency substance of the level a by a distribution (a k-bin histogram) of the wavelet coefficients. The histogram, treated as a k-vector $\mathbf{x}^a \in \Re^k$, is then used as an input data vector for the kernel based novelty detector, at level a.

2.2 The Kernel Based Novelty Detector

Let us assume that the time series is subdivided into M blocks, meaning that at the resolution level a the data set $X^a = \{\mathbf{x}_1^a, \mathbf{x}_2^a, \ldots \mathbf{x}_M^a\}$ is available. $N < M$ of the data points from the set X^a are included into the training set X_t^a used to estimate parameters of the kernel based novelty detector operating on the ath level data. The detector is trained according to the following steps by finding a *hypersphere* of the minimum radius r^* that contains most of the data points from X_t^a [6].

1. Given the set X_t^a, $\gamma > 0$, and $C > 0$, find the weights $\boldsymbol{\alpha}^*$ by maximizing the *dual* Lagrangian:
$$W(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i \kappa(\mathbf{x}_i^a, \mathbf{x}_i^a) - \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j \kappa(\mathbf{x}_i^a, \mathbf{x}_j^a) \tag{4}$$
with $\kappa(\mathbf{x}, \mathbf{x})$ being a kernel and subject to $\sum_{i=1}^{N} \alpha_i = 1$ and $0 \leq \alpha_i \leq C$, $i = 1, ..., N$.
2. Choose i such that $0 \leq \alpha_i^* \leq C$ and find the radius of the hypersphere r^*: $r^* = \sqrt{\kappa(\mathbf{x}_i^a, \mathbf{x}_i^a) - 2\sum_{j=1}^{N} \alpha_j{}^* \kappa(\mathbf{x}_j^a, \mathbf{x}_i^a) + \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i{}^* \alpha_j{}^* \kappa(\mathbf{x}_i^a, \mathbf{x}_j^a)}$
3. Calculate $T = \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i^* \alpha_j^* \kappa(\mathbf{x}_i^a, \mathbf{x}_j^a) - r^{*2} - \gamma$
4. Calculate the function $f(\mathbf{x}^a)$ to test the inclusion of the data point $\mathbf{x}^a$ in the hypersphere:
$$f(\mathbf{x}^a) = \mathcal{H}\big[\kappa(\mathbf{x}^a, \mathbf{x}^a) - 2\sum_{i=1}^{N} \alpha_i^* \kappa(\mathbf{x}_i^a, \mathbf{x}^a) + T\big] \tag{5}$$
where the Heaviside function $\mathcal{H}[y(\mathbf{x}^a)] = 1$ if $y(\mathbf{x}^a) \geq 0$ and -1 otherwise.

2.3 Combining Detectors into a Committee

Abnormalities of some type manifest themselves at several resolution levels $a \in D$ of a time series. Therefore, outputs of novelty detectors operating at those levels are combined into a committee. To address the short time process variations, short term moving averages τ_t^a, $a \in D$ of the detector outputs $y(\mathbf{x}_t^a)$ are also used as information sources in the combination process.

In an operational mode, given the data point $\mathbf{x}_t^a$, the detector operating at the ath level outputs the real valued signal $y(\mathbf{x}_t^a)$ and the binary decision $f(\mathbf{x}_t^a)$. The short term moving average τ_t^a is then given by
$$\tau_t^a = \frac{1}{P}\big[\tau_{t-1}^a (P-1) + y(\mathbf{x}_t^a)\big] \tag{6}$$
with P being the size of the averaging window.

To aggregate outputs of the information sources, the weighted averaging scheme exploiting the data dependent aggregation weights has been applied in this study. Given the data point ($\mathbf{x}_t$), the aggregation weights for the ath detector and the ath short term information source are defined by $w^a_{y,t} = |y(\mathbf{x}^a_t)|$ and $w^a_{\tau,t} = |\tau^a_t|$, respectively. The real valued committee output $y^c(\mathbf{x}_t)$ is then given by

$$y^c(\mathbf{x}_t) = \frac{\sum_{a \in D} \left[f(\mathbf{x}^a_t) w^a_{y,t} + \mathcal{H}\left[\tau^a_t\right] w^a_{\tau,t} \right]}{\sum_{a \in D} [w^a_{y,t} + w^a_{\tau,t}]} \ . \tag{7}$$

The binary committee output is obtained by applying the Heaviside function.

3 Experimental Investigations

As it has already been mentioned, the motivation for this work comes from the paper making industry aiming to create a tool for on-line multi-resolution screening of abnormalities in the paper structure. A set of developed detectors are to continuously analyze a time series characterizing the paper structure and provide the process operator with an indication when the rate of abnormalities detected is higher than expected.

3.1 The Data Set

The data set used in the experimental investigations has been acquired at a Swedish paper mill and corresponds to 120 km of newsprint divided into $M = 604$ blocks. By applying the DWT, each block t, $t = 1, 2, \ldots, M$, is represented by L vectors $\mathbf{x}^a_t$, $a = 1, 2, ..., L$, where L is the number scales used in the DWT. Based on the process knowledge, $N = 366$ of the consequent blocks (60% of the available data) were selected to represent the normal process conditions. These blocks constitute the training data set X_t. The remaining 238 blocks ($t = N+1, ..., M$) were used as the test data set. The abnormalities screened in this study, were located in the wavelength region of $0.4 - 1.6$ m. The novelty detectors of $a = 3$ and $a = 4$ scale operate in this region. Based on the *a priori* knowledge, it was known that for these two detectors the data points $t = 387, ..., 542$ have to be labelled as novel, while the others ($t = N + 1, ..., 386; t = 543, ..., M$) as non-novel—inliers. Having this knowledge, the experimental results can be assessed by calculating the test data set detection accuracy for: *True Positives* (TP) - novel classified as novel, *False Positives* (FP) - novel classified as inlier, *True Negatives* (TN) - inlier classified inlier and *False Negatives* (FN) - inlier classified as novel.

3.2 Parameters and Results

In this work, the Gaussian kernel $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\{-||\mathbf{x}_i - \mathbf{x}_j||^2/\sigma^2\}$, governed by the Gaussian width parameter σ, has been utilized. In addition to σ, there are three more parameters to set, namely, the number of histogram bins k, the parameter γ allowing some slack in the radius of the minimal hypersphere, and

the regularization constant C controlling the trade-off between minimizing the the radius and controlling the slack variables.

Values of the parameters were found by cross validation. The number of histogram bins required is scale dependent. The value of $k = 67$ provided the high enough resolution for both scale 3 and 4. The values of $\gamma = 0.01$, $C = 0.015$ and $\gamma = 0.05$, $C = 0.01$ were found to give good trade-off between the *a priori* process knowledge and the rates obtained for TP/FP and TN/FN. The appropriate kernel width value was found to be $\sigma = 4$.

Figure 1 presents the output values for the single $y(\mathbf{x}_t^3)$, $y(\mathbf{x}_t^4)$, τ_t^3, τ_t^4 as well as the committee based novelty detectors $y^c(\mathbf{x}_t)$ and τ_t^c. The committees $y^c(\mathbf{x}_t)$ and τ_t^c were obtained by setting the weights $w_{\tau,t}^a$ and $w_{y,t}^a$ (Eq. 7) to zero, respectively. The vertical dashed line at $t = N = 366$ partitions the data set into the training and test sets. The horizontal line at the zero value represents the threshold used for discriminating between the novel and non-novel data points.

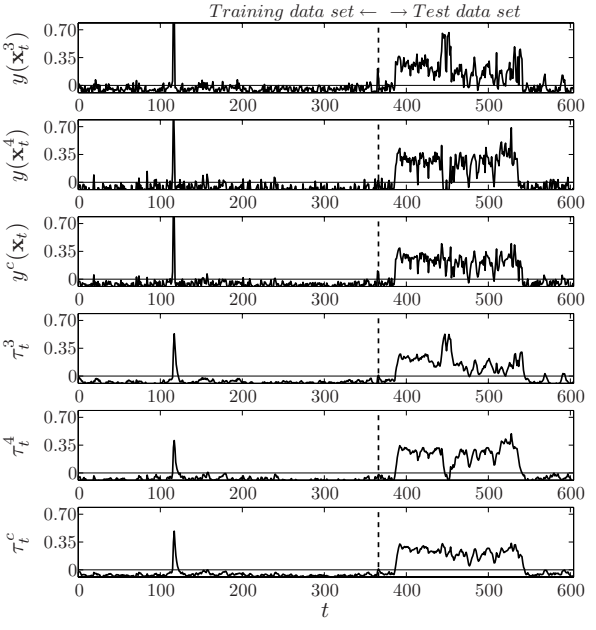


Fig. 1. The output values for the single $y(\mathbf{x}_t^3)$, $y(\mathbf{x}_t^4)$, τ_t^3, τ_t^4 and the committee based novelty detectors $y^c(\mathbf{x}_t)$ and τ_t^c

Table 1 summarizes the test data analysis results from the detection accuracy viewpoint. Four aforementioned parameters TP, FP, TN, FN characterizing the detection accuracy are presented in the Table. As it can be seen from Table 1 and Figure 1, the detectors based on the short term moving average provide a higher detection accuracy than the ones exploiting information from a single data point only. An evident improvement in detection accuracy is obtained when using committees if compered to the single detector cases.

Table 1. Novelty detection accuracy for the test data set

Detector type	Detector output	TP[%]	FP[%]	TN[%]	FN[%]
Single	$y(\mathbf{x}_t^3)$	92.9	7.1	79.5	20.5
Single	$y(\mathbf{x}_t^4)$	90.4	9.6	84.3	15.7
Committee	$y^c(\mathbf{x}_t)$	**97.4**	**2.6**	**86.8**	**13.2**
Single	τ_t^3	98.7	1.3	89.1	10.8
Single	τ_t^4	94.2	5.8	98.8	1.2
Committee	τ_t^c	**100**	**0**	**97.6**	**2.4**

4 Conclusions

We presented a tool to build a set of novelty detectors for a multi-resolution analysis of a time series. Aggregation of several single detectors operating in different frequency regions allowed to substantially improve the detection accuracy. The results of the experimental investigations performed have shown that the detection accuracy obtained is high enough for the approach to be successfully used to screen abnormalities in the paper structure. Steps have been taken to implement the tool at a paper mill in Sweden for monitoring the production process.

References

1. Keller, D. S. and Lewalle, J., Luner, P.: Wavelet Analysis of Simulated Paper Formation. Paperi ja Puu 81(7) (1999) 499–505
2. Starr, K.: Asset Optimization through Data Variation Analysis. Tappi Journal (1)(1) (2001) 1–11
3. Nesic, Z., Davies, M., Dumont, G.: Paper Machine Data Analysis and Compression using Wavelets. Tappi Journal 80(10) (1997) 191–204
4. Timberlake, A., Strom, E.: Do You Know What Causes the Variability in the Paper You Produce?. 2004 Paper Summit, Spring Technical & International Environmental Conference, TAPPI Proceedings (2004)
5. Daubechies, I.: The Wavelet Transform, Time-Frequency Localization and Signal Analysis. IEEE Trans Information Theory 36(5) (1990) 961–967
6. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press. (2004)

Using Directed Acyclic Graph Support Vector Machines with Tabu Search for Classifying Faulty Product Types

Ping-Feng Pai[1] and Yu-Ying Huang[2]

[1] Department of Information Management, National Chi Nan University,
1 University Rd., Puli, Nantou, 545, Taiwan
paipf@nunc.edu.tw
[2] Department of Industrial Engineering and Technology Management, Da-Yeh University,
112 Shan-Jiau Rd., Da-Tusen, Chang-hua, 51505, Taiwan
r9315001@mail.dyu.edu.tw

Abstract. Diagnosing quality faults is one of the most crucial issues in manufacturing processes. Many techniques have been presented to diagnose fault in manufacturing systems. The SVM approach has received more attention due to its classification ability. However, the development of support vector machines (SVM) in the diagnosis of manufacturing systems is rare. Therefore, this investigation attempts to apply the SVM in the diagnosis of manufacturing systems. Furthermore, the tabu search is employed to determine two parameters SVM model correctly and efficiently. A numerical example in the previous literature was used to demonstrate the diagnosis ability of the proposed DSVMT (directed acyclic graph support vector machines with tabu search) model. The experiment results show that the proposed approach can classify the faulty product types correctly.

1 Introduction

An important emerging application of manufacturing process technology is the monitoring and diagnosis of faults in manufacturing systems. Hu et al. [1] developed a fault tree analysis model combined with logic and sequential control systems in monitoring the operational faults of a flexible manufacturing system. The authors concluded that the proposed model can minimize the downtime and maintain an efficient output of a flexible manufacturing system. An intelligent integrated fault-diagnosis system was presented by Hou et al.[2] to monitor the manufacturing process of a belt manufacturing system. The developed system was implemented to an existing textile machinery plant and yielded good results. Khoo et al. [3] integrated graph theory, fuzzy sets and genetic algorithms to diagnose manufacturing systems. An example from the previous literature was used to compare the performance of their hybrid system. The simulation showed that the hybrid diagnosis model led to better results. Son et al. [4] used probabilistic reasoning mechanism to diagnose faults for a variety of production processes. The authors claimed that the presented diagnostic model is suitable for entire manufacturing systems instead of only individual machine. Shen et al. [5] applied the rough set theory to diagnose the value fault of a

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1117 – 1125, 2006.

multi-cylinder diesel engine. They reported that the rough set diagnosis model is suitable for scenarios lacking priori knowledge.

This study tries to investigates the feasibility of SVM in categorizing faulty product types. However, most of the manufacturing system diagnosis problems belong to multi-calss classification categories. Therefore, the directed acyclic graph support vector machines (DAGSVM) [6] is used to deal with multi-calss classification issues. Because of the high computation complexity of DAGSVM model and the cross validation procedure of SVM, the tabu search algorithms [7] are employed to increase the efficiency and correctness of the developed DSVMT model in categorizing faulty product types. The rest of this article is organized as follows. Section 2 introduces the SVM models in classification. The proposed model is presented in Section 3. In Section 4, a numerical example taken from the literature is used to demonstrate the ability of the developed model in classifying faulty product types. Finally, Section 5 draws conclusions.

2 Support Vector Machines in Classification

Originally, SVM was designed for two-class classification. By determining the separate boundary with maximum distance to the closest points of the training data set, SVM obtains a category decision. SVM is able to prevent a possible misclassification efficiently by minimizing structural risk. Consequently, SVM classifier owns better generalization ability than that of trditional classifying approaches.

Supposed a training data set $Tr = \{X_i, Y_i\}_{i=1}^{n}$, where $X_i \in \Re^n$ is the i-th input vector with binary output $Y_i \in \{-1, +1\}$, the classification function is expressed as Eq. (1).

$$Y_i = W^T \Gamma(X_i) + b \,. \tag{1}$$

where $\Gamma: \Re^n \to \Re^m$ is the feature which maps the input space to a high dimensional feature space nonlinearly. The data points are linearly partitioned by a hyperplane defined by the pair ($W \in \Re^m, b \in \Re$) [8]. The optimal hyperplane that separates the data is represented by the following equation.

$$\begin{aligned} &\text{Minimize} && \vartheta(w) = \|W\|^2 / 2 \\ &\text{Subject to} && Y_i \left[W^T \Gamma(X_i) + b \right] \geq 1 \qquad i = 1, \ldots, n \end{aligned} \,. \tag{2}$$

where $\|W\|$ is the norm of a normal weights vector of the hyperplane. This constrained optimization problem is obtained by a primal Lagrangian form formulated as Eq. (3):

$$L(W, b, \alpha) = \frac{1}{2}\|W\|^2 - \sum_{i=1}^{n} \alpha_i \left[Y_i (W^T \Gamma(X_i) + b) - 1 \right] . \tag{3}$$

where α_i represent Lagrange multipliers. Using Karush-Kuhn-Tucker conditions, the solutions of the dual Lagrangian problem , α_i^*, determine the parameters w_* and b_* of the optimal hyperplane. Finally, the decision function is depicted by Eq. (4):

$$D(X_i) = \mathrm{sgn}\left(W_0^T \Gamma(x_i) + b_0\right) = \mathrm{sgn}\left(\sum_{i=1}^{n} \alpha_i^* Y_i K(X, X_i) + b_0\right), \quad i = 1, \ldots, N \quad (4)$$

Any function satisfies Mercer's conditions [8] is a good candidate for the kernel function, $K(X,X_i)$. Besides, the value of $K(X,X_i)$ is expressed as the inner product of two vectors X and X_i in the feature space. In this work, the Gaussian radical basis function is used for the SVM classifier model.

However, in many situations, the ability to categorize multiple-class is required for SVM model. Therefore, some multi-class classification techniques have been proposed for SVM models. The most typical approach for multi-class problems is the so called one-versus-rest (1-v-r) method classifying one class from the other class [8], [9], [10]**.** The disadvantage of the 1-v-r method is the learning time scales linearly with the number of classes. One-versus–one (1-v-1) combining all possible two-class classifier is another methodology for dealing with multi-class problems [11], [12], [13]. Knerr et al. [12] combined the "AND" gate with the 1-v-1 approach. Friedman [11] and KreBel [13] used the Max Wins algorithm to enhance the 1-v-1 method. However, the drawback of the 1-v-1 approach is the size of classifier grows super-linearly with the number of classes. Therefore, the execution time of 1-v-1 approach is much worse than the 1-v-r method. Proposed by Platt et al. [6], DAGSVM (Directed Acyclic Graph Support Vector Machines, **Fig.1**) is one of the most popular approaches for multi-class classification methods. The training stage of the DAGSVM is the same as 1-v-1 model. However, the DAGSVM method uses a rooted binary directed acyclic graph to test the model. Therefore, the testing time of using DAGSVM model is less than that of the 1-v-1 approach. Dong et al. [14] presented a novel multi-class SVM approach by considering all classes at once. The authors reported that the developed model is superior to the 1-v-a as well as the 1-v-r techniques in classifying the Iris and Glass data sets. By using large data set, Hsu and Lin [15] compared the classification performance of multi-class SVM methods based on the binary classification concept and methods solving multi-class SVM in one step. Experimental results show that the classification approaches based on binary classification are more suitable for practical use, and the classification technique considering all data at once need fewer support vectors. Angulo et al.[16] introduced a support vector classification-regression machine by embodying new training algorithms with ternary outputs into the traditional SVM model. Three data sets, Iris, Wine, and Glass were used to demonstrate the classification accuracy of the proposed model. The experimental results reveal that the developed model outperforms the other three conventional SVM-based approaches. Liu et al. [17] presented a multistage SVM technique combining many two-class classifiers into one multi-class classifier to conduct multi-class pattern recognition problems. Iris data set and Crab data set were used to show the classification performance of the multistage SVM technique. Simulation results indicate that the multistage SVM technique outperforms1-v-1, 1-v-r, and DAGSVM methods. Maruyama et al. [18] developed a hybrid multi-class SVM by exploiting the unique

strength of the DAGSVM model and the Max-Win algorithm. The proposed model was reported to greatly improve the classification performance of the single 1-v-r model, 1-v-1 model, and DAGSVM model. Statnikov et al. [19] had a comprehensive evaluation of multi-class classification techniques including several SVM-based approaches and three non-SVM methods in cancer diagnosis. The experimental results reveal that all SVM-based approaches outperform the non-SVM techniques in classification accuracies and the relative classifier information. However, the non-SVM techniques have less execution time in some cases. Pöyhönen et al. [20] compared the classification performance of three SVM-based approaches with the multi-layer perceptorn (MLP) neural network in fault diagnosis of cage motor. They reported that the SVM with mixture matrix coupling strategy achieve almost the same classification as MLP neural networks. However, the training and tuning of MLP neural networks is an exhausting task. Therefore, the authors concluded that the SVM with mixture matrix is a practical choice in diagnosing cage motor systems.

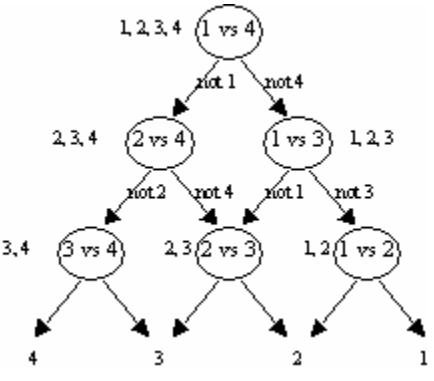


Fig. 1. The decision DAG for finding the best class out of four classes [12]

3 Directed Acyclic Graph Support Vector Machines with Tabu Search

For the traditional DAGSVM model, one SVM parameter is fixed then the other parameter is adjusted to decrease the classification error. However, in the DAGSVM structure, k(k-1)/2 SVMs are conducted. The k is the number of classifications. Furthermore, the cross-validation technique is adopted to prevent the overfitting of classification. The parameter selection is performed for each cross validation. Therefore, the computation process is very time consuming when the traditional parameter selection approach is employed in the DAGSVM model. In this investigation, the tabu search algorithms were employed to determine two parameters of SVMs simultaneously and sequentially in the DAGSVM model. The proposed DSVMT model is illustrated as Fig.2.

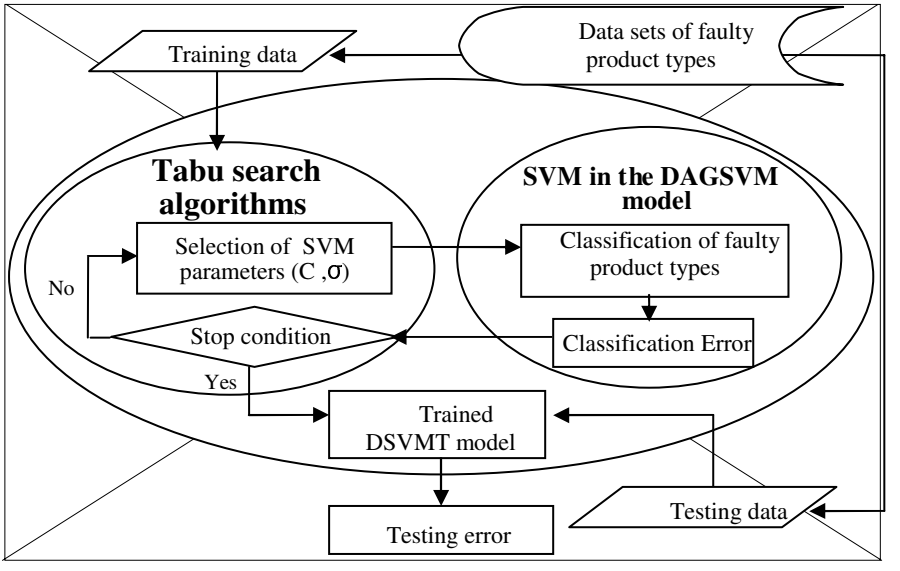


Fig. 2. The flow chart of Tabu search algorithms in the DSVMT model

The detailed procedure of applying tabu search algorithms in determining SVM parameters is described as follows.

Step 1: Generation of initial solutions

The initial solutions, represented by C and σ, are generated randomly. In this investigation, the boundaries of C and σ are from 0 to 10 and from 0 to 5. Additionally, the number of initial solutions is twenty.

Step 2: Evaluation of incumbent solutions

The SVM classification error is used as an index to evaluate the quality of incumbent solutions. A smaller classification error indicates a better combination of two SVM parameters (C,σ). For the first iteration, the twenty solutions are treated as incumbent solutions and recorded in the tabu list.

Step 3: Move selection

The best solution of incumbent solutions is the "move" of this iteration.

Step 4: Tabu list

Compare the selected move in step 3 with solutions in the TABU list. If the move is the same as one of the solution in the tabu list then go back to step1. Otherwise, include this move into the tabu list. The forbidden moves are removed from the tabu list when a predetermined number of iterations is reached. In this investigation, the predetermined number of iteration is thirty.

Step 5: The stopping condition

The tabu search is terminated when the classification error is not reduced after fifty iterations.

Additionally, the cross-validation method is applied to prevent the overfitting of classification. Here, we use 3-fold cross validation [21] in the developed DSVMT model. The training data set is divided into 3 parts with equal size. One third of the data are used for estimating the SVM. The other data are applied to determine the parameters of SVM.

4 A Numerical Example

An example of manufacturing process presented by Hou et al. [2] was used to demonstrate the diagnosis performance of the proposed DSVMT model. The illustrated example is a manufacturing process of industrial conveyor belts. Eight key factors including processing speed, three measurements of belt thickness and four types of temperature determine the faulty product types of the manufacturing systems.

In this example, the fault types are classified into five categories. Table 1 depicts the diagnosis results by applying the DSVMT model to the manufacturing process. It is shown that the proposed DSVMT can diagnose the faulty products correctly and efficiently. Table 4 shows the correct percentage when the tabu search algorithms were not employed to select the parameters for DAGSVM model. For both cases, the computation times are performed on a Penttium IV 1.5GHz personal computer. In this experiment, it is indicated that the proposed DSVMT model can achieve more correct diagnosis results with less computation time.

Table 1. The training data sets of the numerical example [2]

Data	Speed	Thick1	Thick2	Thick3	Temp.1	Temp.2	Temp.3	Temp.4	Fault
1	3.72	5.58	5.58	5.6	77.8	82.1	80.6	80.6	1
2	3.62	5.58	5.58	5.61	76.7	82.3	81.6	80.6	7
3	3.58	5.79	5.62	5.6	76.6	82.3	81.6	80.6	7
4	3.42	5.67	5.61	5.65	76.6	82.1	80.5	80.6	3
5	3.6	5.61	5.61	5.65	76.7	82.3	81.6	80.7	7
6	3.42	5.64	5.62	5.65	76.7	82.1	80.3	80.7	3
7	3.58	5.58	5.63	5.45	76.4	82.4	80.3	80.6	9
8	3.72	5.57	5.59	5.59	77.8	82.1	80.3	80.7	1
9	3.62	5.65	5.75	5.43	76.7	82.4	80.1	80.6	9
10	3.6	5.65	5.75	5.43	76.6	82.4	80.1	80.7	9
11	3.42	5.57	5.59	5.58	76.8	82.1	80.3	80.7	3
12	3.6	5.61	5.64	5.45	76.4	82.4	80.6	81.7	9
13	3.42	5.57	5.58	5.61	76.7	82.1	80.6	80.6	3
14	3.6	5.58	5.68	5.59	76.8	82.3	81.6	81.7	7
15	3.72	5.79	5.61	5.58	77.8	82.1	80.5	80.7	1
16	3.6	5.64	5.61	5.61	76.8	82.3	81.6	80.6	7
17	3.42	5.6	5.65	5.58	76.7	82.2	80.6	80.7	3
18	3.72	5.6	5.65	5.58	77.8	82.2	80.8	80.6	1
19	3.58	5.54	5.58	5.43	76.8	82.4	80.8	80.1	9
20	3.6	5.59	5.68	5.43	76.7	83.4	80.7	80.7	9
21	3.58	5.51	5.52	5.43	76.4	83.3	80.8	80.7	9
22	3.6	5.67	5.61	5.61	76.8	82.3	81.6	80.6	7
23	3.42	5.61	5.58	5.6	76.7	82.2	80.6	80.6	3
24	3.6	5.57	5.58	5.59	76.7	82.2	80.5	80.7	0
25	3.62	5.58	5.6	5.6	76.8	82.1	80.6	80.7	0
26	3.61	5.6	5.61	5.61	76.6	82.2	80.5	80.6	0
27	3.62	5.58	5.61	5.59	76.6	82.1	80.6	80.6	0

Table 2. The testing data sets of the numerical example [2]

Data	Speed	Thick1	Thick2	Thick3	Temp.1	Temp.2	Temp.3	Temp.4	Fault
1	3.6	5.61	5.65	5.45	76.4	83.3	80.6	80.7	9
2	3.42	5.6	5.61	5.61	76.7	82.1	80.6	80.6	3
3	3.6	5.61	5.61	5.58	76.8	82.5	81.6	80.7	7
4	3.75	5.61	5.61	5.58	77.8	82.1	80.6	80.7	1
5	3.6	5.62	5.62	5.62	76.8	82.5	81.6	80.6	7
6	3.42	5.62	5.62	5.58	76.7	82.3	80.6	80.7	3
7	3.75	5.6	5.61	5.61	77.8	82.3	80.8	80.6	1
8	3.6	5.59	5.61	5.43	76.7	83.3	80.8	80.1	9
9	3.6	5.59	5.58	5.42	76.7	83.3	80.7	80.7	9
10	3.6	5.51	5.52	5.42	76.4	83.3	80.8	80.7	9
11	3.6	5.62	5.58	5.62	76.8	82.5	81.6	80.6	7
12	3.43	5.61	5.58	5.62	76.8	82.3	80.6	80.6	3
13	3.6	5.61	5.59	5.62	76.8	82.1	80.6	80.6	0
14	3.6	5.6	5.63	5.62	76.8	82.3	80.6	80.6	0
15	3.61	5.61	5.61	5.61	76.7	82.1	80.6	80.6	0

Table 3. Diagnosis performance and associated parameters by the DSVMT model

Parameters of i-th SVM in the DSVMT model		
i	σ	C
1	1.668	6.230
2	3.649	7.236
3	0.316	4.996
4	2.935	0.576
5	1.011	2.885
6	4.824	8.242
7	0.295	1.599
8	1.099	8.597
9	3.122	4.213
10	1.369	1.850
Computation time		1'38"77
Correct percentage		100%

Table 4. Diagnosis performance and associated parameters without applying tabu search in the DSVMT model

Parameters of i-th SVM in the DSVMT model		
i	σ	C
1	0.8	1
2	0.2	1
3	0.2	1
4	1.6	1
5	0.2	1
6	1.6	3
7	0.2	1
8	0.2	1
9	0.2	1
10	0.2	1
Computation time		2'59"08
Correct percentage		93.33%

5 Conclusions

In this investigation, a directed acyclic graph support vector machines with tabu search (DSVMT) model is developed to diagnose faulty product types in a belt manufacturing process. Due to the high computation complexity of the DAGSVM model as well as the cross validation approach of the DSVMT model, the tabu search algorithms are employed to determine SVM parameters. By using a numerical example in the previous literature, the proposed DSVMT model can categorize the faulty product types correctly and efficiently. For the future research, more complex manufacturing process data could be used to test the performance of the proposed model. In addition, the presented DAGSVM model could be applied to the other classification problems in the medical domain and business systems.

Acknowledgement

This research was conducted with the support of National Science Council (NSC 94-2213-E-260-023).

References

1. Hu, W., Starr, A.G., Leung, A.Y.T.: Operational Fault Diagnosis of Manufacturing System. Journal of Materials Processing Technology 133 (2003) 108-117
2. Hou, T.H., Liu, W.L.: Intelligent Remote Monitoring and Diagnosis of Manufacturing Processes Using an Integrated Approach of Neural Networks and Rough Sets. Journal of Intelligent Manufacturing 14 (2003) 239-253
3. Khoo, L.P., Ang, C.L., Zhang, J.: A Fuzzy-based Genetic Approach to the Diagnosis of Manufacturing Systems. Engineering Applications of Artificial Intelligence 13 (2000) 303-310
4. Son, J.P., Park, J.H., Cho, Y.Z.: An Integrated Knowledge Representation Scheme and Query Processing Machanism for Fault Diagnosis in Heterogeneous Manufacturing Environments. Robotics and Computer Integrated Manufacturing 16 (2000) 133-141
5. Shen, L., Tay, F.E.H., Qu, L., Shen, Y.: Fault Diagnosis Using Rough Sets Theory. Computers in Industry 43 (2000) 61-72
6. Platt, J.C., John, S-T., Nello, C.: Large Margin DAGs for Multiclass Classification. in Proceedings of Neural Information Processing Systems 12 (2000) 547-553
7. Glover, F.: Future Paths for Integer Programming and Links to Artificial Intelligence. Computers and Operations Research 13 (1986) 533-549
8. Vapnik, V. (eds.): The Nature of Statistical Learning Theory. Springer-Verlag, New York (1995)
9. Cortes, C., Vapnik, V.: Support Vector Network. Machine Learning 20 (1995) 273-297
10. Bottou, L., Cortes, C., Denker, J., Drucker, H., Guyon, I., Jackel, L., LeCun, Y., Muller, U., Sackinger, E., Simard, P., Vapnik, V.: Comparison of Classifier Methods: A Case Study in Handwriting Digit Recognition. in Proc. Int. Conf. Pattern Recognition (1994) 77-87
11. Friedman, J.: Another Approach to Polychotomous Classification. Stanford University (1996)

12. Knerr, S., Personnaz, L., Dreyfus, G.: Single-layer Learning Revisited: A Stepwise Procedure for Building and Training a Neural Network. in Neurocomputing: Algorithms, Architectures and Applications, Fogelman, J. (eds.), New York: Springer-Verlag,(1990)
13. KreBel, U.: Pairwise Classification and Support Vector Machines. in Advances in Kernel Methods—Support Vector Learning, Schölkopf B., Burges, C.J.C, Smola, A.J. (ed.), Cambridge, MA: MIT Press,(1999) 255-268
14. Dong, X., Zhaohui, W., Yunhe P.: A New Multi-class Support Vector Machines. IEEE. (2001) 1673-1676
15. Hsu, C.W., Lin C.J.: A Comparison of Methods for Multiclass Support Vector Machines. IEEE Transactions on Neural Networks. 13 (2002) 415-425
16. Angulo, C., Parra, X., Catalá, A.: K-SVCR. A Support Vector Machine for Multi-class Classification. Neurocomputing. 55 (2003) 57-77
17. Liu, X.P., Xing, H.J., Wang, X.Z.: A Multistage Support Vector Machine. in Proceedings of the Second International Conference on Machine Learning and Cybernetics.(2003)
18. Maruyama, K.I., Maruyama, M., Miyao, H., Nakano, Y.: A Method to Make Multiple Hypotheses with High Cumulative Recognition Rate Using SVMs. Pattern Recognition. 37 (2004) 241-251
19. Statnikov, A., Aliferis, C.F., Tsamardino, I., Hardin, D., Levy, S.: A Comprehensive Evaluation of Multicategory Classification Methods for Microarray Gene Expression Cancer Diagnosis. Bioinformatics Advance Access published September 16 (2004) 1-14
20. Pöyhönen, S., Arkkio, A., Jover, P., Hyötyniemi, H.: Coupling Pairwise Support Vector Machines for Fault Classification. Control Engineeing Practice. 13 (2005) 759-769
21. Burman, P.: A Comparative Study of Ordinary Cross-validation, V-fold Cross-validation and the Repeated Learning-testing Methods. Biometrika. 76 (1989) 503-514

Product Quality Prediction with Support Vector Machines

Xinggao Liu

National Laboratory of Industrial Control Technology,
Department of Control Science and Technology,
Zhejiang University,
Hangzhou 310027, P.R. China
liuxg@iipc.zju.edu.cn

Abstract. Reliable prediction of melt index (MI) is crucial in practical propylene polymerization processes. In this paper, a least squares support vector machines (LS-SVM) soft-sensor model is developed first to infer the MI of polypropylene from other process variables. A weighted least squares support vector machines (weighted LS-SVM) approach is further proposed to obtain rather robust estimate. Detailed comparative researches are carried out among standard SVM, LS-SVM, and weighted LS-SVM. The research results confirm the effectiveness of the presented methods.

1 Introduction

Production of polypropylene is a multi-billion business with annual consumption growth rate about 5%. Melt index (MI), the most important parameter in determining the product's grade and quality control of polypropylene, is usually evaluated off-line with an analytical procedure that takes almost 1 hour to complete in the laboratory. An alternative is to develop on-line estimators [1] based on available process information to supervise the process and avoid any mismatch of product quality. Rallo [2] provided a fuzzy ARTMAP neural system and two hybrid networks to infer the melt index of six different LDPE grades. Han [3] compared the performance of support vector machines, partial least squares, and artificial neural networks for MI estimation of San and PP processes. They concluded that the standard SVM yields the best prediction among the three toward the studied problems. Unfortunately, further research about SVM on this topic hasn't been carried out.

A LS-SVM model is therefore developed to infer the MI of polypropylene from other readily measurable process variables. The weighted LS-SVM approach is further presented, as it is well known that the use of a SSE cost function without regularization, as it is in the case with LS-SVM, might lead to estimates which are less robust, e.g. with respect to outliers on the data or when the underlying assumption of a Gaussian distribution for the error variables is not realistic. Up to now, litter, however, has appeared in the literature on these matters.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1126 – 1131, 2006.

2 The Support Vector Method of Function Estimation

2.1 Standard Support Vector Machines

SVMs can be applied to regression problems by the introduction of an alternative loss function. Consider regression in the following set of functions

$$f(x) = w^T \varphi(x) + b \tag{1}$$

with given training data $\{x_i, y_i\}_{i=1}^{M}$, where M denotes the number of training data, x_i is the input data and y_i is the output data. The nonlinear mapping φ maps the input data into a high dimensional feature space where a linear regression problem is obtained and solved. In the support vector method one aims at minimizing the regularized risk

$$R(w,b) = \gamma \frac{1}{M} \sum_{i=1}^{M} L_\varepsilon(y_i, f(x_i)) + \frac{1}{2} w^T w \tag{2}$$

$$L_\varepsilon(y_i, f(x_i)) = \begin{cases} 0, & |y - f(x)| \le \varepsilon \\ |y - f(x)| - \varepsilon, & \text{otherwise} \end{cases} \tag{3}$$

In Eq. (2), L_ε is the so-called ε-insensitive loss function, which indicates that it does not penalize errors below ε. The second term, $\frac{1}{2} w^T w$, is used as a flatness measurement of function (1) and γ is a regularized constant determining the tradeoff between the training error and the model flatness. The estimation problem is formulated then as the optimization problem

$$\min_{w,b,\xi^*,\xi} R(w, \xi^*, \xi) = \frac{1}{2} w^T w + \gamma \{ \sum_{i=1}^{M} \xi_i^* + \sum_{i=1}^{M} \xi_i^* \} \tag{4}$$

subject to the constraints

$$\begin{cases} y_i - w^T \varphi(x_i) - b \le \varepsilon + \xi_i^*, & i = 1, \cdots, M \\ -y_i + w^T \varphi(x_i) + b \le \varepsilon + \xi_i, & i = 1, \cdots, M \\ \xi_i^*, \xi_i \ge 0, & i = 1, \cdots, M \end{cases} \tag{5}$$

where ξ, ξ^* are slack variables. One obtains $w = \sum_{i=1}^{M} (\alpha_i^* - \alpha_i) \varphi(x_i)$, where α^*, α are obtained by solving a quadratic program and are the Lagrange multipliers related to the first and second set of constraints.

2.2 Least Squares Support Vector Machines

Least squares version [4] of the support vector method corresponds to the following form of ridge regression

$$\min_{w,b,\xi} R(w,b,\xi) = \frac{1}{2} w^T w + \frac{1}{2} \gamma \sum_{i=1}^{M} \xi_i^2 \tag{6}$$

subject to the equality constraints

$$y_i = w^T \varphi(x_i) + b + \xi_i, \quad i = 1, \cdots, M \tag{7}$$

One defines the Lagrangian

$$L(w,b,\xi,\alpha) = R(w,b,\xi) - \sum_{i=1}^{M} \alpha_i (w^T \varphi(x_i) + b + \xi_i - y_i) \tag{8}$$

with α_i Lagrange multipliers. The conditions for optimality

$$\begin{cases} \dfrac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{i=1}^{M} \alpha_i \varphi(x_i) \\ \dfrac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^{M} \alpha_i = 0 \\ \dfrac{\partial L}{\partial b} = 0 \rightarrow \alpha_i = \gamma \xi_i \qquad\qquad i = 1, \cdots, M \\ \dfrac{\partial L}{\partial b} = 0 \rightarrow w^T \varphi(x_i) + b + \xi_i - y_i = 0 \quad i = 1, \cdots, M \end{cases} \tag{9}$$

can be written as the solution to the following set of linear equations after elimination of w and ξ_i

$$\begin{bmatrix} 0 & 1_v^T \\ 1_v & K + \gamma^{-1} I \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \tag{10}$$

with $y = [y_1, \cdots, y_M]$, $1_v = [1, \cdots, 1]$, $\alpha = [\alpha_1, \cdots, \alpha_M]$ and I is a identity matrix.

The resulting LS-SVM model for function estimation becomes

$$f(x) = \sum_{i=1}^{M} \alpha_i K(x, x_i) + b \tag{11}$$

.

2.3 Weighted Least Squares Support Vector Machines

In order to obtain a robust estimate [5] based upon the previous LS-SVM solution, the error variables $\xi_i = \alpha_i / \gamma$ by weighting factors v_i is weighted, which leads to the optimization problem:

$$\min_{w^*,b^*,\xi^*} R(w^*,\xi^*) = \frac{1}{2} w^{*T} w^* + \frac{1}{2}\gamma \sum_{i=1}^{M} v_i \xi_i^{*2} \tag{12}$$

Subject to the equality constraints and the Lagrangian becomes

$$y_i = w^{*T}\varphi(x_i) + b^* + \xi_i^*, \quad i = 1, \cdots, M \tag{13}$$

$$L(w^*,b^*,\xi^*,\alpha^*) = R(w^*,\xi^*) - \sum_{i=1}^{M} \alpha_i^* \{w^{*T}\varphi(x_i) + b^* + \xi_i^* - y_i\} \tag{14}$$

The unknown variables for this weighted LS-SVM problem are denoted by the * symbol. From the conditions for optimality and elimination of w^*, ξ^*, one obtains the KKT system

$$\begin{bmatrix} 0 & 1_v^T \\ 1_v & K + V_\gamma \end{bmatrix} \begin{bmatrix} b^* \\ \alpha^* \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \tag{15}$$

where the diagonal matrix V is given by

$$V_\gamma = \mathrm{diag}\{\frac{1}{\gamma v_1}, \ldots, \frac{1}{\gamma v_M}\} \tag{16}$$

The choice of the weights v_i is determined based upon the error variables $\xi_i = \alpha_i / \gamma$ from the LS-SVM case (10). Robust estimates are obtained then by taking

$$v_i = \begin{cases} 1 & \text{if } |\xi_i / \hat{s}| \le c_1 \\ \dfrac{c_2 - |\xi_i / \hat{s}|}{c_2 - c_1} & \text{if } c_1 \le |\xi_i / \hat{s}| \le c_2 \\ 10^{-4} & \text{otherwise} \end{cases} \tag{17}$$

where $\hat{s}$ is a robust estimate of the standard deviation of the LS-SVM error variables ξ_i:

$$\hat{s} = \frac{IQR}{2 \times 0.6745} \tag{18}$$

The interquartile range, IQR, is the difference between the 75th percentile and 25th percentile. The constants c_1, c_2 are typically chosen [6] as c_1 = 2.5 and c_2 = 3. Eventually, the procedure (12) (17) can be repeated iteratively.

3 Results and Discussion

The modeling data used for training and validating the soft-sensor have been acquired from the historical logs recorded in a real propylene polymerization plant. Data from

the records of the process variables and MI are separated into training, test and generalization sets that are constructed from the time series of recorded plant data. And the test set is obtained from the same batch as the training set, while the generalization set is derived from another batch.

The standard SVM proposed by Han [3] is also built as a basis of comparison research. The difference between the output of the models and the desired output is referred to as the error and can be measured in different ways. Here, mean absolute error (MAE), mean relative error (MRE), root mean squared error (RMSE), standard deviation (STD), Theil's Inequality Coefficient (TIC) [6], are employed as derivation measurements between measured and predicted values. There is

$$TIC = \frac{\sqrt{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}}{\sqrt{\sum_{i=1}^{N} y_i^2} + \sqrt{\sum_{i=1}^{N} \hat{y}_i^2}} \tag{19}$$

where y_i, $\hat{y}_i$ denote the measured value and predicted result.

Table 1. Performance for the testing data sets

Methods	MAE	MRE (%)	RMSE	STD	TIC
Weighted LS-SVM	0.0754	3.27	0.0198	0.1055	0.0223
LS-SVM	0.0842	3.66	0.0214	0.1116	0.0240
SVM	0.1105	4.80	0.0274	0.1394	0.0307

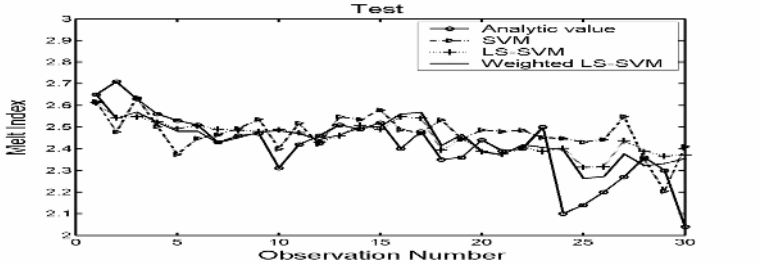


Fig. 1. Comparison of predictions between LS-SVM and standard SVM

The detailed comparison of test performance among weighted LS-SVM, LS-SVM and standard SVM is listed in Table 1. It is shown that the weighted LS-SVM model functions best on the overall, with mean absolute error of 0.0754, compared with those of 0.0842 and 0.1105 obtained from the corresponding LS-SVM and SVM models respectively. The RMSE confirms the prediction accuracy of the proposed methods. Weighted LS-SVM yields the smallest STD among three, which indicates the predictive stability of the method. TIC of weighted LS-SVM is quite acceptable compared to those of standard SVM and LS-SVM, which indicates a good level of agreement between the proposed model and the studied process. A visual impression of

the agreement between the measured MI and the models output is shown in Fig. 1, where the weighted LS-SVM model yields consistently good predictions.

4 Conclusions

LS-SVM and weighted LS-SVM methods are proposed to infer MI of polypropylene from other process variables. The weighted LS-SVM model predicts MI with mean relative error of approximately 3.27%, compared with those of 3.66% and 4.80% obtained from the corresponding LS-SVM and SVM models respectively. The results indicate that the proposed method provides prediction reliability and accuracy and supposed to have promising potential for practical use.

Acknowledgements

This work is supported by Zhejiang Provincial Natural Science Foundation of China (Grant Y105370), National Natural Science Foundation of China (Grant 20106008), National HI-TECH Industrialization Program of China (Grant Fagai-Gaoji-2004-2080) and Science Fund for Distinguished Young Scholars of Zhejiang University (Grant 111000-581645), and their supports are thereby acknowledged.

References

1. Hunt, K.J., Sbarbaro, D., Zbikowski, R., Gawthrop, P.J.: Neural Networks for Control Systems — a Survey. Automatica 28(6) (1992) 1083-1112
2. Rallo, R., Ferre-Giné, J., Arenas, A., Giralt, F.: Neural Virtual Sensor for the Inferential Prediction of Product Quality from Process Variables. Comp. Chem. Eng. 26(12) (2002) 1735-1754
3. Han, I.-S., Han, C., Chung, C.-B.: Melt Index Modeling with Support Vector Machines, Partial Least Squares, and Artificial Neural Networks. J. Appl. Polym. Sci. 95(4) (2005) 967-974
4. Suykens, J.A.K., Vandewalle, J.: Least Squares Support Vector Machine Classifiers. Neural Proc. Letters 9(3) (1999) 293-300
5. Suykens, J.A.K., Vandewalle, J.: Weighted Least Squares Support Vector Machines: Robustness and Sparse Approximation. Neurocomputing 48(1) (2002) 85-105
6. Smith, D.J.M.: Methods for External Validation of Continuous System Simulation Models: a Review. Math. Comput. Model Dyn. Syst. 4(1) (1998) 5-31

Hierarchical Neural Network Based Product Quality Prediction of Industrial Ethylene Pyrolysis Process

Qiang Zhou[1], Zhihua Xiong[1], Jie Zhang[2], and Yongmao Xu[1]

[1] Department of Automation, Tsinghua University, Beijng, 100084, P.R. China
zhxiong@tsinghua.edu.cn
[2] School of Chemical Engineering and Advanced Materials,
University of Newcastle,
Newcastle upon Tyne, NE1 7RU, UK
jie.zhang@ncl.ac.uk

Abstract. A two-layer hierarchical neural network is proposed to predict the product qualities of an industrial KTI GK-V ethylene pyrolysis process. The first layer of the model is used to classify these changes into different operating conditions. In the second layer, the process under each operating condition is modeled using bootstrap aggregated neural networks (BANN) with sequential training algorithm. The overall output is obtained by combining all the trained networks. Results of application to the actual process show that the proposed soft-sensing model possesses good generalization capability.

1 Introduction

Ethylene pyrolysis is a key and fundamental process in the petrochemical industry. It is necessary to on-line obtain the product yields and the cracking degree not only for advanced control but also for optimization and monitoring of the process [1]. To overcome the difficulties in building mechanistic models, empirical models based on process input-output data can be utilized. Neural networks (NNs) have been shown to be able to approximate any continuous nonlinear functions and have been applied to nonlinear process modeling and control [2]. If properly trained and validated, these NN models can be used to predict steady-state and dynamic process behavior reasonably well, hence, leading to improved process optimization and control performance [3]. Multiple networks emerge as a powerful solution to improve the generalization capability, because significant improvement in model generalization can be achieved when the individual networks to be combined are independent or less correlated [4][5]. For a BANN model, less correlated neural network models can be obtained by using bootstrap re-sampling to generate training data for individual networks so that they are trained on different data sets [6].

In practice, the operation condition of ethylene pyrolysis process usually varies due to the changes of feedstock properties and the formation of coke in the cracking coils. Zhou and Xu [1] showed that classification of the operation conditions using data clustering method followed with modeling of each operation condition using a single neural network would improve the prediction accuracy. In this study, the focus is on building a hierarchical neural network to cope with different operation conditions and to improve the performance of model predictions by combining multiple networks.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1132 – 1137, 2006.

2 Two-Layer Hierarchical Soft-Sensing Model

In this study, a practical KTI GK-V pyrolysis process in an ethylene plant is considered. An ethylene pyrolysis furnace contains two sequential parts: a convection box and a fire box. Hydrogenation residue oil is fed to the coils inside the convection box and mixed with diluted steam on the top, pre-heated by the air from the fire box, and then enter the coils inside the fire box. The residue oil is cracked at high temperature in the fire box. The ethylene product is obtained in the cracked gas at the outlet of fire box. The production yield is influenced strongly by the cracking degree, which represents the sufficiency of the cracking reaction happened in the coils. In the cracking degree, rate of methane to propylene (CH_4/C_3H_6) and rate of ethylene to propylene (C_2H_4/C_3H_6) are two important indices, but they cannot be measured on-line. The aim of the soft-sensing model is to predict these two product qualities.

From a large number of historical operation data of the ethylene pyrolysis process, it can be observed that the operation condition usually varies due to the changes of feedstock properties and the formation of coke in the cracking coils. Hierarchical models can be used to cope with these variations of process operating conditions [7]. A two-layer hierarchical model is proposed to build the models of product qualities, as shown in Fig. 1. In the first layer, data set are pre-processed and then classified into different operation conditions. In the second layer, each operation condition is modeled using a BANN with sequential training algorithm. And the neural network models are selected according to the process operating conditions when implemented on-line. To improve the model accuracy, the model outputs are also corrected on-line.

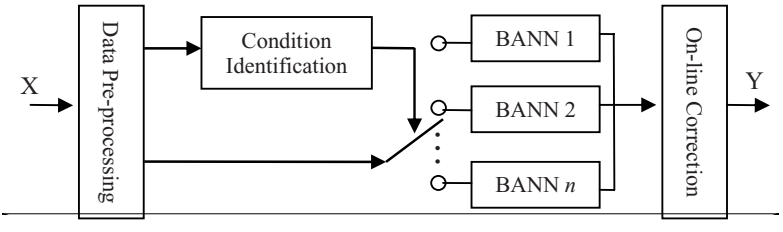


Fig. 1. A two-layer hierarchical model based on multiple BANNs

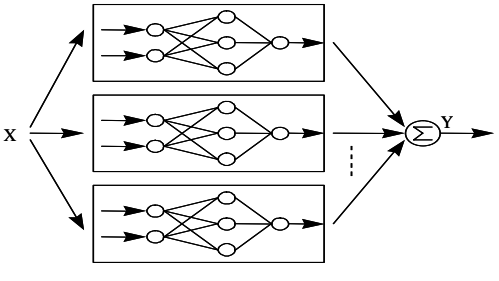


Fig. 2. A bootstrapped aggregated neural network (BANN)

The structure of BANN model is shown in Fig. 2. It can be seen that several neural networks are developed to model the same relationship and then combined together to improve the model generalization capability [5]. The overall output of the ensemble networks can be computed by

$$F(x) = \frac{1}{L}\sum_{i=1}^{L} F_i(x) \tag{1}$$

where L is the number of networks, F represents the output, F_i represents the ith individual network, and x is a vector of network inputs, respectively. Here BANN model is trained by using the sequential training algorithm [6]. The first network is trained to minimize its prediction error, whereas the rest of the networks are trained not only to minimize their prediction errors but also to minimize the correlation among the trained networks. For the training of the first network, the aim is to minimize the prediction errors, which is defined as

$$E_1 = \sum_{j=1}^{N} [F_1(x_j) - T(x_j)]^2 \tag{2}$$

where N is the number of training samples, T is the desired model output, and x_j is the jth input training data sample, respectively. For the training of the ith network, the objective is to minimize not only the individual networks prediction errors but also the correlation among the individual networks, which are defined as

$$E_i = \sum_{j=1}^{N} \left\{ \left[F_i(x_j) - T(x_j)\right]^2 - \lambda \left[F_i(x_j) - F(x_j)\right]^2 \right\} \tag{3}$$

where

$$F_i(x) = \frac{1}{i}\sum_{k=1}^{i-1} F_k(x) \tag{4}$$

and λ is a weighting parameter between 0 and 1. In Eq(3), the first term minimizes the prediction errors of the ith network and the second term minimizes the correlation between the ith neural network and the previously trained networks. Here a back-propagation training algorithm is used to train the BANN, and the network weights are adjusted as follows

$$W_i(k+1) = W_i(k) + \Delta W_i(k+1) \tag{5}$$

$$\Delta W_i(k+1) = \alpha W_i(k) - \eta \frac{\partial E_i}{\partial W_i(k)} \tag{6}$$

$$\begin{aligned} \frac{\partial E_i}{\partial W_i} &= 2\sum_{j=1}^{N} (F_i(x_j) - T(x_j)) \frac{\partial F_i(x_j)}{\partial W_i} - 2\lambda \sum_{j=1}^{N} (F_i(x_j) - F(x_j)) \frac{\partial F_i(x_j)}{\partial W_i} \\ &= 2\sum_{j=1}^{N} \{(1-\lambda) F_i(x_j) - T(x_j) + F(x_j))\} \frac{\partial F_i(x_j)}{\partial W_i} \end{aligned} \tag{7}$$

where $W(k)$ and $\Delta W(k)$ are the vectors of weights and weight adaptations at the training step k, α is the momentum coefficient, and η is the learning rate, respectively. $\partial F_i(x_j)/\partial W_i$ can also be calculated easily according to the topology of the individual network. Training can be terminated when the error gradient is less than a pre-

specified value or terminated by a cross validation based "early stopping" criterion [5][6]. When using "early stopping", data are divided into a training data set and a testing data set. During network training, the network prediction error on the testing data is continuously monitored. Training of an individual network is terminated when the testing error stops decreasing. Training of the aggregated network can be terminated when the aggregated network performance cannot be further improved.

When applied to on-line prediction in the ethylene plant, the model predictions should be on-line corrected due to the presence of unknown disturbances. In this study, the correction method was implemented by adding the average model prediction errors of the previous several samples to the current prediction value.

3 Application Results

In this study, the two-layer hierarchical model is used to predict the cracking degree (CH_4/C_3H_6 and C_2H_4/C_3H_6). Over 7000 data samples were obtained from the plant and used to build the model. The ethylene cracking process is very complicated with over one hundred correlated variables that can be measured on-line.

During the data pre-processing, any outliers or bad samples were first verified and deleted, then the data were filtered. The delays of the analyzed values of cracked gas were also considered. Finally all data were scaled to zero mean and unit variance. After data processing, the entire data were divided into two parts, 5300 sets of data were used to develop the models and 1736 sets of data were used as validation data to test the model generalization capabilities. From the historical operation data, it can be observed that the operation condition varied. Because the feedstock is bought from different suppliers, there exist variations in feedstock properties, which usually cause quick drops of the density.

In the soft-sensing model, the model accuracy is affected strongly by the input predictor variables. The predictor variables are determined through mechanistic analysis of the cracking process and the statistical correlation analysis of data. According to the mechanistic analysis of the practical cracking process, 49 input variables were first selected among over 100 variables. Then the correlation coefficients among all 49 variables were computed and analyzed. The variable whose coefficients with others were small was neglected, and then 34 input variables were left. Furthermore, the contributions to the output of these 34 input variables are also measured by partial least squares (PLS) methods. After neglecting those variables with small contributions to the product qualities, only 26 input variables were left, including 4 feed stock flow rates, 4 diluted stream flow rates, 4 temperatures at the outlet of convection box, 4 temperatures at the inlet of fire box, 4 coil outlet temperature (COT), 3 temperatures in the fire box, gas temperature, the density of feed stocks, and the pressure difference between inlet and outlet of coil. Then a two-layer hierarchical network model with these 26 input variables was developed.

To investigate the performance of the proposed two-layer hierarchical network model, three kinds of models were studied: Model 1 --- a single feedforward neural network trained by the Leverberg-Marquart algorithm; Model 2 --- multi-model soft sensors based on only one BANN for each product quality; Model 3 --- two-layer

hierarchical model based on classification of operation condition and BANN in each operation condition.

When training BANN, the appropriate network topology was determined by examining their relative root-mean-square-error (RRMSE). For the BANN in Model 2, there are 10 individual networks for CH_4/C_3H_6 and 11 networks for C_2H_4/C_3H_6. According to the actual operation data, the operation conditions had been divided into two kinds of conditions. Then in the Model 3, two BANN models were built based on two different operation areas. In the first operation condition, 4 component networks were used for CH_4/C_3H_6 and 5 networks for C_2H_4/C_3H_6. And in the second operation condition, 7 networks were used for CH_4/C_3H_6 and 9 networks for C_2H_4/C_3H_6, respectively. After three models were trained, the performances of their prediction errors on the validation data were compared, as shown in Table 1. It can be observed that the performance of prediction error of the hierarchical neural network in Model 3 improves more slightly than that in Model 2.

Fig. 3 and Fig. 4 show the comparison of predictions of two product qualities (CH_4/C_3H_6 and C_2H_4/C_3H_6) in Model 3 with the actual values on the validation data, where "o" stands for the predictions and "+" stands for the actual values. From the results, it can be seen that the predicted values are quite close to the corresponding true values. It should be also noticed that in Fig. 3 and Fig. 4, there are two different operation areas and the model still can predict the product yields very well.

To validate the accuracy of the trained hierarchical model, it was also applied to the actual process for about two weeks. Error distributions of the model predictions

Table 1. Comparison of prediction errors of three models

RRMSE (%)	Model 1	Model 2	Model 3
CH_4/C_3H_6	0.0136	0.0117	0.0106
C_2H_4/C_3H_6	0.0084	0.0082	0.0078

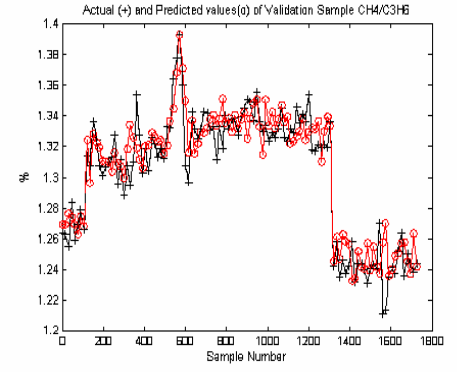


Fig. 3. Model predictions and actual values of C_2H_4/C_3H_6 on the validation data

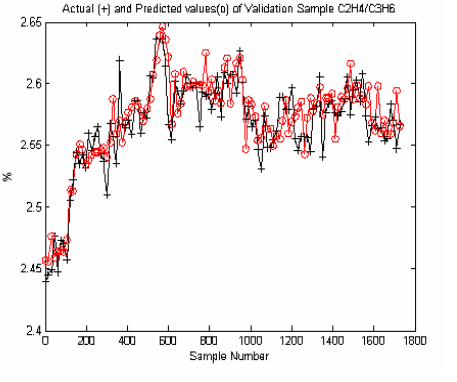


Fig. 4. Model predictions and actual values of CH_4/C_3H_6 on the validation data

were calculated, and the average relative error and maximum relative error were also used to test the performance of the model, as shown in Table 2. It can be seen that for both product qualities, the rates of model error larger than 5% are only about 2%. Such performance is well enough to satisfy the requirement of product qualities.

Table 2. Validation of the hierarchical neural network model on the actual process

Variable	Average Relative Error	Maximum Relative Error	Error Distribution (%)		
			<1%	1~5%	>5%
CH_4/C_3H_6	0.017	0.148	76.0	22.4	1.6
C_2H_4/C_3H_6	0.015	0.141	76.2	21.7	2.1

5 Conclusions

This paper presents a two-layer hierarchical neural network model to predict the product qualities of an industrial ethylene pyrolysis process. The first layer classifies the process operations into different operating conditions. In the second layer, each operating condition is modeled based on bootstrap aggregated neural networks with sequential training method. Such a hierarchical network model has good capability of generalization. Results of application to industrial data and comparison with other models show that the proposed approach is very effective.

Acknowledgements

The authors greatly appreciate the financial support of the NSFC (No. 60404012), SRF for ROCS, SEM of China, China 863 Programme through the Grant 2001 AA413320 and the UK EPSRC through the grant GR/R10875.

References

1. Zhou, Q., Xu, Y.M.: Predict the Product Yields Profile of Ethylene Pyrolysis Furnace by Multi-Models. in Proceedings of the IFAC International Conference on Intelligent Control and Signal Processing, Faro, Portugal, April 8-11, 2003
2. Sjoberg, J., Zhang, Q., Ljung, L., Benveniste, A. et al.: Nonlinear Black-box Modeling in System Identification: a Unified Overview. Automatica **31** (1995) 1691-1724
3. Bhat, N., McAvoy, T.: Use of Neural Nets for Dynamic Modeling and Control of Chemical Process Systems. Comput. Chem. Eng. **14** (1990) 573-583
4. Sridhar, D.V., Seagrave, R.C., Bartlett, E. B.: Process Modeling Using Stacked Neural Networks. AICHE Journal **42** (1996) 2529-2539
5. Xiong, Z.H., Zhang, J.: Optimal Control of Fed-Batch Processes Based on Multiple Neural Networks, Applied Intelligence **22** (2005) 149-161
6. Zhang, J.: Sequential Training of Bootstrap Aggregated Neural Networks for Nonlinear System Modeling. in Proceedings of the American Control Conference, Anchorage, Alaska, U.S.A 2002, 531-536
7. Jordan, M.I., Jacobs, R.A.: Hierarchical Mixture of Experts and the EM Algorithm. Neural Computation **6** (1994) 181-214

A Sub-stage Moving Window GRNN Quality Prediction Method for Injection Molding Processes

Xiao-Ping Guo[1,2], Fu-Li Wang[1], and Ming-Xing Jia[1]

[1] Information Science and Engineering School, Northeastern University, Shenyang, China
flwang@mail.neu.edu.cn, mingxing_jia@yahoo.com
[2] Information Engineering School, Shenyang Institute of Chemical Technology, Shenyang, China
gxping011007@tom.com

Abstract. For injection molding process, a typical multistage batch process, the final product qualities are usually available at the end of the batch, which make it difficult for on-line quality control. A sub-stage moving window generalized regression neural network (GRNN) is proposed for dedicating to reveal the nonlinearly and dynamic relationship between process variables and final qualities at different stages. Firstly, using an clustering arithmetic, PCA P-loading matrices of time-slice matrices is clustered and the batch process is divided into several operation stages, the most relevant stage to the quality variable is defined, and then applying moving windows to un-fold stage data according to time, and sub-stage GRNN models are developed for every windows for on-line quality prediction. For comparison purposes a sub-MPLS quality model of every moving window was establish. The results prove the effectiveness of the proposed quality prediction method is supervior to sub-MPLS quality prediction method.

1 Introduction

Injection molding processes, a typical multistage batch process, is an important polymer processing technique and transforms polymer materials into various shapes and types of products. However, due to the process high dimensionality, complexity, batch-to-batch variation, and also limited product-to-market time, the final product quality are usually available at the end of the batch, which is analyzed (mostly offline) after the batch completion although if feasible and economical, the product quality measurements are also done at finite time intervals as the batch run progresses make, it is difficult for on-line quality control.

Several statistical modeling methods such as multi-way partial least square (MPLS) models, have been reported recently for batch processes[1], Nevertheless, MPLS method is inefficient in revealing time-specific relationships for some multi stage processes and is linear methods. consequently, these methods perform poorly in predicting response variables of nonlinearly behaving batch processes, which are abundant in the chemical/biochemical industry.

However, an artificial neural network (ANN) had the capability to handle the modeling problems associated with nonlinear static or dynamic behaviors. The

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1138 – 1143, 2006.

NNPLS method [2]differs from the direct ANN approach in that, the input–output data are not directly used to train the FFNN, but are preprocessed by the PLS outer transform. Jiabo Zhu et al (1998) [3] proposed a time-delay neural network (TDNN) modeling method for predicting the treatment result. GRNNs are based feedforward networks which were introduced [4] as a generalization of both the radial basis function networks (RBFNs) and probabilistic neural networks (PNNs). With increasing number of training samples, the GRNN asymptotically converges to the optimal regression surface. In addition to having a sound statistical basis, the GRNNs possess a special property in that the networks do not require iterative training.

For multi-stage batch processes,each stage has its own underlying characteristics, and a batch process can exhibit significantly different behaviors over different stages. It is therefore natural to develop stage-based statistical modeling methods to reflect the inherent stage nature to improve the performances of quality control. A stage-based sub-PCA modeling method[5] has been developed, and it has been shown that the stage PCA modeling can overcome many difficulties of MPCA-based monitoring for batch processes.

Thus, in the present paper, for injection molding process, a sub-stage generalized regression neural network (GRNN) is proposed for dedicating to reveal the nonlinearly and dynamic relationship between process variables and final qualities at different stages, and to build a stage-based moving windows on-line quality prediction model. Firstly, using an clustering arithmetic,PCA P-loading matrices of time-slice matrices is clustered and the batch process is divided into several operation stages according to the change of process correlation, the most relevant stage to the quality variable is defined, and then applying moving windows to un-fold stage data according to time, and sub-stage GRNN models are developed for every windows for on-line quality prediction. For comparison purposes a sub-MPLS quality model of every moving window was established. The results prove the effectiveness of the proposed quality prediction method is supervior to sub- MPLS quality prediction method.

2 Process Description

Injection molding [6], an important polymer processing technique, transforms polymer materials into various shapes and types of products. As a typical multistage process, injection molding operates in stages, among which, filling, packing-holding, and cooling are the most important phases. During filling, the screw moves forward and pushes melt into the mold cavity. Once the mold is completely filled, the process then switches to the packing-holding stage, during which additional polymer is “packed” at a high pressure to compensate for the material shrinkage associated with the material cooling and solidification. The packing-holding continues until the gate freezes off, which isolates the material in the mold from that in the injection unit. The process enters the cooling stage; the part in the mold continues to solidify until it is rigid enough to be ejected from the mold without damage. Concurrently with the early cooling phase, plastication takes place in the barrel where polymer is melted and conveyed to the front of barrel by screw rotation, preparing for next cycle.

For injection molding, high degree of automation is possible. After the process conditions are properly set, the process repeats itself to produce molded part at a high rate. The process is, however, susceptible to the production of off-spec products due

to various process malfunctions, drifting of process conditions, changes in materials, and unknown disturbances. Abrupt, gross faults in the key process variables can be easily and reliably detected by the conventional SPC chart. Slow drift or faults involving multiple process variables, however, can not be detected. These process faults, even if they are small and not common, can lead to production of large quantity of bad parts, if they are not detected earlier.

The material used in this work is high-density polyethylene (HDPE). Ten process variables are selected for modeling, that is, Nozzle Pressure, Stroke, Injection Velocity, Hydraulic Pressure, Plastication Pressure, Cavity Pressure, Screw Rotation Speed, SV1 valve opening, SV2 valve opening, and Mold Temperature, respectively. The sampling rates of these process variables are 20 ms. The operating conditions are set as follows: injection velocity is 25mm/sec; mold temperature equals 25ºC; seven-band barrel temperatures are set to be (200, 200, 200, 200, 200, 180, 160, 120) ºC; packing-holding time is fixed to be 3 second. Quality variables are part weight. Totally, 60 batch runs are conducted under 19 different operation conditions, which can cover all the normal operation range. Based on these data, an stage-based sub-MPLS model and GRNN model is developed for process analysis and quality prediction.

3 A Stage-Based Moving Window GRNN Modeling

3.1 Generalized Regression Neural Networks (GRNNs) [4]

Consider a J-dimensional vector, $x=\left[x_1, x_2, ..., x_j\right]^T$,describing process input variables and the corresponding scalar output, y, representing the quality (output) variable.GRNN performs the regression by computing the conditional expectation of y given x. Specifically, the GRNN estimates the joint probability density function (PDF) of x and y, i.e. f(x, y), to create a probabilistic model for predicting y. The PDF estimator model is constructed from the training input–output data set $\left\{x_i, y_i\right\}; i=1,2,...,I$, via nonparametric density estimation (NDE). Given x and assuming that the function being approximated is continuous and smooth, the expected value of y, $E\left[y|x\right]$ can be

$$E\left[y|x\right]=\int_{-\infty}^{\infty} yf(x,y)dy \Big/ \int_{-\infty}^{\infty} f(x,y)dy \tag{1}$$

estimated as Using the training set and assuming Gaussian PDF, the function $f(x,y)$ can be defined as

$$f(x,y)=\frac{1}{(2\pi)^{(J+1)/2}\sigma^{J+1}} \times\frac{1}{I}\sum_{i=1}^{I}\left[\exp\left(\frac{-(x-x_i)^T(x-x_i)}{2\sigma^2}\right)\right]\times\exp\left(\frac{-(y-y_i)^2}{2\sigma^2}\right) \tag{2}$$

Where x_i and y respectively denote the *ith* training input vector and the corresponding output, and σ denotes the width (*smoothing parameter* or *spreading factor*) of the Gaussian PDF. Given x, the corresponding regression estimate, $\hat{y}(x)$, can be determined as a conditional mean by substituting Eq. (2) in Eq. (1).

$$\hat{y}(x) = E\left[y|x\right] = \frac{\sum_{i=1}^{I} y_i h_i}{\sum_{i=1}^{I} h_i}; h_i = \exp\left[-\frac{d_i^2}{2\sigma^2}\right] \tag{3}$$

where h_i denotes the Gaussian radial basis function and σ^2 represents the squared Euclidean distance between vectors x and x_i defined as

$$d_i^2 = (x - x_i)^T (x - x_i). \tag{4}$$

3.2 Sub-GRNN Modeling

The data gathered from injection molding process forms a three-dimensional data matrix, $\mathbf{X}(I \times J \times K)$, where for batch process applications, I denotes cycle number, J denotes variable number, and K denotes the number of samples within a cycle. Wold et al. [7] proposed to solve this problem by rearranging the data matrix into two-dimensional form. The $\mathbf{X}(I \times J \times K)$ is unfolded, with each of the K time slabs concatenated to produce a two-way array, $\mathbf{X}_{new}(I \times JK)$. The multi-way unfolding procedure is graphically shown in Fig. 1.

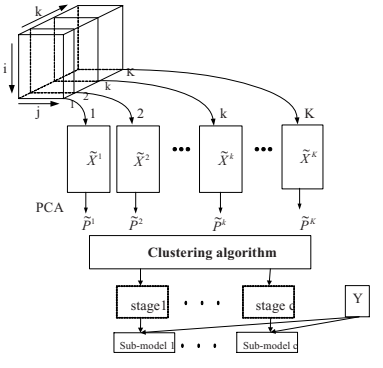


Fig. 1. Sub-stage modeling method

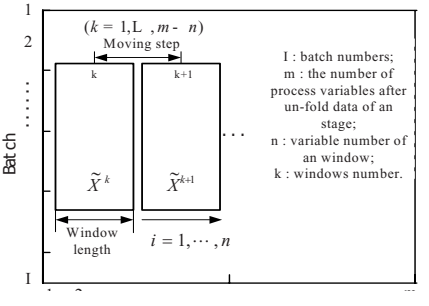


Fig. 2. Illustration moving windows of an stage

Based on the work of sub-PCA stage partition strategy[5]，an batch run is divided into several stages. A moving window of all un-fold data of every stage is proposed to extract the local covariance information of process variable and quality variable. All un-fold data of one stage represented as $X(I \times m)$, where I is batch numbers, and m is the number of process variables after un-fold data of the stage. Moving step can be

set n which can be set as small as 1. With this arrangement, $(m-n)$ number of windows can be resulted for each stage, designated as, $\tilde{X}^k(k=1,\cdots,m-n)$ shown in Fig. 2. Using every moving windows data and quality data establish a sub-GRNN quality model. For comparison purposes a sub-MPLS quality model of every moving window was established.

4 Experimental Results

Without using any prior process knowledge, using sub-PCA based stage-division algorithm, the trajectories of an injection molding batch run is divided into four stages according to the change of local covariance structure, correspond to the four physical operation stages, i.e., injection, packing-holding, plastication and cooling stages shown in Fig. 3.

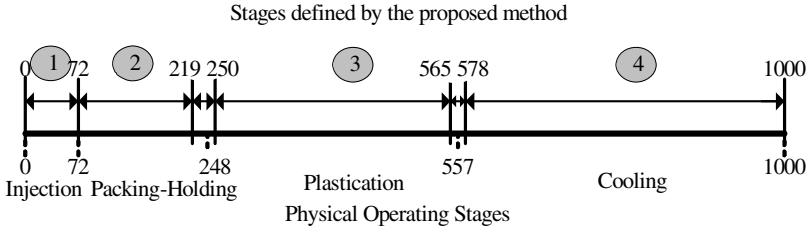


Fig. 3. Stage division results for injection molding process

The final quality variables have weak relation with the plastication and cooling stage. The on-line quality prediction model is a distributed model, the weight variables are estimated by the sub-MPLS models and sub-GRNN models in packing stage.

For illustration, the results of the GRNN and PLS-based output prediction for 30 batches using the 100 sample time of packing stage are shown in Fig. 4. In Fig. 4，the solid line with circle symbols indicates the weight measurements, and the solid line with square symbols plots the corresponding weight prediction using sub-MPLS model and the solid line with triangle symbols plots the corresponding weight prediction using sub-GRNN model. It is clear that the product weight predicted can be more roughly predicted. But there exists an obvious offset between the measured and predicted values for these methods, and the value of the offset varies with different operating conditions, for instance, the offset for batch 5 is much smaller than that of batch 20 for sub-MPLS. An analysis suggests that the problem was caused by forepart data of packing stages. Based on sub-MPLS and sub-GRNN weight prediction model of the packing stage, using the200 sample time of packing stage 30 batch weight prediction result are shown in Fig. 5. Compared to Fig. 4, the predictions of two methods are much closer to the actual weight measurements, indicating significant improvement by using tail data of the packing stage prediction model. But the product weight predicted by sub-GRNN model can be more exactly predicted.

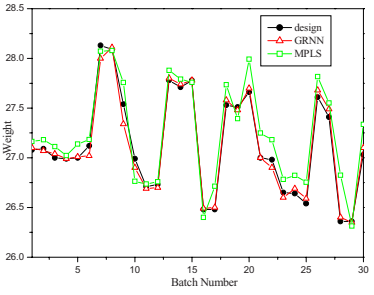


Fig. 4. Predicted results for weight variables using the 100th sample time data

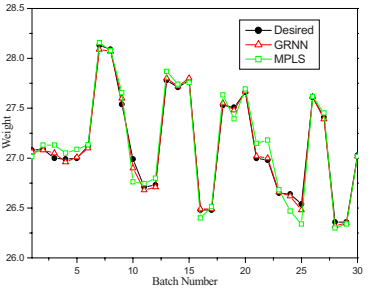


Fig. 5. Predicted results for weight variables using the 200th sample time data

5 Conclusion

For injection molding process, an typical multi-stage batch process, a new quality prediction method has been applied. Firstly, Process stages are determined by analyzing the change of process covariance structure and partitioning time-slice PCA loading matrices using a new clustering algorithm. Then applying moving windows to un-fold stage data, for each windows an sub-GRNN model is developed In addition, the predicted result of sub-GRNN model is compared with the one of the sub-MPLS model, the predicted precision of sub-GRNN model is superior to the one of the sub-MPLS model.

Acknowledgement

This work was supported by the National Science Foundation of China under Grant 60374003 and project 973 under Grant 2002CB312200.

References

1. Nomikos P. and MacGregor J.F. Multiway Partial Least Squares in Monitoring Batch Processes. Chemometrics Intell. Lab. Syst. 30 (1995), 97-108.
2. Qin, S.J., McAvoy, T.J.: Nonlinear PLS Modeling Using NN. Comput. Chem. Eng. 16 (1992) 379-391
3. Zhu, J., Zurcher, J., Rao, M.: Meng, M.Q.H.: An Online Wastewater Quality Predication System Based on a Time-delay Neural Network. Engineering Applications of Artificial Intelligence 11 (1998) 747-758
4. D.F. Specht, A General Regression Neural Network, IEEE Trans. Neural Net. 2 (1991) 568–576.
5. Lu N., Gao F., Yang Y., and Wang F.: A PCA-based Modeling and On-line Monitoring Strategy for Uneven-length Batch Processes. Ind. Eng. Chem. Res. 43 (2004) 3343-3352.
6. Yang, Y.: Injection Molding: from Process to Quality Control. Ph.D. Thesis, Hong Kong University of Science & Technology (2004)
7. Wold, S., Antti, H., Lindgren, F.: Orthogonal Signal Correction of Near-Infrared Spectra. Chemom. Intell. Lab.Syst. 44 (1998) 175-184

Joint Time-Frequency and Kernel Principal Component Based SOM for Machine Maintenance

Qianjin Guo[1,2], Haibin Yu[1], Yiyong Nie[1], and Aidong Xu[1]

[1] Shenyang Inst. of Automation, Chinese Academy of Sciences, Liaoning 110016, China
guoqianjin@sia.cn
[2] Graduate School of the Chinese Academy of Sciences, Beijing 100039, China

Abstract. Conventional vibration signals processing techniques are most suitable for stationary processes. However, most mechanical faults in machinery reveal themselves through transient events in vibration signals. That is, the vibration generated by industrial machines always contains nonlinear and non-stationary signals. It is expected that a desired time-frequency analysis method should have good computation efficiency, and have good resolution in both time domain and frequency domain. In this paper, the auto-regressive model based pseudo-Wigner-Ville distribution for an integrated time-frequency signature extraction of the machine vibration is designed, the method offers the advantage of good localization of the vibration signal energy in the time-frequency domain. Kernel principal component analysis (KPCA) is used for the redundancy reduction and feature extraction in the time-frequency domain, and the self-organizing map (SOM) was employed to identify the faults of the rotating machinery. Experimental results show that the proposed method is very effective.

1 Introduction

Vibration analysis is a basic approach for fault diagnosis of rotating machinery. When a fault begins to develop, the vibration profile will change its characteristic shape. By using an appropriate signal processing method, it is feasible to detect changes in vibration signals caused by faulty components and to judge the conditions of the machinery. Traditional analysis has generally relied upon spectrum analysis based on FFT or STFT [1]. Unfortunately, FFT-based methods are not suitable for non-stationary signal analysis and are not able to reveal the inherent information of non-stationary signals. However, various kinds of factors, such as the change of the environment and the faults from the machine itself, often make the vibration signal of the running machine contain non-stationary components. Usually, these non-stationary components have abundant fault information, so it is important to analyze the non-stationary signal [2]. Hitherto, time-frequency analysis is the most popular method for the analysis of non-stationary vibration signals, such as the Gabor transform [3], and the bilinear time-frequency representation [4,5]. In this study, the auto-regressive model based pseudo-Wigner-Ville distribution is used for an integrated time-frequency signature extraction of the machine vibration. And then kernel

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1144 – 1154, 2006.

Principal component analysis (KPCA) is used for feature extraction, which transforms a high-dimensional input vector into a low-dimensional one whose components are uncorrelated.

PCA performs poorly due to its assumption that the process data are linear [6,7]. To handle the problem, a new nonlinear PCA technique for tackling the nonlinear problem, called kernel PCA, has been in development in recent years [8-10]. KPCA can efficiently compute Principal components (PCs) in high-dimensional feature spaces by means of integral operators and nonlinear kernel functions.

Based on the features extracted from the time-frequency moments using KPCA method, the machine fault diagnoses are to be classified through the SOM network. The self-organizing map (SOM), proposed by Kohonen [11,12], has been widely used in many industrial applications such as pattern recognition, biological modeling, data compression, signal processing, and data mining [12-16]. It is an unsupervised and nonparametric neural network approach. The success of the SOM algorithm lies in its simplicity that makes it easy to understand, simulate and be used in many applications.

2 Time-Frequency Analysis

The analysis of a signal is usually performed in the time or frequency domain. The energy $|x(t)|^2$ of the signal *x(t)* per unit time at the particular time *t*, can be obtained when time domain analysis is used. At the same time the energy density of the signal per unit frequency at the particular frequency *f* is given by $|x(f)|^2$ in the frequency domain, where $x(f)$ is the Fourier transform of $x(t)$. When the spectral content of the signal is changing in time, neither the time nor the frequency domain is sufficient to describe the signal properties accurately. The analysis of a signal using the joint time-frequency approach is called time-frequency analysis. There are many different time-frequency distributions with various properties [4,5]. The WVD used in this paper was the most widely used in practice. For the sake of completeness, different forms of the WVD distribution are defined below [4,5,17-21].

2.1 Wigner-Ville Distribution

The Wigner distribution (WD) can be derived by generalizing the relationship between the power spectrum and the autocorrelation function for non-stationary, time-variant processes. The physical interpretation of the generalized power spectrum *F(t, f)* is that it represents the instantaneous power density spectrum. This approach leads to the WD defined as [19]

$$W_x(t,f) = \int_{-\infty}^{+\infty} x^*(t-\frac{\tau}{2})x(t+\frac{\tau}{2})e^{-i2\pi f\tau}d\tau \tag{1}$$

The WD can be represented in terms of the spectrum by

$$W_X(f,t) = \int_{-\infty}^{+\infty} X^*(f-\frac{\theta}{2})X(f+\frac{\theta}{2})e^{-i2\pi\theta t}d\theta \tag{2}$$

The Wigner distribution (WD) of the analytic signal is called the Wigner-Ville distribution (WVD). Here the analytic signal is defined as

$$x_A = x(t) + j\hat{x}(t) \tag{3}$$

where $\hat{x}(t)$ is the Hilbert transform of $x(t)$

$$H[x(t)] = \hat{x}(t) = \frac{1}{\pi}\int_{-\infty}^{+\infty} x(\tau)\frac{1}{t-\tau}d\tau \tag{4}$$

It is well known that the spectrum of the analytic signal does not have any negative frequencies and can be obtained as

$$X_A(f) = X(f) + jX(f) = X(f) + \mathrm{sgn}\, fX(f) = \begin{cases} 0, & \text{if } f<0 \\ X(f), & \text{if } f=0 \\ 2X(f), & \text{if } f>0 \end{cases} \tag{5}$$

Thus, the WVD is very often used in practical applications since it avoids interference between positive and negative frequencies [19-20].

2.2 Smoothed and Weighted Wigner-Ville Distribution

One of the main difficulties with the WVD is its bilinear character that leads to interference in the time- frequency domain. This interference can be removed by a smoothing procedure that can be defined as [4,5]

$$W_s(t,f) = \int L(t-t', f-f')W(t', f')dt'df' \tag{6}$$

where $L(t, f)$ is a smoothing function. The optimal smoothing function is a Gaussian given by [4,5]

$$L(t,f) = \frac{1}{\alpha\beta}e^{-\beta/\alpha-(2\pi f)^2\beta} \tag{7}$$

The smoothed WVD is often called the pseudo-WVD. It has been shown that any postprocessing procedure applied to the distribution gives a poor energy concentration resulting in the loss of time and frequency resolution [19]. Additionally the possibility of detection of weak non-stationary events may be significantly reduced when smoothing is applied [19]. An alternative to smoothing in the time and frequency domain is a modified form of the WVD called the weighted WVD defined as [21].

3 Kernel Principal Component Analysis (KPCA)

According to Cover's theorem, the nonlinear data structure in the input space is more likely to be linear after high-dimensional nonlinear mapping [22]. This higher-dimensional linear space is referred to as the feature space (F). KPCA finds a computationally tractable solution through a simple kernel function that intrinsically constructs a nonlinear mapping from the input space to the feature space. As a result, KPCA performs a nonlinear PCA in the input space [23]. If a PCA is aimed at decoupling nonlinear correlations among a given set of data (with zero mean), $x_k \in R_m$, $k = 1,2,\ldots, M$ through diagonalizing their covariance matrix, the covariance can be expressed in a linear feature space F instead of the nonlinear input space, i.e.,

$$\bar{C} = \frac{1}{M}\sum_{j=1}^{M} \phi(x_j)\phi(x_j)^T \tag{8}$$

where it is assumed that $\sum_{k=1}^{M}\phi(x_k) = 0$, and $\phi(\cdot)$ is a nonlinear mapping function that projects the input vectors from the input space to F. Note that the dimensionality of the feature space can be arbitrarily large or possibly infinite [8]. To diagonalize the covariance matrix, one has to solve the eigenvalue problem in the feature space. We now have to find eigenvalues $\lambda > 0$ and eigenvectors $V \in F \setminus \{0\}$ satisfying

$$\lambda V = \bar{C}V \tag{9}$$

Here, $\bar{C}V$ can be expressed as follows:

$$\bar{C}V = \left(\frac{1}{M}\sum_{j=1}^{M} \phi(x_j)\phi(x_j)^T\right)V = \frac{1}{M}\sum_{j=1}^{M} \langle \phi(x_j), V \rangle \phi(x_j) \tag{10}$$

where $<x, y>$ denotes the dot product between x and y. This implies that all solutions V with $\lambda \neq 0$ must lie in the span of $\phi(x_1),\ldots, \phi(x_M)$. Hence Eq.(9) is equivalent to

$$\lambda \langle \phi(x_k), V \rangle = \langle \phi(x), \bar{C}V \rangle, \quad k = 1, 2, \ldots, M \tag{11}$$

and there exist coefficients $\alpha_i (i = 1, 2, \ldots, M)$ such that

$$V = \sum_{i=1}^{M} \alpha_i \phi(x_i) \tag{12}$$

Combining Eqs. (11) and (12), we obtain

$$\lambda\sum_{i=1}^{M}\alpha_i \langle \phi(x_k), \phi(x_i) \rangle = \frac{1}{M}\sum_{i=1}^{M}\alpha_i \left\langle \phi(x_k), \sum_{j=1}^{N}\phi(x_j) \right\rangle \left\langle \phi(x_j), \phi(x_i) \right\rangle = \frac{1}{M}\sum_{i=1}^{M}\alpha_i \left(\phi(x_k) \cdot \sum_{j=1}^{N}\phi(x_j) \right) \left(\phi(x_j) \cdot \phi(x_i) \right) \tag{13}$$

for all $k = 1,2,\ldots,N$. Note that the eigenvalue problem in Eq.(13) only involves dot products of mapped shape vectors in the feature space.

Now, let us define an $M \times M$ matrix K by $[K]_{ij} = K_{ij} = \langle \phi(x_i), \phi(x_j) \rangle$. Then the left-hand side of Eq.(13) can be expressed as

$$\lambda \sum_{i=1}^{M} \alpha_i \langle \phi(x_k), \phi(x_i) \rangle = \lambda \sum_{i=1}^{M} \alpha K_{ki} \tag{14}$$

Since $k = 1,2,\ldots,M$, Eq. (14) becomes $\lambda K\alpha$. The right-hand side of Eq.(13) can be expressed as

$$\frac{1}{M}\sum_{i=1}^{M} \alpha_i \left\langle \phi(x_k), \sum_{j=1}^{M} \phi(x_j) \right\rangle \left\langle \phi(x_j), \phi(x_i) \right\rangle = \frac{1}{M}\sum_{i=1}^{M} \alpha_i \sum_{j=1}^{M} K_{kj} K_{ji} \tag{15}$$

Since $k = 1,2,..M$, Eq.(15) becomes $(1/N)K^2\alpha$. Combining Eqs.(13) and (14), we obtain

$$\lambda MK\alpha = K^2\alpha \tag{16}$$

where α denotes the column vector with entries $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_M]^T$. To find solutions of Eq.(16), we solve the eigenvalue problem

$$M\lambda\alpha = K\alpha \tag{17}$$

for nonzero eigenvalues. A justification of this procedure is given in [8]. Now, performing PCA in F is equivalent to resolving the eigen-problem of Eq.(17). This yields eigenvectors $\alpha_1,\alpha_2,...,\alpha_M$ with eigenvalues $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_M$. We normalize $\alpha_1,\alpha_2,...,\alpha_p$ by requiring that the corresponding vectors in F be normalized, i.e.,

$$\langle v_k , v_k \rangle = 1 \qquad for\ all\ k = 1,2,...,p \tag{18}$$

Using $v_k = \sum_{i=1}^{M} \alpha_i^k \phi(x_i)$, Eq. (18) leads to

$$1 = \left\langle \sum_{i=1}^{M} \alpha_i^k \phi(x_i), \sum_{j=1}^{M} \alpha_j^k \phi(x_j) \right\rangle = \sum_{i=1}^{M} \sum_{j=1}^{M} \alpha_i^k \alpha_j^k \langle \phi(x_i), \phi(x_j) \rangle = \sum_{i=1}^{M} \sum_{j=1}^{M} \alpha_i^k \alpha_j^k K_{ij} = \langle \alpha_k , K\alpha_k \rangle = \lambda \langle \alpha_k , \alpha_k \rangle \tag{19}$$

The principal components t of a test vector x are then extracted by projecting $\phi(x)$ onto eigenvectors v_k in F, where $k = 1,2,\ldots,p$.

$$t_k = \langle v_k , \phi(x) \rangle = \sum_{i=1}^{M} \alpha_i^k \langle \phi(x_i), \phi(x) \rangle \tag{20}$$

To solve the eigenvalue problem of Eq.(17) and to project from the input space into the KPCA space using Eq. (20), one can avoid performing the nonlinear mappings and computing both the dot products in the feature space by introducing a kernel function of form $k(x, y) = \langle \phi(x), \phi(y) \rangle$ [8][23].

There exist a number of kernel functions. The requirement on the kernel function is that it satisfies Mercer's theorem [24]. Before applying KPCA, mean centering in the high dimensional space should be performed [24].

4 Self-Organizing Maps

Compared with the multilayer artificial neural network (ANN) [12], [13], the SOM is an unsupervised learning network that organizes itself according to the natural structure of the data. It projects input space onto a number of prototype vectors that symbolize properties of the data [13]. The structure of SOM and the batch-training algorithm [12] proposed for the SOM are described below.

Suppose the SOM consists of U units with each unit i, $i=1,2...,U$, having d-dimensional prototype weight vector, $w_i = [w_{i1},...,w_{id}]$ (see Fig.1). In this paper, the dimension of d is equal to the number of features extracted from KPCA. The U units are connected to adjacent ones by neighborhood relation [13].

The initial prototype weight vectors ω_i of the network are given randomly, based on which computed are the distances between an input feature vector y and all of the prototype vectors. By using the criterion of minimum Euclidean distance, the winning unit c (the winner) is found out of the i units, $i=1,2,...,U$

$$\|x - w_c\| = \min_i \{\|x - w_i\|\} \tag{21}$$

The prototype vectors of the winner and its topological neighbors on the SOM are moved closer to the input vector of the input space. Accordingly, the prototype

vectors of the SOM are updated. The updating rule as shown in (22) adjusts the weights of the winner and its neighbors.

$$w_i(g+1) = w_i(g) + \alpha(g) h_{c,i}(g)[x - w_i(g)] \tag{22}$$

where g is the iterating number, $\alpha(g)$ denotes the learning rate belonging to [0,1], and $h_{c,i}(g)$ is a neighborhood function with the Gaussian form

$$h_{c,i}(g) = e^{-\frac{\|r_c - r_i\|^2}{2\sigma^2(g)}} \tag{23}$$

where r_c and r_i are positioning numbers of units c and i on the SOM grid and $\sigma(g)$ is the neighborhood radius decreasing with the iterating number g.

The efficiency of the mapping configuration depends on the neighborhood function, the learning rate, and the initial weight vectors. To more efficiently train the SOM, the batch-training algorithm [13] described is employed.

5 Experiments for Machine Maintenance

The fault diagnosis system is shown in Fig.1. The system mainly consists of four sections: data acquisition, signal processing, feature extraction and fault diagnosis. The raw time signal is obtained by the accelerometer from the machinery fault simulator. Then the features of the data are extracted through the pseudo-Wigner-Ville distribution and KPCA algorithms. Feature extraction algorithms make data quantity from the view of statistics. Finally, the SOM is trained and used to classify the faults of machinery.

Experiments were performed on a machinery fault simulator, which can simulate the most common faults, such as misalignment, unbalance, resonance, radial rubbing, oil whirling and so on. The schematic of the test apparatus mainly consists of a motor, a coupling, bearings, discs and a shaft etc. In this system, the rotor is driven by an electromotor, and the bearing is the journal bearing. The measurements with acceleration, velocity, or displacement data from rotating equipment are acquired by the NI digital signal acquisition module, and then are collected into an embedded controller.

5.1 Signal Processing

5.1.1 Unbalance

For rotating machines, unbalance is a kind of serious malfunction, which often presents a serious hazard to machines. The time-frequency method is used to analyze the experimental data of unbalance in this section. The unbalance is achieved by putting a known extra mass on a well-balanced disk manually.

Fig.1 shows a set of horizontal vibration data, which were sampled at a speed of 1.6 kHz. The rotary speed is 3000 r.p.m. The left side of Fig.1 is the spectrum of the unbalance signal, through which it can be seen that the unbalance fault excites the components of 1X. It is obvious that the energy of signal mainly centralizes at 1X component. Compared with the spectrum, the Wigner-Ville distribution (right side of Fig.1) shows the components clearly, and the approximate time for the appearance of components is also shown. The comparison result indicates that the Wigner-Ville method

can reflect the inherent features of the unbalance signal and can provide information for unbalance machines analysis and more useful features for fault diagnostics.

5.1.2 Coupling Misalignment

Coupling misalignment, one of the most familiar faults, often denotes the slant or misalignment between the axes of two nearly rotors. When misalignment exists, a series of dynamic responses undesired will occur in the rotor system, such as coupling defection, bearing abrasion and oil collapsing, etc. So it is very important to find misalignment as early as possible for ensuring the safe running of the machines.

The main frequency feature of the coupling misalignment is the increase of 2X component. Through the adjustment of the coupling, coupling misalignment can be manually simulated, as shown in Fig.2. The sampling speed is 1.6 Hz, and the rotating speed is 3000 r.p.m. The spectrum of coupling misalignment signals is given in Fig.2. 2X component, which is the spectrum feature of the coupling misalignment, can be seen on the figures. But in the order of distinctness, the Wigner-Ville is the better.

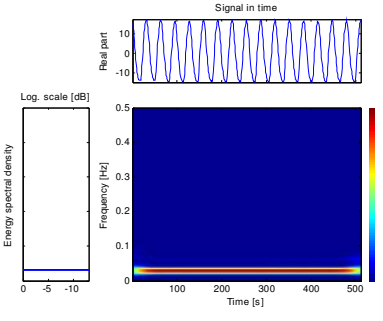


Fig. 1. Spectrum and weighted Wigner-Ville distribution of unbalance

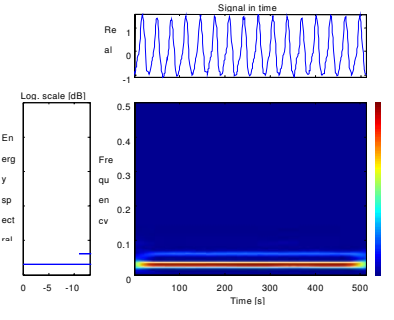


Fig. 2. Spectrum and weighted Wigner-Ville distribution of misaligment

5.1.3 Rotor to Stator Rub-impact

For rotating machines, rub-impact between rotor and stator is a kind of serious malfunction, which often happens at the positions with small clearances [25]. The factors that influence rub-impact between rotor and stator are complicated, and the vibration phenomenon of a rub-impact rotor system is also complicated. The weighted Wigner-Ville is used to analyze the experimental data of rub-impact in this section. The radial rub-impact between the rotor and stator is achieved by using a copper block to contact with rotor manually. Fig.3 shows a set of horizontal vibration data, which were sampled at a speed of 1.6 kHz. The rotary speed is 3000 r.p.m.

The left side of Fig.3 is the spectrum of the rotor-to-stator rub signal, through which it can be seen that the rub-impact fault excites the components of 1/2X, 1X, 3/2X, 2X, 3X, etc. It is obvious that the energy of signal mainly centralizes at 1X component, while other components are small and are not shown clearly in the spectrum. Compared with the spectrum, the weighted Wigner-Ville distribution (right side of Fig.3) shows all components clearly, and the approximate time for the appearance of components is also shown. Through the Wigner-Ville distribution, we can see the energy

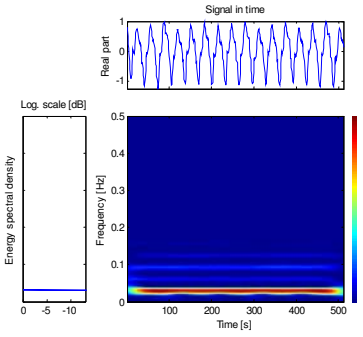


Fig. 3. Spectrum and weighted Wigner-Ville distribution of rub-impact

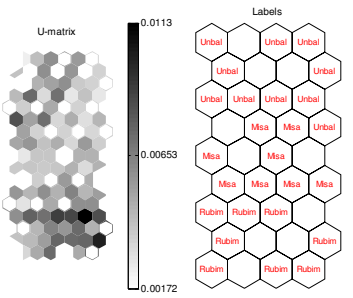


Fig. 4. The U matrix and SOM tested with labeled data set (Unbal, unbalance; Misa, misalignment; Rubmi, rub impact)

density is represented by different shades of the color. In addition, the 2X, 3X and other components can be seen clearly. The comparison result indicates that the Wigner-Ville distribution can reflect the inherent features of the rub-impact signal and can provide information for rub-impact machines analysis and more useful features for fault diagnostics.

5.2 Feature Extraction

Since time-frequency moments, described in Section 1 tend to be asymptotically Gaussian [17], one can model the machine behavior through the parameters of a multivariate Gaussian distribution function describing the distribution of the time-frequency moments collected during different stages of machine operation (training process). Usually, one can observe a high degree of correlation between the moments, and the uncorrelated portion of the information contained in the time-frequency moments can be extracted through the use of the kernel Principal Component Analysis (KPCA).

Table 1. The Principal Components of Feature Extraction

Fault class	PCs1	PCs2	PCs3	PCs4	PCs5	PCs6
Unbalance	0.0000	0.0000	0.0000	1.0000	0.0042	0.0035
Misalignment	0.0000	0.0000	0.0000	0.8000	1.0000	0.0010
Rub-impact	0.2100	0.1800	0.1652	1.0000	0.1765	0.1900

KPCA is a standard technique for data reduction and feature extraction. Feature extraction is a process of transforming a data space into a feature space that has the same dimensionality measure as the data space, but represents the data space by a reduced number of important features (See Table 1). In other words, KPCA algorithm accounts for equal information distribution among input vectors in the data set. Transformed data was then inputted to the SOM networks, which consist of 6 neurons in the input layer and 3 features in the representational layer.

5.3 Fault Classification

In training the SOM, the selected 100 examples from the time-frequency moments using KPCA method were used to train a 4×9 SOM. The reason why a medium number of neurons are used is because it makes classification easier and more flexible. When more neurons are used in the network, the regions corresponding to a certain classification become larger, and classification becomes more flexible. In this case, the neurons all corresponded to machine diagnosis. After the network is trained successfully, it can be used to simulate the judgment of an expert to identify spot faults. The analytical symptomatic data is input into the network, and then the network will give the judging result on its rules. The unified distance matrix (u-matrix)[26] is a simple and effective tool to show the possible cluster structure on SOM grid visualization. It shows the distances between neighboring units using a gray scale representation on the map grid. In the U-matrix, dark color indicates large distance between adjacent map units and light color small distance. This gives an impression of “mountains”(long distances) which divide the map into “fields”(dense parts, i.e.,clusters).See Fig.4, the 'U-matrix' shows distances between neighboring units and thus visualizes the cluster structure of the map. High values on the U-matrix mean large distance between neighboring map units, and thus indicate cluster borders. Clusters are typically uniform areas of low values. Refer to colorbar to see which colors mean high values, respectively. For each of the fault features, a 4×9 representative SOM can be shown in Fig.4, where all the fault variables are shown on a single graph. The observations in the testing data set were labeled “unbalance”, “rub-impact” or “misalignment”, corresponding to the machine fault components. The best matching units for these data were looked up on the SOM, and the program was instructed to place the labels of the test data on the corresponding best matching neurons. As shown in Fig.4, the three letters in the figure correspond to the according faults and the blanks denote the neurons that are not excited for all the input vectors. The map not only could figure out the fault modes but also indicate the similar samples. Because it indicates that the locations of two neurons in the output map will be close to each other if their fault symptoms are similar. It is an important topology-mapped characteristic of the SOM neural network. The monitoring system obtained a near 100% success rate in distinguishing among all test fault diagnosis.

6 Conclusions

In this paper, a new system to machine degradation assessment and monitoring is proposed. The pseudo-Wigner-Ville distribution has been applied to this machinery fault diagnosis system due to their advantages in the representation of signals in both the time-frequency domains. This feature of the time-frequency analysis meets the requirements for analyzing vibration signals that are non-stationary signals. Based on the features extracted from the time-frequency moments using KPCA method, the machine fault diagnoses were to be classified through the SOM network. An experimental test rig is set up to simulate machinery faults, and the performance of the developed system has been evaluated in the simulate fault samples. The test results showed that this system was effective in detecting machinery fault diagnosis.

Acknowledgement

Foundation item: Project supported by the National High-Tech. R&D Program for CIMS, China (Grant No. 2003AA414210).

References

1. Yen, G.G., Lin, K.C..: Wavelet Packet Feature Extraction for Vibration Monitoring, IEEE Transactions on Industrial Electronics 47 (2000) 650-667.
2. Pan, M.C., Sas, P.: Transient Analysis on Machinery Condition Monitoring. IEEE Proceedings of International Conference on Signal Processing 2 (1996)1723-1726.
3. Bastiaans, M.J.: Gabor'expansion of a Signal into Gaussian Elementary Signals. Proceedings of the IEEE 68(1980).538-539.
4. Cohen, L.: Time-frequency Distribution—a Review, Proc. IEEE 77 (1989) 941-981.
5. Cohen, L.: Time-Frequency Analysis, Prentice-Hall, Englewood Cliffs, NJ 1995.
6. Wise, B.M., Gallagher, N.B.: The Process Chemometrics Approach to Process Monitoring and Fault Detection. Journal of Process Control 6 (6) (1996)329-348.
7. Dong, D., McAvoy, T.J.: Nonlinear Principal Component Analysis Based on Principal Curves and Neural Networks. Computers and Chemical Engineering 20 (1) (1996) 65-78.
8. Schölkopf, B., Smola, A.J., MJuller, K.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation 10 (5) (1998)1299-1399.
9. Mika, S., Schölkopf, B., et al.: Kernel PCA and De-noising in Feature Spaces. Advances in Neural Information Processing Systems 11 (1999)536-542.
10. Romdhani, S., Gong, S., Psarrou, A.: A multi-view Nonlinear Active Shape Model Using Kernel PCA. Proceedings of BMVC, Nottingham, UK (1999) 483-492.
11. Kohonen,T.: Self-Organizing Maps, 3rd ed. Berlin, Germany: Springer.(2001)138–140.
12. Kohonen,T.: Self-organized Formation of Topologically Correct Feature Maps, Biol. Cybernetics 43 (1982) 59-69.
13. Juha ,V., Esa, A.: Clustering of the Self-organizing Map, IEEE Trans. Neural Networks 11 (2000) 586–600.
14. Kohonen,T., Kaski, S., Lagus, K., Salojärvi, J., et al.: Self-organizing of a Massive Document Collection, IEEE Trans. Neural Networks 11 (2000)574–585.
15. Chang, C.H., Xu, P.F., Xiao, R., Srikanthan, T.: New Adaptive Color Quantization Method Based on Self-Organizing Maps, IEEE Trans. Neural Networks 16(1) (2005)237-249
16. Vermasvuori, M., Enden, et al: The Use of Kohonen Self-organizing Maps in Process Monitoring, Proceedings of IEEE International Symposium Intelligent Systems 3 (2002)2-7.
17. Qian, S., Chen, D.: Joint Time-frequency Analysis, IEEE Signal Processing Magazine 16(2) (1999)52-67.
18. Qian, S.: Introduction to Time-Frequency and Wavelet Transforms, NJ, Prentice Hall (2002)
19. Classen, T. A. C. M., Mecklenbrauker, W. F. G.: The Wigner Distribution--a Tool for Time-frequency Analysis, Philips Journal of Research 35 (1980)217-250,276-300.
20. Hammond, J. K., White, P. R.: The analysis of Nonstationary Signals Using Time-Frequency Methods, Journal of Sound and Vibration 190(1996)419-447.

21. Mcfadden, P. D., Wang, W. J.: Analysis of Gear Vibration Signatures by the Weighted Wigner-Ville Distribution ,Proceedings of the Fifth International Conference on Vibrations in Rotating Machinery, University of Bath, ImechE (1992)387-393
22. Haykin, S.: Neural Networks. Prentice-Hall, Englewood Cliffs, NJ (1999).
23. Romdhani, S., Gong, S., Psarrou, A.: A Multi-view Nonlinear Active Shape Model Using Kernel PCA. Proceedings of BMVC, Nottingham,UK (1999)483-492
24. Christianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press, UK (2000).
25. Hu, N.Q., Chen, M., Wen, X.S. :The Application of Stochastic Resonance Theory for Early Detecting Rubbing Caused Impacts Fault of Rotor System, Mechanical Systems and Signals Processing 17 (2003) 883-895.
26. Ultsch, A., Siemon, H.P.:Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis. In Proceedings Intern. Neural Networks, Kluwer Academic Press, Paris (1990) 305-308

Automatic Recognition and Evaluation of Natural Language Commands

Maciej Majewski and Wojciech Kacalak

Technical University of Koszalin, Department of Mechanical Engineering,
Raclawicka 15-17, 75-620 Koszalin, Poland
{maciej.majewski, wojciech.kacalak}@tu.koszalin.pl

Abstract. New applications of artificial neural networks are capable of recognition and verification of effects and safety of commands given by the operator of the technological device. In this paper there is a review of the selected issues on estimation of results and safety of the operator's commands as well as supervision of the technological process. A view is offered of the complexity of effect analysis and safety assessment of commands given by the operator using neural networks. The first part of the paper introduces a new concept of modern supervising systems of the technological process using a natural language human-machine interface and discusses the general topics and issues. The second part is devoted to a discussion of more specific topics of the automatic command verification that have led to interesting new approaches and techniques.

1 Intelligent Two-Way Communication by Voice

The advantages of intelligent two-way voice communication between the technological devices and the operator in Fig. 1 include the following [1,3]:

- More resistance from the operator's errors and more efficient supervising of the process with the chosen level of supervision automation.
- Elimination of scarcities of the typical co-operation between the operator and the technological device.
- Reaching a higher level of organizing realization of the technological process equipped with the intelligent two-way voice communication system, which is relevant for its efficiency and production humanization.
- No need of an operator being present at the work stand by the technological device (any distance from the technological device) [7].

The intelligent two-way voice communication layer in Fig. 2 is equipped with the following intelligent mechanisms: operator identification, recognition of words and complex commands, command syntax analysis, command result analysis, command safety assessment, technological process supervision, and also operator reaction assessment [2].

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1155–1160, 2006.

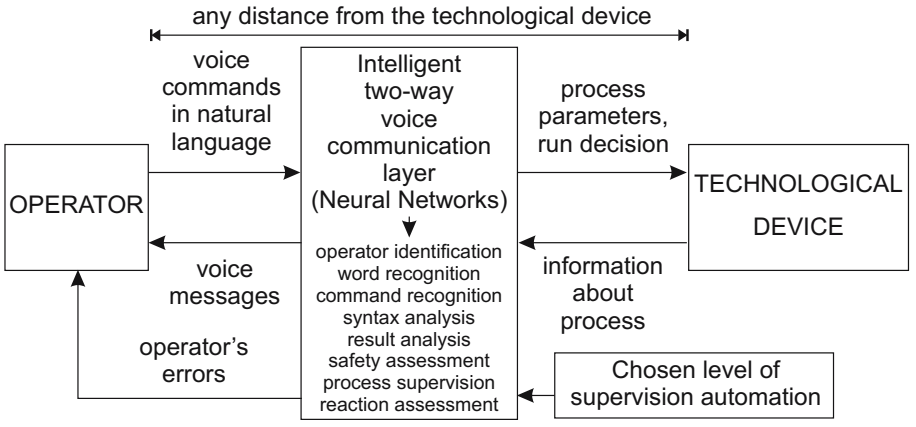


Fig. 1. General scheme of intelligent two-way voice communication between the technological device and the operator

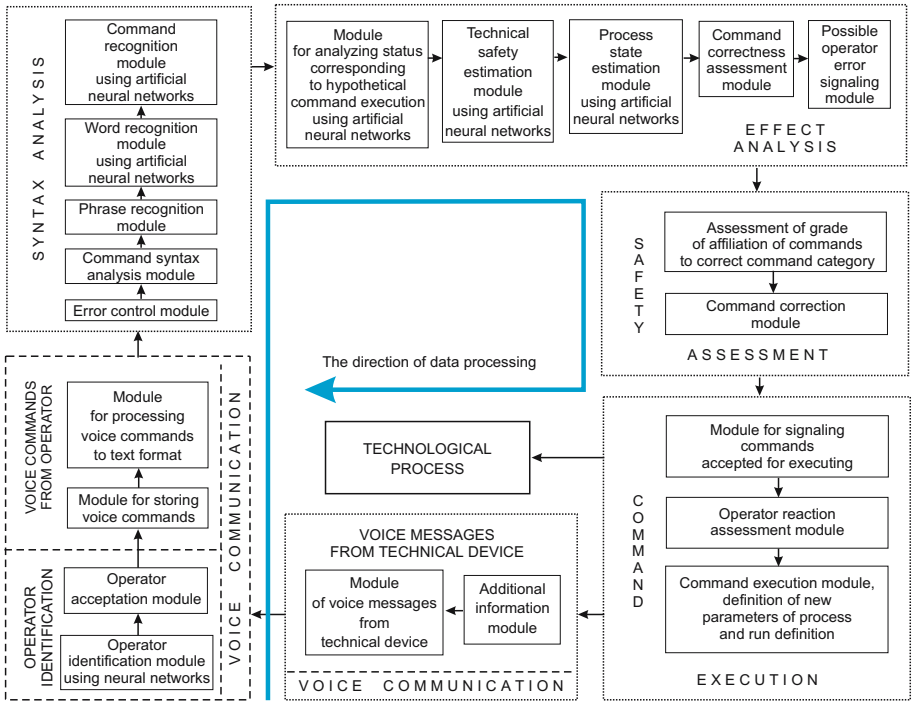


Fig. 2. Scheme of the intelligent layer of two-way voice communication

2 Command Safety Estimation

The effect analysis module, shown in Fig. 3a, makes analysis of the recognised command. The technical safety of the technological device is checked by analysing the state of execution of the commands required to have been done as well as the commands to execute in next decisions. The process parameters to be modified by executing the command are checked and the allowable changes of the parameter values are determined. The analysis of the parameter values is based on the technological process features. The values of the parameter changes are the input signals of the neural network of the process state assessment system. The neurons of the neural network represent solutions to the diagnostics problem. The neural network also makes an estimation of the grade of safety of the recognised command. The system for checking the state of the automatic device for grinding of small ceramic elements that is shown in Fig. 3c, before executing next commands is presented in Fig. 3d. The technological safety assessment system, shown in Fig. 3b, is based on a neural network which is trained with the model of work of the technological device. New values of the process parameters are the input signals of the neural network [6]. As the work result of the system, voice messages from the technological device to the operator about the possibility of executing of the command are produced [4,5].

There was an algorithm created for assessing the technological safety of commands. In Fig. 4, the lines present dependence of the force on the grinding

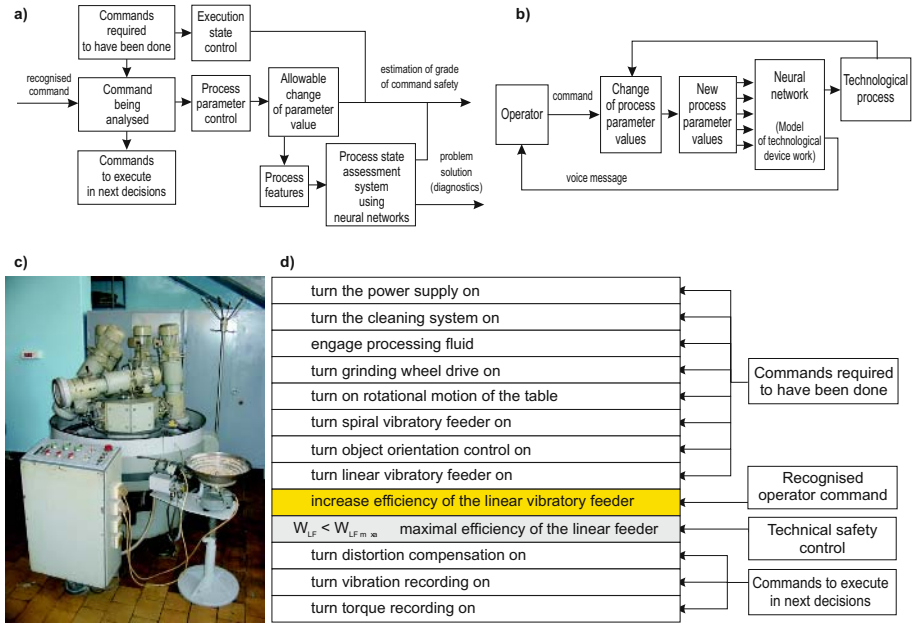


Fig. 3. Scheme of the command effect analysis and safety assessment system

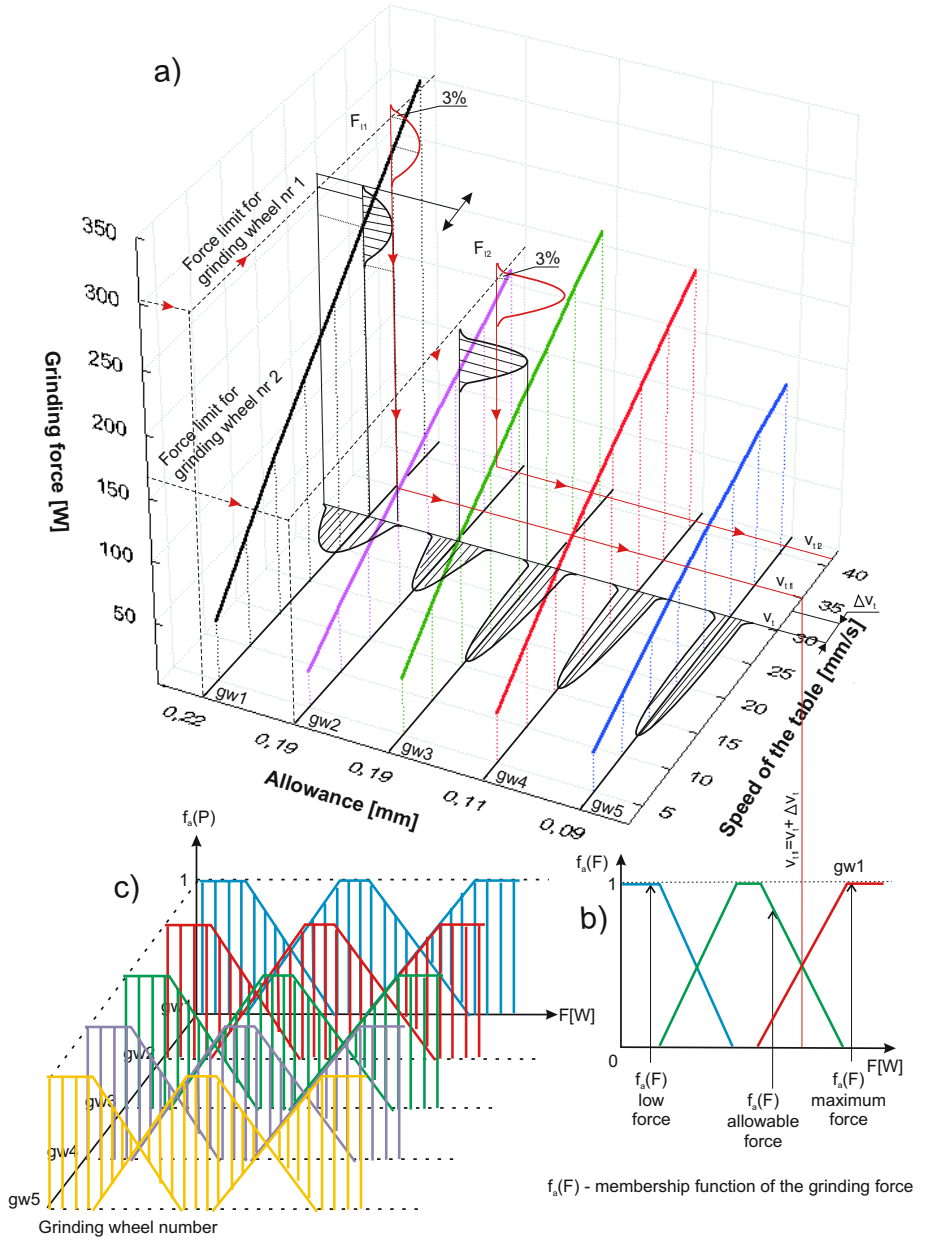


Fig. 4. Algorithm for assessing the technological safety of commands based on the real technological process

process parameters for particular grinding wheels. Basing on the specified criteria, there is the grinding force limit determined for each grinding wheel. Basing on the grinding force limit, there is the table speed limit assigned. According

to the operator's command, if the increase of the speed makes a speed of the table smaller than the smallest speed determined from the force limit for all the grinding wheels, then the command is safe to be executed.

3 Research Results

The simulation set of the technological device diagnostics and the process state assessment, built for creating and training artificial neural networks is shown in Fig. 5a. The neural networks are trained with the model of the technological process. The applied neural network architecture is presented in Fig. 5b. The networks consist of two layers of neurons with the competitive mechanism.

The ability of the neural network to learn to recognise specific process states depends on the number of learning epochs. The specified time of learning enables the network to minimize the error so that it could work more efficiently. Based on the research, the following conclusion has been reached as shown in Fig. 5c.

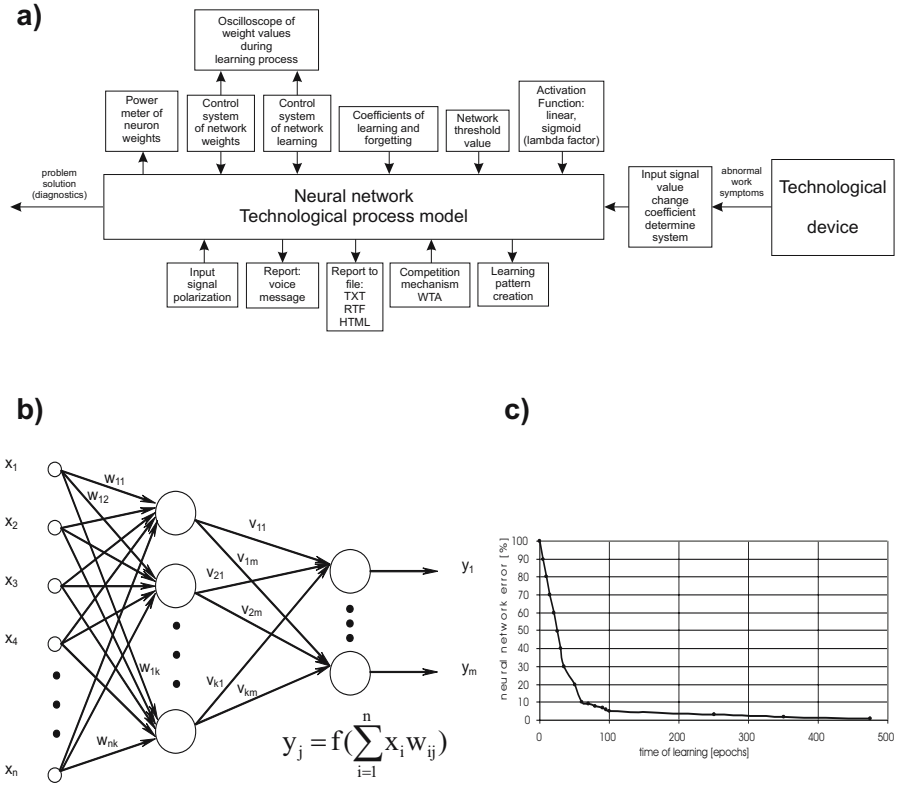


Fig. 5. Neural network simulations of the technological process models, neural network architecture and error rate

The error is about 20% at learning time equals 50 epochs and 5% at 100 epochs. The error has been dropped for about 90% after training with 60 series of all patterns.

4 Conclusions and Perspectives

In the automated processes of production, the condition for safe communication between the operator and the technological device is analyzing the state of the technological device and the process before the command is given and using artificial intelligence for assessment of the technological effects and safety of the command. In operations of the automated technological processes, many process states and various commands from the operator to the technological device can be distinguished. A large number of combined technological systems characterize the realization of that process. In complex technological processes, if many parameters are controlled, the operator is not able to analyse a sufficient number of signals and react by manual operations on control buttons. The research aiming at developing an intelligent layer of two-way voice communication is very difficult, but the prognosis of the technology development and its first use shows a great significance in efficiency of supervision and production humanisation.

References

1. Kacalak, W., Majewski, M.: Recognition and Safety Estimation of Voice Commands in Natural Language Using Artificial Neural Networks. Artificial Neural Networks in Engineering ANNIE 2004 St. Louis. ASME Press New York (2004) 831-836
2. Kacalak, W., Majewski, M.: Selected Problems of Automatic Recognition and Evaluation of Voice Commands in Natural Language Given by the Operator. World Multi-Conference on Systemics, Cybernetics and Informatics, Orlando, Florida (2005)
3. Kacalak, W., Majewski, M.: Intelligent Layer of Two-Way Speech Communication of the Technological Device with the Operator. International Conference on Artificial Intelligence ICAI 2005, Las Vegas, Nevada. CSREA Press (2005) 779-785
4. Majewski, M., Kacalak, W.: Selected Problems of Automatic Evaluation of Commands Using Artificial Neural Networks. International Conference on Intelligent Computing ICIC2005, Hefei Anhui, China (2005) 3605-3614
5. Majewski, M., Kacalak, W., Selected Problems of Automatic Evaluation of Commands Using Artificial Neural Networks. Engineering Mechanics International Journal for Theoretical and Applied Mechanics 12(3) (2005) 185-192
6. Principe, J.C., Euliano, N.R., and Lefebvre, W.C.: Neural and Adaptive Systems: Fundamentals through Simulations. John Wiley and Sons Inc., New York (2000)
7. O'Shaughnessy, D.: Speech Communications: Human and Machine. IEEE Press, New York (2000)

Natural Language Human-Machine Interface Using Artificial Neural Networks

Maciej Majewski and Wojciech Kacalak

Technical University of Koszalin,
Department of Mechanical Engineering,
Raclawicka 15-17, 75-620 Koszalin, Poland
{maciej.majewski, wojciech.kacalak}@tu.koszalin.pl

Abstract. In this paper there is a natural language interface presented, which consists of the intelligent mechanisms of human identification, speech recognition, word and command recognition, command syntax and result analysis, command safety assessment, technological process supervision as well as human reaction assessment. In this paper there is also a review of the selected issues on recognition of speech commands in natural language given by the operator of the technological device. A view is offered of the complexity of the recognition process of the operator's words and commands using neural networks made of a few layers of neurons. The paper presents research results of speech recognition and automatic recognition of commands in natural language using artificial neural networks.

1 Intelligent Two-Way Speech Communication

If the operator is identified and authorized by the natural language interface in Fig. 1, a produced command in continuous speech is recognized by the speech recognition module and processed to the text format. Then the recognised text is analysed with the syntax analysis subsystem. The processed command is sent to the word and command recognition modules using artificial neural networks to recognise the command, which next is sent to the effect analysis subsystem for analysing the status corresponding to the hypothetical command execution, consecutively assessing the command correctness, estimating the process state and the technical safety, and also possibly signalling the error caused by the operator. The command is also sent to the safety assessment subsystem for assessing the grade of affiliation of the command to the correct command category and making corrections. The command execution subsystem signalises commands accepted for executing, assessing reactions of the operator, defining new parameters of the process and run directives [4]. The subsystem for voice communication produces voice commands to the operator [5].

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1161–1166, 2006.

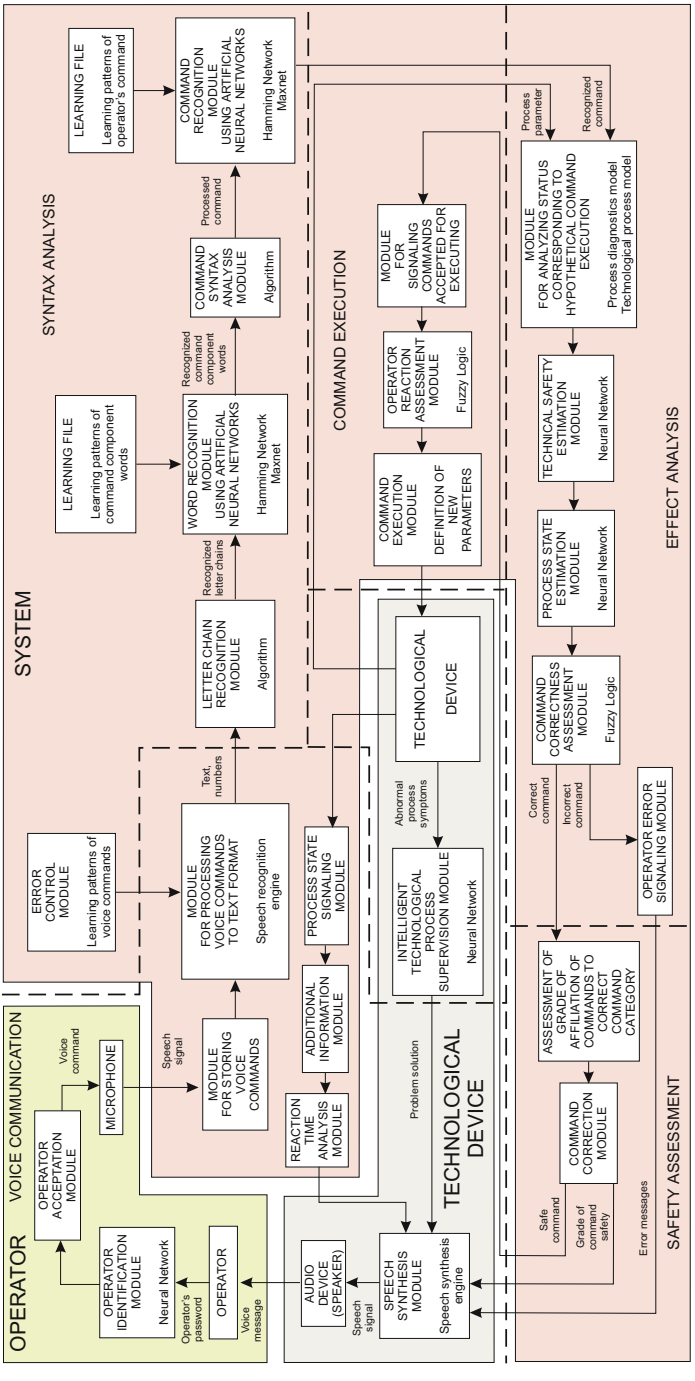


Fig. 1. Architecture of the natural language human-machine interface

2 Recognition of Commands in Natural Language

In the automatic command recognition system as shown in Fig. 2, the speech signal is processed to text and numeric values with the module for processing voice commands to text format. The speech recognition engine is a continuous density mixture Gaussian Hidden Markov Model system which uses vector quantization for speeding up the Euclidean distance calculation for probability estimation [1,2]. The system uses context dependent triphonic cross word acoustic models with speaker normalization based on vocal tract length normalization, channel adaptation using mean Cepstral subtraction and speaker adaptation using Maximum Likelihood Linear Regression. The separated words of the text are the input

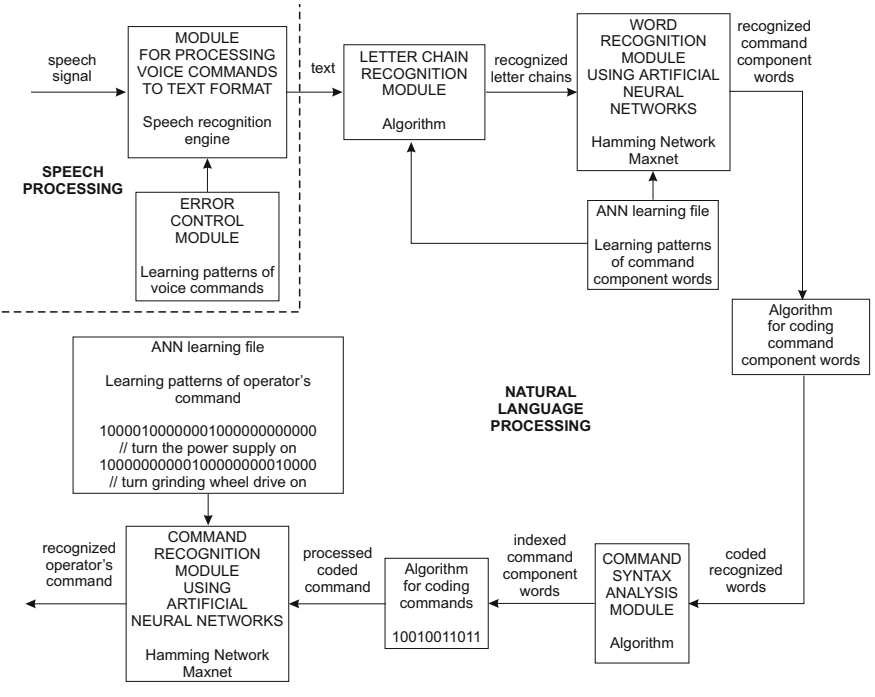


Fig. 2. Scheme of the automatic command recognition system

signals of the neural network for recognizing words. The network has a training file containing word patterns. The network recognizes words as the operator's command components, which are represented by its neurons. The recognized words are sent to the algorithm for coding words. Next the coded words are transferred to the command syntax analysis module. It is equipped with the algorithm for analysing and indexing words. The module indexes words properly and then they are sent to the algorithm for coding commands. The commands

are coded as vectors and they are input signals of the command recognition module using neural network. The module uses the 3-layer Hamming neural network in Fig. 3, either to recognize the operator's command or to produce the information that the command is not recognized. The neural network is equipped with a training file containing patterns of possible operator's commands [3].

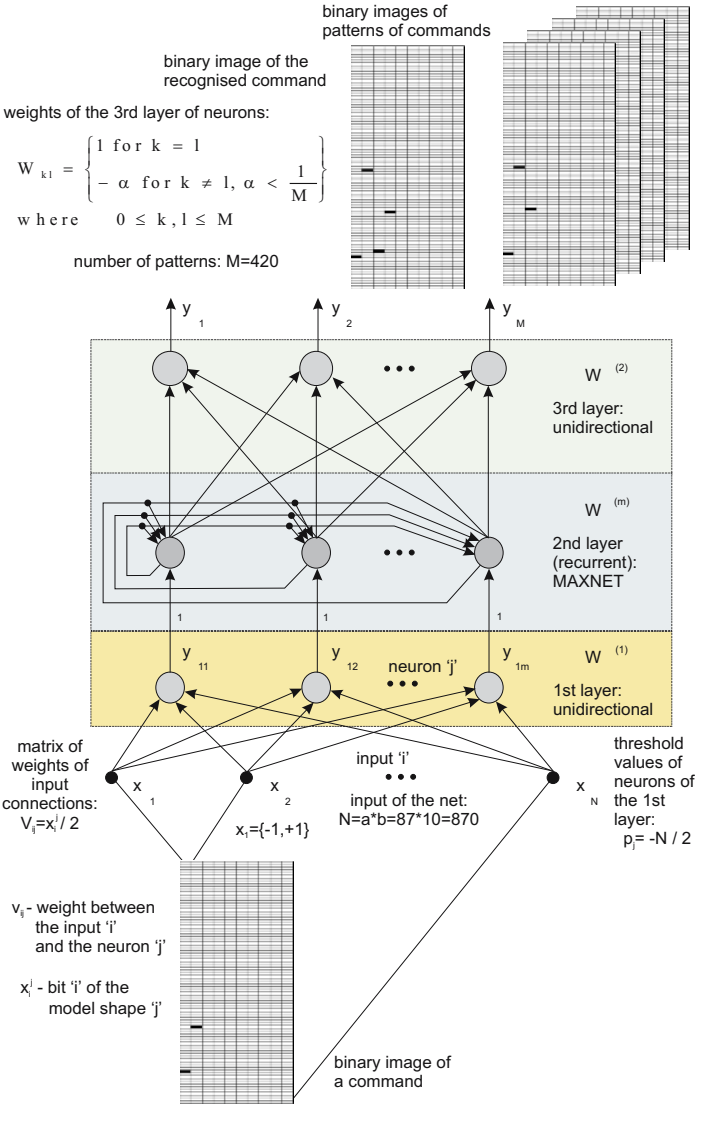


Fig. 3. Scheme of the 3-layer neural network for automatic command recognition

3 Research Results of Automatic Command Recognition

As shown in Fig. 4a, the speech recognition module recognizes 85-90% of the operator's words correctly. As more training of the neural networks is done, accuracy rises to around 95%. For the research on command recognition at different noise power, the microphone used by the operator is the headset. As shown in Fig. 4b, the recognition performance is sensitive to background noise. The recognition rate is about 86% at 70 dB and 71% at 80 dB. Therefore, background noise must be limited while giving the commands. For the research on command

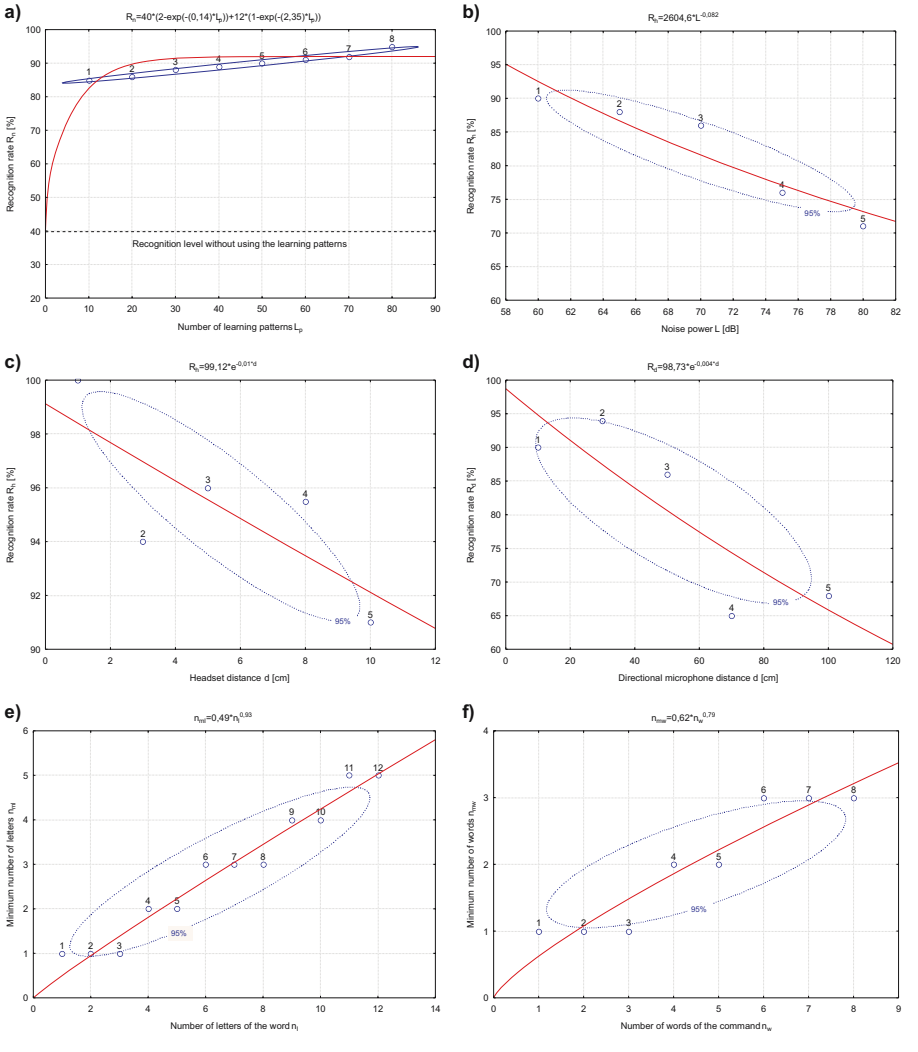


Fig. 4. Speech and command recognition rate

recognition at different microphone distances, the microphone used by the operator is the headset. As shown in Fig. 4c, the recognition rate decreases when the headset distance increases. The recognition rate has been dropped for 9% after the headset distance is changed from 1 to 10 cm. Also for the research on command recognition at different microphone distances, the microphone used by the operator is the directional microphone. As shown in Fig. 4d, the recognition rate after 50 cm decreases reaching rate about 65%. As shown in Fig. 4e, the ability of the neural network to recognise the word depends on the number of letters. The neural network requires the minimal number of letters of the word being recognized as its input signals. As shown in Fig. 4f, the ability of the neural network to recognise the command depends on the number of command component words. Depending on the number of component words of the command, the neural network requires the minimal number of words of the given command as its input signals.

4 Conclusions and Perspectives

The condition of the effectiveness of the presented system is to equip it with mechanisms of command verification and correctness. In operations of the automated technological processes, many process states and various commands from the operator to the technological device can be distinguished. A large number of combined technological systems characterize the realization of that process. In complex technological processes, if many parameters are controlled, the operator is not able to analyze a sufficient number of signals and react by manual operations on control buttons. The research aiming at developing an intelligent layer of two-way voice communication is very difficult, but the prognosis of the technology development and its first use shows a great significance in efficiency of supervision and production humanization.

References

1. Kacalak, W., Majewski, M.: Automatic Recognition of Voice Commands in Natural Language Using Artificial Neural Networks. Artificial Neural Networks in Engineering ANNIE 2004 Conference, St. Louis. ASME Press, New York (2004) 831-836
2. Majewski, M., Kacalak, W.: Intelligent Human-Machine Voice Communication System. Engineering Mechanics International Journal 12(3) (2005) 193-200
3. Majewski, M., Kacalak, W.: Intelligent Layer of Human-Machine Voice Communication. 9th International Conference KES2005, Melbourne, Australia (2005), Lecture Notes in Computer Science 3683. Springer-Verlag (2005) 930-936
4. Majewski, M., Kacalak, W.: Intelligent Human-Machine Speech Communication System. International Conference on Intelligent Computing ICIC2005, Hefei Anhui, China (2005) 3441-3450
5. O'Shaughnessy, D.: Speech Communications: Human and Machine. IEEE Press, New York (2000)

Implementing a Chinese Character Browser Using a Topography-Preserving Map

James S. Kirk

Union University, Jackson, TN, USA
jkirk@uu.edu

Abstract. The Chinese Character Browser is a user interface designed for the search and exploration of a database of Chinese characters, Chinese pronunciations (pinyin), and English definitions. The browser uses a technology based upon Kohonen's self-organizing map to map the 10-dimensional feature vector describing each Chinese character onto a discrete two-dimensional grid, which forms the foundation for the browser. The Chinese Character Browser was designed to demonstrate the importance not only of topology preservation, but topography preservation in such mappings. In brief, to the extent that a mapping is topography-preserving, the structure of the output map grid can reflect the structure of the original data at several levels of granularity simultaneously, allowing the assumption of a hierarchical organization in the output map that corresponds to the structure of the input data set. This can significantly speed up search as well as make the mapping more useful for a variety of applications.

1 Introduction

1.1 Related Self-Organizing Map Applications

Numerous applications of the self-organizing map (SOM) have arisen because of the ability of the SOM to condense a data set. This is accomplished through vector quantization during the competitive stage of SOM training, and through regression on the data, which is represented in the form of adjacencies on the output map lattice. Applications for the SOM have naturally arisen in data mining, compression, summarization, and visualization. A related application for the SOM has been to provide a user interface to high-dimensional data. Probably the best-known such interfaces have been produced at the Helsinki University of Technology. One such system, WEBSOM [8] [10], provides a world wide web interface to over one million documents from Usenet groups based upon contents of each document.[1] A similar interface has been applied to an even larger set of documents, a collection of nearly seven million patents [8]. Yet another SOM-based interface, PicSOM,[2] uses the hierarchical Tree-Structured SOM to provide content-based image retrieval using features derived from image colors, textures, and shapes [9] [11].

[1] The WEBSOM interface is available at http://websom.hut.fi/websom/comp.ai.neural-nets-new/html/root.html (accessed Nov. 2005).

[2] The PicSOM interface is available at http://www.cis.hut.fi/picsom/ (accessed Nov. 2005).

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1167 – 1172, 2006.

The Chinese Character Browser (CCB) is similar in concept to the applications mentioned above, but an attempt is made to impose a stricter requirement on the mapping than does the SOM. Whereas a well-trained SOM preserves the topology of the input data set, applications such as the Chinese Character Browser would benefit from the more stringent standard of topography preservation.

1.2 Definition and Importance of Topography Preservation

The SOM performs a mapping Ω from an often continuous, higher-dimensional space V to a discrete lower-dimensional space A, producing what has often been loosely termed a "topology-preserving" mapping. Because of reduction of dimensionality, it is usually impossible for the mapping to represent the data with sufficient resolution without folding into the input space. In fact, the image of a one-dimensional SOM forming a Peano-like space-filling curve in two-dimensional data space has become something of an icon for the SOM. Therefore, although a well-trained SOM can usually preserve continuity in the inverse mapping $\Omega^{-1}: A \rightarrow V$, it can rarely preserve continuity in the mapping $\Omega: V \rightarrow A$.

For many applications, it would be beneficial to use a mapping that, unlike the SOM, is based on global as well as local relationships among data. Attempts at such mappings could use multidimensional scaling (MDS) methods, or the Sammon projection. Recently, genetic methods have been applied to this problem [3] [4] [5]. A novel heuristic approach was applied by Su and Chang [12].

The value of a mapping that preserves a useful balance of global and local relationships among data is hard to overestimate. Kirk and Zurada [6] have termed a mapping "topography-preserving" to the degree that it preserves data relationships at multiple levels of granularity. To the extent that a mapping is topography-preserving, it imposes an implicit hierarchy on prototype vectors that corresponds to the structure of the original data on several levels of granularity, or resolution. In this way, it can allow a single codebook vector to serve as the prototype vector not only for data points in its own receptive field or Voronoi cell, but also, at a lower granularity, as a prototype for data mapping to neurons nearby in the discrete output space, A.

2 Design of the Chinese Character Browser (CCB)

2.1 Identification of a Suitable Data Vector to Represent Chinese Characters

The first stage in the design of the browser involved locating Chinese computer input methods and dictionary lookup methods in which similar input values identify Chinese characters that are visually similar. If dist($\boldsymbol{m}$, $\boldsymbol{n}$) indicates some function that returns a measure of dissimilarity between vectors $\boldsymbol{m}$ and $\boldsymbol{n}$, then this goal can be represented as follows:

$$\text{dist}(\boldsymbol{i}_p,\ \boldsymbol{i}_q) > \text{dist}(\boldsymbol{i}_p,\ \boldsymbol{i}_r) \Rightarrow \text{dist}(\boldsymbol{c}_p, \boldsymbol{c}_q) > \text{dist}(\boldsymbol{c}_p,\ \boldsymbol{c}_r),$$

where $\boldsymbol{i}$ represents an input vector for some visual representation $\boldsymbol{c}$ of a Chinese character. Although it is a simple matter to apply a metric to numeric input vectors, the similarity between the corresponding characters must be judged perceptually. Clearly, some Chinese input methods are not useful for this. For example, pinyin,

which is frequently used in China to look up characters in a dictionary, cannot be adopted since many characters with identical pinyin values are disparate visually, and many characters which are visually similar have vastly different pinyin. For example, pinyin *qī* indicates the pronunciation of all of the following characters: 其, 起,七, 器, and 汽, which have little in common visually.

In contrast, there are several methods currently being used to look up Chinese characters in dictionaries and enter characters at a keyboard that may be applicable because the input codes identify visual elements of the characters. For example, the Four Corner System, which has been used in both Chinese and Japanese dictionaries, divides stroke shapes into ten atomic classes and identifies a Chinese character by the class of the shape at different locations in the character. It uses five-component vectors. For example, Four Corner codes with 6 as the second component signify a "box" in the upper right corner, thus partitioning the universe of characters according to a visually-identifiable element. Examples include 旭 (Four Corner code 46010), 況 (36110), and 器 (66663).

Another applicable input method is Jack Halpern's SKIP method, which has been used in the *NTC's New Japanese-English Character Dictionary* [2]. In this scheme, which uses three-component vectors, the first component reflects the overall organization of the Chinese character, classifying it as either divided between left and right sections, top and bottom sections, enclosing and enclosed sections, or solid. The second and third components of the vector indicate the number of strokes in each classified section of the character.

In addition to the methods above, character lookup by radical and by stroke count are also based on visually identifiable features of characters. The 214 Chinese radicals are a set of standard character subcomponents. They have been used to classify characters since the late Ming dynasty, before which there was another radical system utilizing a greater number of radicals. As an example, radical 尸 makes up a visually-identifiable component of characters 尼, 局, 展, and 屏. Stroke count also provides a crude visual classification. For example, it may reasonably be thought that characters with three strokes like 三 and 千 are more similar to each other visually than they are to characters with twenty-five strokes, like 鬱 or 籱.

In the original version of the CCB, the data vectors representing Chinese characters are made up of ten components. Each vector is produced by concatenating the Four Corner values, the SKIP codes, the radical number, and the stroke count for the character identified by the vector. Each component of the data vector is shifted and scaled to preprocess the values in terms of standard deviations from the mean.

2.2 Training of the Topography-Preserving Map

The mapping from ten to two dimensions could be accomplished with a self-organizing map. However, as discussed above, it is important in applications such as the CCB to impose the requirement of global as well as local ordering on the codebook vectors. One goal is to prevent as much as possible the mapping of visually dissimilar characters to neurons that are nearby on the output grid. Similarly, a goal is to prevent the mapping of visually similar characters to neurons that are distant on the output grid. In other words, it is important to avoid discontinuities in both the

mapping Ω and the inverse mapping Ω^{-1}. Related to this is the desire for the mapping to impose the hierarchical structure of the original data set as much as possible when prototype vectors are being assigned to neurons on the output lattice. This will allow the browser to traverse the character set at varying levels of granularity, allowing a user to search first for a character by finding the most similar among a set of vastly different characters and then continuing the search by doing comparisons at finer and finer levels of resolution.

Because of the reasons discussed above, it was determined to use the training heuristic proposed by Su and Chang [12], which assigns the distance between neurons $\boldsymbol{r}_m$ and $\boldsymbol{r}_n$ on the output lattice A according to the distances between their associated weights (prototype vectors) in input space V. The goal is that data points $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ which map to the same, or to nearby neurons in A should be nearby in V:

$$\text{dist}_V(\boldsymbol{v}_i, \boldsymbol{v}_j) < \delta \Rightarrow \boldsymbol{r}_m = \Omega(\boldsymbol{v}_i) = \Omega(\boldsymbol{v}_j), \text{ or}$$
$$\text{dist}_V(\boldsymbol{v}_i, \boldsymbol{v}_j) < \delta \Rightarrow \boldsymbol{r}_m = \Omega(\boldsymbol{v}_i), \boldsymbol{r}_n = \Omega(\boldsymbol{v}_j), \text{ where } \text{dist}_A(\boldsymbol{r}_m, \boldsymbol{r}_n) < \gamma \, .$$

Similarly, it is desired that the data vectors within the receptive fields of neurons that are nearby in A will themselves be nearby in V:

$$\text{dist}_A(\boldsymbol{r}_m, \boldsymbol{r}_n) < \delta \Rightarrow \text{dist}_V(\boldsymbol{v}_i, \boldsymbol{v}_j) < \gamma, \text{ where}$$
$$\boldsymbol{v}_i \in \Omega^{-1}(\boldsymbol{r}_m) \text{ and } \boldsymbol{v}_j \in \Omega^{-1}(\boldsymbol{r}_n) \, .$$

In determining the size of the output lattice, principal component analysis was used on the data set, revealing that the variance along the first two principal components had a ratio of approximately 1.7. The ratio of the sides of the rectangular output grid was set accordingly.

2.3 Implementing the Browser Interface

The Chinese Character Browser interface was built using HTML, PHP, and JavaScript to take advantage of the functionality of existing web browsers. The database of Chinese characters, pinyin pronunciations, and English definitions is a relational database created from Jim Breen's KANJIDIC database [1], which contains 6355 Chinese characters. The browser accesses the trained topography-preserving map, which indicates which characters should be loaded into the browser at a given time. A query is then sent to the database to load the character, pronunciation, and English definition information to be displayed for the user in the CCB display panel.

Typical use of the browser would proceed as follows. When a user has found a Chinese character that he or she does not recognize in some text, the CCB is started at its lowest resolution and displays 25 prototype Chinese characters from across the entire domain of the character set. Using the computer mouse, the user selects the character which appears most similar to the unknown character in the browser display panel. This causes the selected character to appear in the highlighted central cell of the display panel, and it shifts the mapping to the next higher degree of search resolution. In addition to the option of selecting a visible character, the user may click one of the directional arrows in the browser control panel. This causes the browser to

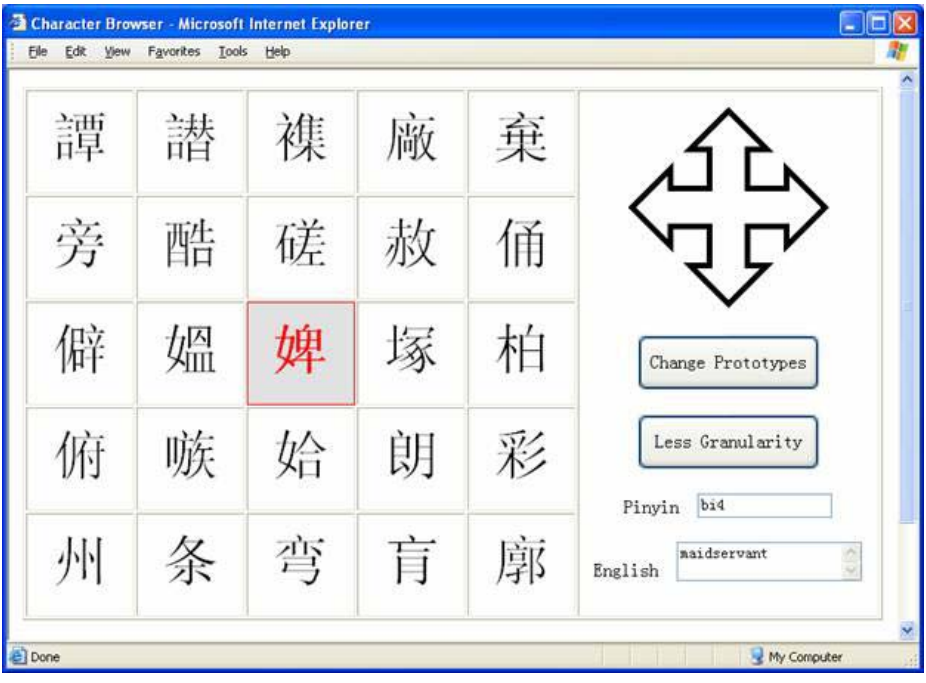


Fig. 1. Chinese Character Browser interface

load the set of characters that appear at the same level of search granularity but in the indicated direction on the two-dimensional map output grid. A user may also click the *Less Granularity* button on the browser control panel, which causes prototype characters to be loaded from the next lower level of resolution, keeping the highlighted character as the prototype for the central display cell. Finally, the *Change Prototypes* button allows a user to display different characters from the receptive fields of the currently displayed prototypes, and to treat these as new prototypes at this level of granularity. At each step in the character browsing, the pinyin pronunciation and English definition of the central, highlighted character appears in the lower part of the control panel.

Because of the assumption of topography preservation and the preservation of hierarchical structure of the data that it implies, the traversal of the data set at different levels of granularity is much simplified. Changing to a lower level of granularity for the search involves simply increasing the size of the offset between characters to be displayed. At the highest level of resolution, each character is its own prototype vector, and the offset between characters to be displayed is merely 1.

3 Conclusions

The usefulness of the Chinese character browser depends upon how well it meets two challenges. First, it is essential that the feature vectors used in map training can

describe Chinese characters in a way that corresponds to the user's perception of visual similarity and difference. Second, the implementation of the browser relies upon an acceptable degree of topography preservation during dimensionality reduction as the map is trained. To the extent that these challenges are met, the CCB may prove useful as an interface to an online dictionary, especially one designed for foreign learners of Chinese, who are often intimidated by the traditional methods of looking up Chinese characters. In addition, the CCB remains useful as an experiment in the preservation of data structure during dimensionality reduction.

References

1. Breen, J.: KANJIDIC Documentation. Unpublished. Available at http://www.csse.monash. edu.au/~jwb/kanjidic.html, accessed Nov. 2005
2. Halpern, J.: NTC's New Japanese-English Character Dictionary. National Textbook Co., Lincolnwood, Illinois (1993)
3. Kirk, J.S.: Algorithms to Improve Topography Preservation in Self-Organizing Maps. Ph.D. dissertation, University of Louisville (2003)
4. Kirk, J.S., Zurada, J.M.: An Evolutionary Method of Training Topography-Preserving Maps. In: Proceedings of the 2001 INNS-IEEE International Joint Conference on Neural Networks, held in Washington, D.C., July 14-19 (2001) 2230–4
5. Kirk, J.S., Zurada, J.M.: Motivation for a Genetically-Trained Topography-Preserving Map. In: Proceedings of the 2002 International Joint Conference on Neural Networks, held in Honolulu, HI, May 12-17 (2002) 394–9
6. Kirk, J.S., Zurada, J.M.: Topography-Enhanced BMU Search in Self-Organizing Maps. In Yin, F., Wang, J., and Guo, C. (eds.): Advances in Neural Networks – ISNN 2004, Lecture Notes in Computer Science, Vol. 3174. Springer, Berlin (2004) 695–700
7. Kohonen, T.: Self-Organizing Maps, 3rd edn. Springer, Berlin (2001)
8. Kohonen, T., Kaski, S., Lagus, K., Salojärvi, J., Honkela, J., Paatero, V., Saarela, A.: Self Organization of a Massive Document Collection. IEEE Trans. Neural Networks 11(3) (2000) 574–585
9. Koskela, M., J. Laaksonen, S. Laakso, and E. Oja. The PicSOM Retrieval System: Description and Evaluations. In: Eakins, J.P., Enser, P.G.B. (eds.): Proceedings of Challenge of Image Retrieval. Brighton, UK. (2000)
10. Lagus, K., Kaski, S., Kohonen, T.: Mining Massive Document Collections by the WEBSOM Method. Information Sciences 163(1-3) (2004) 135–156
11. Oja, E., Laaksonen, J., Koskela, M., and Brandt, S.: Self-Organizing Maps for Content-Based Image Database Retrieval. In: Oja, E., Kaski, S. (eds.): Kohonen Maps. Elsevier, Amsterdam (1999) 349–62
12. Su, M.C., Chang, H.T.: Fast Self-Organizing Feature Map Algorithm. IEEE Trans. Neural Networks 11(3) (2000) 721–733

A Soft Computing Method of Economic Contribution Rate of Education: A Case of China

Hai-xiang Guo, Ke-jun Zhu, Jin-ling Li, and Yan-min Xing

School of Management, China University of Geosciences, Wuhan, Hubei, 430074 China
faterdumk0732@sina.com

Abstract. Economic contribution rate of education is the key factor of education economy. In this paper, a soft computing method of economic contribution rate of education is proposed. The method is composed of four steps: The first step is doing fuzzy soft-clustering to object system based on levels of science technology and getting optimal number of clusters, which determines number of fuzzy rules. The second step is that the fuzzy neural networks FNN1 from human capital to economic growth is constructed and we obtain economic contribution rate of human capital α_k . The third step is that the fuzzy neural networks FNN2 from education to human capital is constructed and we obtain human capital contribution rate of education α'_k . The fourth step is calculating economic contribution rate of education $ECE_k = \alpha_k \times \alpha'_k$. At last, the economic contribution rate of education of China is obtained.

1 Introduction

Education can promote economy development directly which was confirmed by many economists. [1-4] The research on ECE is to calculate the increase of GDP induced from unit investment of education. Since Schultz (1961) and Denison (1962) published the innovative evaluation of education usage on America economic growth in early 60s, economists in respective countries have done much research [2,5]. Chinloy (1980) used Translog function based on the interrelation of factors in labor index, Bishop (1989) used general intellectual achievement and Barro (1992) did comparison between countries and so on, which all rooted in Romer-Lucas economic model. Recent years, Shan-mai Wang, Zhen-sen Qu, Xian-zuo Fan, Tie-ming Zhang and Lan desheng who are all education economist in China also published many papers and books about economic contribution rate of education. They proposed statistic measure of education-labor productivity-economy development on the basis of Marxism theory and useful experience of west economy. [2,5] Traditional measure of economic contribution rate of education is not only transform people's old idea that education is pure consumption, but also make people have a quantitative cognition on economic contribution rate of education. But social economic system is large and complicated and influencing factors of social economic system are extensive. With the development of technology and economy, the factors are increasing, at the same time these factors interact to effect economic growth. Facing to the large, complicated and uncertain social economy system, the traditional statistical measures with precise ideas are deficient. In order to measure economic contribution rate of education, the tools or methods must be equipped with characters as follows:

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1173 – 1180, 2006.

① The measure tools or methods should have high capabilities of nonlinear mapping and parallel processing. At the same time, they should have the capabilities of self-learning and self-adaptive, because social economy system is high nonlinear and complicated.

② The measure tools or methods should have capabilities of dealing with fuzzy concept and using qualitative cognition of human beings because social economy system is uncertain and fuzzy.

③ The measure tools or methods should have capabilities of robust and stabilization because information or data is incomplete.

Soft computing just has the charters above. Soft computing is a new method used to set up intelligent system in recently years, which has attracted many researchers. [6] At present, people need intelligent systems [7,8] combining different knowledge, technologies and methods in order to deal with many practical problems under the uncertain and imprecise conditions. Soft computing is corresponding to human brain that cooperates with many different technologies when dealing with practical problems. Artificial neural networks (ANN) deals with pattern recognized and self-adapt adjustment. Fuzzy system (FS) deals with inference and making decision. Genetic algorithm (GA) optimizes the system. ANN, FS and GA exert each other's functions and constitute soft computing jointly.

We propose the algorithm of economic contribution rate of education (see Fig 1). Labor force with same human capital have different contribution rates under different productivity condition, even they are incomparable. So we must classify object system based on the levels of technology and science firstly. Here classification is realized by GA-ISODATA. [9] Secondly economic contribution rate of human capital α_k is obtained by constructing fuzzy neural networks FNN1 from human capital to economic growth. Human capital contribution rate of education α'_k is obtained by constructing fuzzy neural networks FNN2 from education to human capital. Finally, economic contribution rate of education $ECE_k = \alpha_k \times \alpha'_k$ is computed.

2 Soft Classifications by Scientific Technology

The objective of soft classification is to partition samples into c clusters, and data included in same cluster is similar. Fuzzy c-means (FCM) algorithm proposed by Bezdek is most widely used one. [10] This algorithm requires pre-defining the number of clusters (c); however, it is not possible to know c in advance. Therefore, how to obtain the best classification and corresponding optimal number of clusters is an important problem. The paper uses GA-ISODATA algorithm to partition 31 areas in China by science technology.

2.1 Select Index and Sample

If the real output is y and the number of input elements is n, the general form of production function is [11]:

$$Y = F(x_1, x_2, \cdots, x_n; t) \tag{1}$$

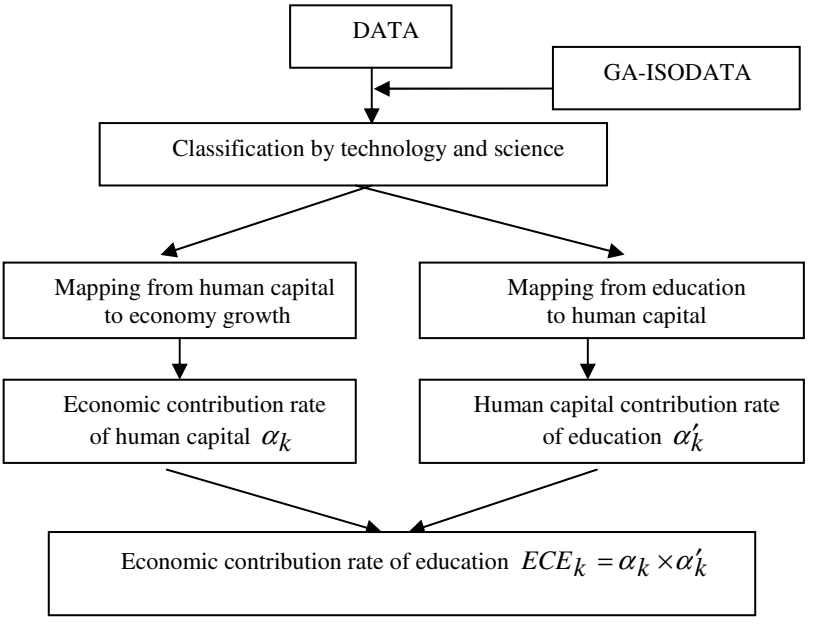


Fig. 1. Flow of measuring economic contribution rate of education

which represents certain dependent relationship of output and the input elements. In order to achieve soft-partition for productivity of 31 areas in China, we extend the well-known C-D function (Cobb-Douglas) as follows:

$$Y = F(x_1, x_2, \cdots, x_n; t) = AK_t^{\alpha} L_t^{\beta} \tag{2}$$

where α、β are constants, and $\alpha + \beta = 1$. Here K and L represent input of capital and input of labor respectively. According to (2) we can obtain (3) easily:

$$A = YK_t^{-\alpha} L_t^{-\beta} \tag{3}$$

(3) expresses that technology improvement of an area is dependent on output, input of capital and labor of this area, Based on this dependent relationship we do soft-partition for technology improvement of 31 areas in China. So we make average GDP in 2002 of each area denotes Y, the sum of fixed assets from 1991 to 2003 after doing discount and depreciation to divide total population in 2002 for each area denotes K, average human capital of laborers denotes L.

2.2 Analysis of Clustering

We can obtain the optimal number of cluster $c = 3$ applying the algorithm GA-ISODATA. The result of clustering shown in table 1 is obtained according to the principle of the maximal membership value. Each cluster of technology improvement in China is exactly identified. The first cluster stands for developed areas. The second

Table 1. The result of classification for 31 areas in China

Class	Areas
One	Beijing、Shanghai 、Tianjin
Two	Hebei 、Neimenggu 、Liaoning 、Jilin 、Heilongjiang 、Jiangsu 、Zhejiang 、Fujian、Shandong、Hubei、Guangdong、Hainan、Xinjiang
Three	Shanxi、Anhui、Jiangxi、Henan、Hunan、Guangxi、Chongqing、Sichuan、Guizhou、Yunnan、Xizang、Shan'xi、Gansu、Qinghai、Ningxia

cluster stands for coastal areas, old industry base; developing areas in middle part and part of outlying districts. The third cluster stands for underdeveloped areas in west and depressed areas in middle part. The result completely accords with technology improvement of each area and economic development in China.

3 Structure of Fuzzy Neural Networks FNN1 from Human Capital to Economic Growth

Fuzzy Neural Networks combine fuzzy logic with neural networks. Fuzzy logic uses into expressing brain macroscopically capability with language and concepts. Fuzzy Neural Networks deal with all kinds of fuzzy information based on membership function and series of rules. So FNN has many strong points like fuzzy logic and neural networks. This paper sets up FNN mapping from human capital to economy growth with three rules and linear conclusion, which we obtain economic contribution rate of human capital α_k .

3.1 FNN1 Structure and Learning Algorithm

The FNN1 structure is TSK model. The FNN1's consequence is three fuzzy rules based on the optimal classification above and FNN1 learning algorithm is BP algorithm. BP algorithm has two steps that one is forward propagation and the other is backward propagation.

3.2 Analyses of FNN1 Mapping Result

Inputs of FNN1 are average fixed assets, average human capital, average plantation acreage and output of FNN1 is average GDP. Then run the FNN1 by BP algorithm and get three fuzzy rules is shown in Table 2.

Table 2. Fuzzy rules of economy systems

Cluster	Precondition parameter						Consequence parameter			
	Fixed assets		Human capital		Plantation acreage		w_{i0}	w_{i1}	w_{i2}	w_{i3}
	wc	wg	wc	wg	wc	wg				
One	0.67	3.72	0.71	6.98	0.08	22.56	0.32	0.26	0.28	0.007
Two	0.22	7.49	0.50	5.13	0.31	3.06	0.20	0.06	0.11	0.003
Third	0.06	11.10	0.18	5.91	0.33	3.43	0.04	0.004	0.01	0.004

In the table 2, wc_{ij} and $\frac{1}{wg_{ij}}$ denote the center and width of gauss function that is membership function. Consequence parameters of rules are w_{ij}, $i = 1,2,3$, $j = 0,1,2,3$.

Table 3. Economic contribution rate of education of each factors

	w_{i1}	w_{i2}	w_{i3}	$s = \sum_{i=1}^{3} w_{ij}$	$\frac{w_{i1}}{s}$	$\frac{w_{i2}}{s}$	$\frac{w_{i3}}{s}$
One	0.2606	0.2778	0.0071	0.5457	47.77%	50.92%	1.31%
Two	0.0569	0.1095	0.0032	0.1698	33.57%	64.54%	1.89%
Third	0.0042	0.0104	0.0044	0.0192	22.34%	54.35%	23.31%

From table 2, when average fixed assets, average human capital and average plantation acreage are included by the fuzzy sets of Precondition of rule, Consequence parameters of rules are composed of average GDP of economy system. Obviously, The first cluster is better than the second cluster because $w_{10} > w_{20}$ and the second cluster is better than the third cluster because $w_{20} > w_{30}$. From table 3, in economy system of the first cluster, GDP contribution rate of fixed assets is 47.77%, GDP contribution rate of human capital is 50.92% and GDP contribution rate of plantation acreage is 1.31%. In economy system of the second cluster, GDP contribution rate of fixed assets is 33.57%, GDP contribution rate of human capital is 64.54% and GDP contribution rate of plantation acreage is 1.89%. In economy system of the third cluster, GDP contribution rate of fixed assets is 22.34%, GDP growth contribution rate of human capital is 54.35% and GDP contribution rate of plantation acreage is 23.31%.

In the times of industry, economy contribution of fixed assets and human capital are the same. So economy can grow in effect when fixed assets and human capital are harmonized.

4 Structure of Fuzzy Neural Networks FNN2 from Education to Human Capital

Human capital is composed of education, health and experience. So we can structure FNN2 like FNN1, which can get human capital contribution rate of education α'_k.

According to T.W.Sheulz's definition of human capital, human capital includes education, health and experience. See the function

$$H = f(e, h, s) + \varepsilon \tag{4}$$

While H denotes value of human capital, e denotes input of education, h denotes health, s denotes experience, ε denotes random variable. So we select index such as human capital, education capital, health capital and experience capital. Education

capital, health capital and experience capital are input of FNN2 and human capital is output of FNN2.

4.1 Compute Education Capital, Health Capital and Experience Capital

Education of the labor is divided into illiteracy or semi-illiteracy, elementary, junior, senior and university, five grades.

$$E_t = (0.5 \times ILL_t + 1 \times PRI_t + 1.2 \times JUN_t + 1.8 \times SEN_t + 2.5 \times HIGH_t) \times C_t \div P_t \quad (5)$$

While E_t denotes human capital formed by education in t year. ILL_t denotes the number of people with illiteracy and semi-illiteracy. PRI_t denotes the number of people with elementary. JUN_t denotes the number of people with junior. SEN_t denotes the number of people with senior. $HIGH_t$ denotes the number of people with university. P_t denotes the number of people.

We use the number of bed of sanitation, workers of sanitation, death rate of people, outlay of sanitation constituted of health capital.

Experience skill is accumulating of work, practice and so on. So we use professional workers, bargain of technology markets and real wage constituted of experience capital.

4.2 Analyses of FNN2 Mapping Result

Education capital, health capital and experience capital are input of FNN2 and human capital is output of FNN2. We use BP algorithm and obtain three rules is shown in table 4. In the table4, wc_{ij} and $\frac{1}{wg_{ij}}$ denote the center and width of gauss function that is membership function. Consequence parameters of rules are w_{ij}, $i = 1,2,3,\ j = 0,1,2,3$.

From table 5, in economy system of the first cluster, human capital contribution rate of health capital is 36.89%, human capital contribution rate of experience capital is 37.42% and human capital contribution rate of education capital is 25.7%. In economy system of the second cluster, human capital contribution rate of health capital is 25.47%, human capital contribution rate of experience capital is 33.46% and

Table 4. Fuzzy rules of human capital

Cluster	Precondition parameter						Consequence parameter			
	Education capital		Health capital		Experience capital		w_{i0}	w_{i1}	w_{i2}	w_{i3}
	wc	wg	wc	wg	wc	wg				
One	0.52	3.37	0.77	3.91	0.78	4.55	0.31	0.16	0.23	0.24
Two	0.43	2.10	0.22	8.13	0.32	3.37	0.31	0.14	0.09	0.12
Three	0.21	4.40	0.13	5.13	0.28	3.39	0.12	0.06	0.07	0.07

Table 5. Economic contribution rate of education of each factors

	w_{i1}	w_{i2}	w_{i3}	$s = \sum_{i=1}^{3} w_{ij}$	$\frac{w_{i1}}{s}$	$\frac{w_{i2}}{s}$	$\frac{w_{i3}}{s}$
One	0.1632	0.2343	0.2377	0.6353	25.7%	36.89%	37.42%
Two	0.1423	0.0882	0.1159	0.3467	41.07%	25.47%	33.46%
Three	0.0588	0.0698	0.0736	0.2024	29.06%	34.53%	36.41%

human capital contribution rate of education capital is 41.07%. In economy system of the third cluster, human capital contribution rate of health capital is 34.53%, human capital contribution rate of experience capital is 36.41% and human capital contribution rate of education capital is 29.06%.

5 Conclusions

According to the steps of measure economic contribution rate of education, economic contribution rate of human capital in the first cluster $\alpha_1 = 50.92\%$. Economic contribution rate of human capital in the second cluster $\alpha_2 = 64.54\%$. Economic contribution rate of human capital in the third cluster $\alpha_3 = 54.35\%$. Human capital contribution rate of education in the first cluster $\alpha_1' = 25.7\%$. Human capital contribution rate of education in the second cluster $\alpha_2' = 41.7\%$. Human capital contribution rate of education in the third cluster $\alpha_3' = 29.06\%$. So ECE in the first cluster $C_1 = \alpha_1 \times \alpha_1' = 50.92\% \times 25.7\% = 13.09\%$. In the second cluster $C_2 = \alpha_2 \times \alpha_2' = 64.54\% \times 41.7\% = 26.51\%$. In the third cluster $C_3 = \alpha_3 \times \alpha_3' = 54.35\% \times 29.06\% = 15.79\%$. See Table 6

Table 6. Economic contribution rate of education of each clusters

Cluster	Name of provinces	ECE
One	Beijing Tianjin Shanghai	13.09%
Two	Hebei Leimenggu Liaoling Jiling Heilongjiang Jiangsu Zhejiang Fujian Shandong Hubei Guangdong Hainan Xinjiang	26.51%
Three	Shanxi Anhui Jiangxi Henan Hunan Guangxi Chongqing Sichuan Guizhou Yunnan Xizang Shan'xi Gansu Qinghai ningxia	15.79%

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant No.70273044 and by the plan of outstanding young teacher in China University of Geosciences.

References

1. Min, W., Ding, X.: Review and Expectation of Advanced Education Economy in China. Advanced Education Research 1(3) (1999) 6-11
2. Wang, S.: Research of Input and Output of Education. Hebei Education Press, China (1999)
3. Zhang, T.: Education Industry. Advanced Education Press in Guangdong Province, Guangzhou (1998)
4. Yang, D.: Real Significant and Necessary of Developing Education Industry. Advanced Education Research 2(6) (1999) 51-56
5. Lai, D.: Education and Income Allocation. Beijing Normal University Press, Beijing (1998)
6. Guo, S., Gang, C.: Soft Computing in Information Science. Northeast University Press, Shenyang, China (2001)
7. Zeng, K., Zhang, N., Xu, L.: Linear T-S Fuzzy System' Sufficiency Condition. Autoimmunization Transaction 27(5) (2001) 606-612
8. Yao, M., Huang, Y.: Research of Fuzzy System. System Engineering Theory and Application 22(5) (2000) 98-103
9. Zhu, K., Su, S., Li, J.: Fuzzy c-mean of Optimal Clusters. System Engineering Theory and Application 23(3) (2005) 52-61
10. Bezdek J. C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
11. Fan, X.: Review and Expectation of Education Economy in China. Advanced Correspondence Transaction 2(2) (1997) 27-32

Improving Population Estimation with Neural Network Models

Zaiyong Tang[1], Caroline W. Leung[1], and Kallol Bagchi[2]

[1] College of Administration and Business
Louisiana Tech University, Ruston, LA 71272, USA
[2] Dept. of Information & Decision Sciences
University of Texas at El Paso, TX, USA

Abstract. Intercensal and postcensal population estimates are essential in federal, state, and local governments planning and resource allocation. Traditionally, linear regression based models are widely used for projecting population distributions in a given region. We constructed population projection models with various types of artificial neural networks. Using historical census data, we tested the performance of the neural network models against the ratio correlation regression model that we have used for the last 20 years. The results indicate that properly trained neural networks outperform the regression model in both model fitting and projection. Among the different neural network models we tested, the fuzzy logic based neural network performed the best.

1 Introduction

Population estimation plays a vital role in government policy and budgeting. In the United States, the Census Bureau conducts nationwide census every ten years. Many different approaches have been used by demographers to conduct local population estimation. Smith and Mandell [9] compared the Housing Unit Methods with Component II, Ratio Correlation, and Administrative Records. Swanson and Tedrow [10] used regression models with improved measurement of temporal change. Each method has its strengths and weaknesses. Among available methods, the Cohort-component method is data and calculation intensive, and develops separate mortality, fertility and migration modules. Due to its simplicity, the trend extrapolation method has low cost, is timely, and has small data requirements. Trend extrapolation's main failing is that it does not account for the components of growth. Given this, it does not describe the characteristics of the projected population [8].

Artificial Neural Networks have been successfully applied to prediction and forecasting in various disciplines [1]. However, to the best of our knowledge, neural network models have not been used in population projection. The goal of this article is to explore Artificial Neural Networks (ANN) as an alternative approach to traditional methods in population projection. We are interested in tapping into the computing power and nonlinear characteristic of neural networks. The key research question is whether ANN models can pick up seasonal changes and recognize patterns of fluctuation that has been elusive to traditional linear regression models.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1181 – 1186, 2006.

2 Neural Networks

Artificial neural networks are emerging computational models that have sparked wide interest in recent years. In contrast to conventional computer processing, which is centralized and sequential, an ANN consists of massively connected simple processing units, which are analogous to the neurons in the biological brain. Through elementary local interactions among these simple processing units, sophisticated global behaviors emerge.

Many neural network paradigms have been developed during the last two decades. One of the most widely used neural network models is the feedforward neural network, where neurons are arranged in layers [6]. Besides an input layer and an output layer, there are one or more hidden layers between the input and the output layer. N×H×O is commonly used to denote a neural network with N input nodes, H hidden nodes, and O output notes.

Neural networks as general nonlinear models have remarkable ability of learning the underlying relationship from data. Before a neural network is used, it is first trained with known examples. Training a neural network means modifying the weights on the links such that the network can accurately map an input pattern to a target output pattern. A widely used training algorithm for feedforward neural networks is the well known backpropagation algorithm. Radial basis function (RBF) networks use Gaussian transfer functions, rather than the standard sigmoidal functions at the hidden layer. These networks tend to learn much faster than feedforward networks. Fuzzy inference systems built within neural networks can approximate complex functions [3],[12]. For a complete coverage of the backpropagation training algorithm and many of its variations, the reader is referred to Rumelhart and McClelland[6], van der Smagt [7], and Buckley and Hayashi [2]. Pirincipe et al. [5] provides both theoretic coverage and application examples of various neural network models.

3 Traditional Population Estimate Models

In general, total Population consists of the combination of Household population and Group Quarters population (those living in dormitories, military installations, jails, etc). The Group Quarters population is recorded by each Group Quarters facility itself and demographers collect the population counts directly from the facilities. Due to the tremendous time and expense of conducting an actual physical count, Household population counts are not collected yearly. Instead, population estimation methods are developed for estimating the Household population component of the total population.

Ratio Correlation Regression Method is the population estimation methods we have been using at our Research Division. Ratio Correlation Method and its variations have been among those most widely used methodologies [4]. Formally, it is a multiple regression-based model which estimates changes in the household population as a result of changes in selected independent variables. Those independent variables are symptomatic indicators to estimate the “Change” in the proportionate share of the population between the base year and the estimate date. The coefficients of the model are calculated based upon the observed relationship of changes in the independent variables between two Censuses.

4 Empirical Test Results

4.1 Data Description

We have developed a regression model to estimate the household population of each of 64 parishes (counties) in the State of Louisiana, USA. The three independent variables are: Births, Deaths and School Enrollment. Data for the first 2 variables are collected from Office of Public Health, Louisiana Department of Health & Hospitals. Obtain each parish's last 3 years "Births" and "Deaths" before 2 continuous Censuses. Then calculate the change of parish's proportionate share of the "Births" and "Deaths" of state total from the first Census to the second Census. Data for the "School Enrollment" variable are collected from State Department of Education. It is the last single year school enrollment in grades 1-8 (including special education) before the 2 Censuses. Then we calculate the change of parish's proportionate share of the state total "School Enrollment" from the first Census to the second Census. Our regression model for parish population estimation is based on the "component change" method. For a detailed explanation of the data and the model, the reader is referred to the document Methodology for Estimates of State and County Total Population [11], available at the US Census website.

4.2 The Regression Model

In order to build the model and calculate the coefficients, we used the changes of parish household population proportionate share of the state total from Census 1980 to Census 1990 as the observed dependent variable. The variables are described below:

HP = "Ratios of per parish's share of 1990 Census state total household population" divided by "Ratios of per parish's share of 1980 Census state total household population"
B = "Ratios of per parish's share of 1988-1990 state total of births" divided by "Ratios of per parish's share of 1978-1980 state total of births"
D = "Ratios of per parish's share of 1988-1990 state total of deaths" divided by "Ratios of per parish's share of 1978-1980 state total of deaths"
SE = "Ratios of per parish's share of 1990 state total of school enrollment" divided by "Ratios of per parish's share of 1980 state total of school enrollment"

Ordinary least square linear regression results in the following model with r square = 0.8254.

$$HP = 0.08553 + 0.3542\,B + 0.26212\,D + 0.3037\,SE$$

Using coefficients from this model, we estimate the changes of parish household population proportionate share of the state total for year 2000. Multiplying this expected HP_{2000} by original proportionate share of the state total in the base year will obtain ratio of proportionate share of the state total for each parish in year 2000. Parish household population is the product of the parish share ratio and the state total. Group Quarters population which was collected individually from each facility is then added to the household population to obtain the parish total population.

4.3 The Neural Network Models

Although simple feedforward neural networks with back-propagation training algorithm have been widely used, their limitations such as lengthy training time, sensitivity to initial weights, and the likelihood of getting stuck in local minima, are well documented [5]. We have decided to test other neural network models such as radio base function and fuzzy logic neural network models. The fuzzy sets are encoded as connection strengths (weights) [2]. The Fuzzy neural network model integrates adaptable fuzzy inputs with a modular neural network to rapidly and accurately approximate complex functions. One of the advantages of the fuzzy neural networks is that they combine the explanatory nature of rules (fuzzy membership functions) with the power of feedforward neural networks, which are often referred as "Black boxes."

The neural network model uses the same number of independent variables as the regression model. The number of hidden layers and number of hidden nodes in each hidden layer are typically dictated by the complexity of the problem. For this initial investigation, we use a simple neural network model with one hidden layer. The number of hidden nodes is typically chosen between 1 to 2 times the numbers of input nodes for feedforward networks (FN). We varied the number between 4 and 6. Since the results do not show significant difference with different number of hidden nodes, we report only the results from a 3x4x1 structure for FN and generalized feedforward networks (GFN). GFNs are feedforward networks with direct connections from the input nodes to the output node. The radial base function (RBF) model has a structure of 3x15x1. The fuzzy logic neural network model has a structure of 3x54x1 due to the construction of fuzzy member functions in the hidden layer.

Each model is trained 1000 iterations using the 1990 over 1980 ratio data. The models are then used to estimate the 2000 population proportions in the 64 counties. The estimates are then compared with actual year 2000 census data. Table 1 lists the mean sum-of-square error (MSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) of the linear regression model and various neural network models. For neural network models, the results are average of 10 runs starting with random initial weights.

Table 1. Estimate Error Comparison of Neural Network Models vs. Linear Regression

Model	MSE	MAE	MAPE
NN-MLP	5.94288E-07	0.000415	0.032863
NN-GFN	4.76612E-07	0.000406	0.033258
NN-RBF	5.71349E-07	0.000398	0.033271
NN-Fuzzy	5.01001E-07	0.000380033	0.031469
Linear Regression	5.39215E-07	0.000418	0.033632

Among the error measures, MAE is the most important as it can be converted to number of people over/under estimated. Although feedforward neural networks (the most widely used) show improvement over the regression model, the best model for this task seems to be fuzzy logic neural networks. Table 2 shows the minimum,

Table 2. NN-Fuzzy Improvement over Regression (in parenthesis)

	MSE	MAE	MAPE
Minimum	4.53E-07 (15.96%)	0.000358036 (14.35)	0.02962 (11.93%)
Maximum	4.766E-07(-2.89%)	0.000406 (3.83%)	0.03320 (1.28%)
Average	5.01E-07 (7.13%)	0.000380033 (9.08%)	0.03147 (6.49%)

maximum, and average estimation errors of the NN-Fuzzy model. Using the regression result as the base, out of the 10 runs, the MAE is reduced from 3.83 percent to 14.35 percent. The average reduction is 9.08 percent.

5 Conclusions

Common estimation methods, like Component method, Housing Unit method, and Ratio Correlation method have their shortcomings. In practice, if possible, demographers will use 2 or 3 different estimation methods, and then assign certain weights to those independent methods to obtain an average. The decision of choosing estimate methods depends on the data availability and reliability of the data in each geographic level. As an example, Births and Deaths data may not be available in sub-county level.

Artificial neural networks have been used successfully as general non-linear models for a wide variety of pattern recognition and forecasting problems. Our preliminary results of applying neural networks for population estimation indicate that this type of models can outperform conventional models. The significant results of this study foretell the use of neural networks in new directions for population projection. The true power of neural networks lies in their ability to represent both linear and non-linear relationships, as compared with traditional regression models which represent only linear relationships among the variables. There is great potential in the ability of neural network models to pick up the non-linear population changes such as the seasonal and unexpected fluctuations. Further research can be conducted with a variety of other neural network models. It leaves a wide open territory for further studies in the application of neural networks for population projection and estimation.

References

1. Bishop, C.M.: Neural Networks for Pattern Recognition. Clarendon, UK (1995)
2. Buckley, J.J. and Hayashi, Y.: Fuzzy Neural Networks: A Survey, Fuzzy Sets and Systems 66 (1994) 1-13
3. Kosko, B.: Neural Networks and Fuzzy Systems. Prentice Hall, Boston (1992)
4. Namboodiri, N.K.: On the Ratio-Correlation and Related Methods of Subnational Population Estimation. Demography 9 (1972) 443-453
5. Principe, J.C., Euliano, N.R., and Lefebvre, W.C.: Neural and Adaptive Systems: Fundamentals through Simulations, John Wiley and Sons, Inc., New York (2000)
6. Rumelhart, D.E., McClelland, J.L., and the PDP Research Group: Parallel Distributed Processing - Explorations in the Microstructure of Cognition, Volume I: Foundations, The MIT Press, Boston (1986)

7. van der Smagt, P.P.: Minimization Methods for Training Feedforward Neural Networks. Neural Networks 7 (1994) 1-11
8. Smith, S.K. and Mandell, M.: A Comparison of Local Population Estimates: the Housing Unit Method vs. Component II, Ratio Correlation and Administrative Records. Journal of the American Statistical Association 79(386) (1984) 282-289
9. Smith, S. K., Tayman, J., Swanson, D.A.: State and Local Population Projections Methodology and Analysis. Kluwer/Plenum, New York (2001)
10. Swanson, D.A. and Tedrow, L.M.: Improving the Measurement of Temporal Change in Regression Models Used for County Population Estimates. Demography 21(3) (1984) 373-381
11. US Bureau of the Census: Methodology for Estimates of State and County Total Population. Available at http://www.census.gov/population/methods/stco99.txt. Revised April, 2000
12. Zadeh, L.A.: Fuzzy Logic, Neural Networks, and Soft Computing. Communications of the ACM 37(3) (1994) 77-84

Application of Fuzzy Neural Network for Real Estate Prediction

Jian-Guo Liu[1], Xiao-Li Zhang[1], and Wei-Ping Wu[2]

[1] Department of Computer Science, Chongqing Technology and Business University, 400067, Chongqing, China
{ljg, jsjzxl}@ctbu.edu.cn
[2] Department of Foreign Languages, Chongqing Technology and Business University, 400067, Chongqing, China
wuweiping@ctbu.edu.cn

Abstract. A FNN prediction model based on hedonic price theory to estimate the appropriate price level for a new real estate is proposed. The model includes a database storing hedonic characteristics and coefficients affecting the real estate price level from recently sold projects that are representative in the local environment. The experimental result shows that the fuzzy neural network prediction model has strong function approximation ability and is suitable for real estate price prediction depending on the quality of the available data.

1 Introduction

The profession of real estate valuers arises because each real estate asset is different from all other real estates. Real estate assets are heterogeneous, that is, their characteristics vary. Recently, hedonic pricing models have also been used to complete the sales comparison approach. This grounds real estate prediction more firmly in modern economics and finance theory and artificial intelligence methods [1]. Fuzzy Neural Network (FNN) is a connecting link between fuzzy logic and neural computing. They are generally applied in electrical engineering and automatic control area to simulate the problem solving process of human brain and assist people in making decisions under complex situations [2]. So fuzzy neural network prediction model based on hedonic price theory was highly appropriate to the tasks of real estate prediction and decision-making. Moreover, adaptive neuro-fuzzy inference systems (ANFIS) provided the means by which the judgments could be formalized without application of an artificial process to make the judgments precise [3]. The main objectives of this paper are to develop a fuzzy neural network price prediction model, which can learn from historical data on the correlations between various factors that influence the prices of real estates and the actual selling prices, so that the model can be used to estimate the appropriate price level for a real estate. This chapter attempts to present fuzzy neural network concepts and issues that are associated with application to prediction of market prices of real estate. The paper is organized as follows. In Section 2, the survey of fuzzy neural network is discussed. Hedonic Price Model is proposed in Section 3. In Section 5, the model of predicting market prices of real estate, it includes data collection and analysis, model construction, conceptual model and error analysis. Section 6 concludes the paper.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1187 – 1191, 2006.

2 Fuzzy Neural Networks

Fuzzy neural network is a concept that integrates some features of the fuzzy logic and the Artificial Neural Networks (ANN) theory. It based on the integration of two complementary theories. Purpose of the integration is to compensate weaknesses of one theory with advantages of the other [4]. Fuzzy logic and neural networks are complementary technologies. ANN extracts information from systems to be learned or controlled, while fuzzy logic techniques most often use verbal and linguistic information from experts. A promising approach to obtain the benefits of both fuzzy systems and ANN and solve their respective problems is to combine them into an integrated system. FNN techniques have several features that make them well suited to a wide range of knowledge engineering applications. These strengths include fast and accurate learning, good generalization capabilities, excellent explanation facilities in the form of semantically meaningful fuzzy rules, and the ability to accommodate both data and existing expert knowledge about the problem under consideration.

3 Hedonic Price Model

Real estate assets are heterogeneous, that is, their characteristics vary. Researchers and practitioners have found that hundreds of factors might affect prices in various situations. Interaction effects and non-linear relationships between prices and hedonic variables complicate the issues. So people interested in prices of particular real estate assets consult valuers who collect and interpret recent sales evidence in order to arrive at a price estimate based on interpretation of differences between real estates [5].

The hedonic price models where price as a function of hedonic characteristics is estimated by multivariate regression represent the insight that people buy a bundle of characteristics of real estates, not a simple, one-dimensional source of utility. Many factors influence the prices people pay for real estate. These models are written as:

Price=coefficients*characteristics +differences.

The coefficients are weights, dollars per unit of characteristic. The characteristics are features of the real estate that have an effect on utility to buyers. Differences mean the price implications, positive or negative, of the differences in hedonic characteristics between the real estates [6]. Usually a large but not very homogeneous sample of sales prices is used to estimate best fitting coefficients in a hedonic price model. The equation is $Ps=\sum b_i X_i$ where X_i are real estate characteristics like size, age, etc. And b_i are coefficients or weights. Normally these coefficients would differ because price is not a linear function of most real estate characteristics.

4 The Application of Fuzzy Neural Network Prediction Model

4.1 Data Collection and Analysis

The main objective of this section is to find out any correlation between various factors and real estate prices for given project characteristics in the local environment. A database storing hedonic characteristics and coefficients was established from

recently sold projects that are representative in the local environment, and A knowledge base in the form of fuzzy rules, neural networks, or regression models that relate factors affecting real estate prices to recommended pricing structures was constructed. The methodology was based on a questionnaire survey to establish the correlations among the variables, supplemented by case study of selected projects using interviews. The following explains types of data needed for the proposed research.

The data to be collected consists of a sample of recently completed building projects of various categories. Hedonic characteristics and coefficients of each case project are grouped into four areas of five factors: Location, Design and appeal, Quality of construction, Above-grade room count/gross living area, and surrounding conditions. The definitions of the above variables and the exact form and manner in which the data items are solicited, e.g. quantitative or qualitative, range and levels, direct or deduced, will be tailored in the design of the questionnaire. Follow-up interviews are used to supplement and clarify information.

4.2 Model Construction

For FNN prediction model, the structure adapted consists of four layers. The function of each layer is discussed below. Layer 1 reads real number input variables xi (i=1, 2, … 5). In the FNN prediction model of this study, layer l reads the evaluated value of each identified influencing factor for markup estimation. Layer 2 fuzzifies the input variables according to the membership functions. Every input value x_i x_i has 2 membership degree $\mu_{A_i^j}(x_i)$ (j=1, 2), which represent the characteristic of the influencing factor. Gaussian functions [7] is applied for all the membership functions, where

$$\mu_{A_i^j} = \exp\left(-\left(\frac{x_i^j - a_i^j}{\sqrt{2}b_i^j}\right)^2\right) \tag{1}$$

$\mu_{A_i^j}$ is the membership degree of x_i, a_i^j is the parameter determining the center value, b_i^j is the parameter determining the width of the membership function. The reason for using Gaussian membership function is because of the good characteristics of this function. The following calculation and training process adapted the Gaussian membership function, the denominator of the training functions will not be zero and fewer parameters need to be adjusted. Layer 3 calculates μ_k (k =1,2…32), the active degree of the *k*th rule according to the relevant fuzzy inference rules collected for markup estimation.

$$\mu_k = \mu_{A_1^j}(x_1)\mu_{A_2^j}(x_2)....\mu_{A_n^j}(x_5) \tag{2}$$

Where μ_k is the active degree of the *k*th rule and $\mu_{A_1^j}(x_i)$ is the membership degree of the *i*th input factor's *j*th characteristics. Layer 4 defuzzifies the final output M of such a neural fuzzy system with centroid defuzzification equation as follows:

$$M = \frac{\sum_{k=1}^{32} \mu_k w_k}{\sum_{k=1}^{32} \mu_k} \tag{3}$$

Where w_k is the markup percentage from the kth rule and M is the final estimated markup percentage.

4.3 Conceptual Model

The FNN prediction model has been implemented under Fuzzy Logic Toolbox in MATLAB. We've applied fuzzy inference to modeling systems whose rule structure is essentially predetermined by my interpretation of the characteristics of the variables in the model [8]. Upon user interaction, using a given training data set that contains desired input/output data pairs of the target system to be modeled, the fuzzy system was trained by adjusting the membership function parameters that best model this data. Then the toolbox function anfis constructs a fuzzy inference system (FIS) whose membership function parameters are adjusted using a backpropagation algorithm. The training will stop after the training data error remains within this tolerance. Model validation is the process by which the input vectors from input/output data sets are presented to the trained FIS model, to see how well the FIS model predicts the corresponding data set output values. When checking data is presented to anfis as well as training data, the FIS model is selected to have parameters associated with the minimum checking data model error.

4.4 Error Analysis

The trained FIS model was verified with new real estate data. The experiment indicates favorable prediction results with the actual real estate prices of detached houses. As the table 1 shows, random Minimum and maximum errors are 1% and 11% respectively, while median error is 5% and 6%.

Table 1. Results of FNN predictive model

No.	FNN Model Output	Actual Prices	Value Difference	Error
1	1947	2100	-153	7.86
2	2193	2030	163	7.43
3	2538	2480	58	2.29
4	2071	2000	71	3.43
5	1776	1750	18	1.02
6	2205	2300	-95	4.31
7	1788	1680	108	6.04
8	2895	1100	-206	10.87
9	2334	2180	154	6.59
10	1807	1900	-93	5.15

While these kinds of calculations may seem tedious and overly time consuming they improve the prediction product by allowing the valuer to make representations supported by evidence regarding the accuracy of the value estimate provided.

Moreover, further automation could allow valuers to test empirically the validity of their price adjustment models. Different coefficients could be used to see which would minimize prediction errors. Optimum sample sizes could also be explored.

Therefore, from this perspective, FNN prediction model in this study gave a reasonable and acceptable value of real estate.

5 Conclusion

We developed a FNN prediction model to estimate the appropriate price level for a new real estate. The experimental result demonstrated that applications of artificial intelligence methods to real estate prediction and decision support allow getting significantly improved predicting results and outperforms the classic prediction methods. Although the results from this study seem appealing in enhancing prediction technique, the fuzzy rules could be fine-tuned in order to produce more accurate results. For instance, the FNN model can be improved for other application domain and can be enhanced to meet current market and user needs. Hence the future study will try to identify more variables to be added in FNN prediction model, i.e. economy and socio-economic factors.

References

1. Englund, P., Hwang, M., Quigley, J.: Hedging Housing Risk. Journal of Real Estate Finance and Economics 24(1) (2002) 167-200
2. Kuo, R.J., Xue, K.C.: Fuzzy Neural Networks with Application to Sales Forecasting. Fuzzy Sets and Systems 108 (4) (1999) 123-143
3. Bagnoli, C., Smith, H.C.: The Theory of Fuzzy Logic and It's Application to Real Estate Prediction. Journal of Real Estate Research 16(2) (1998) 169-199.
4. Lin, C.T., Lee, C.S.G. (eds.): Neural fuzzy systems: A Neural-fuzzy Synergism to Intelligent Systems. Prentice-Hall, Inc. USA(1996)
5. Wolverton, M.L., Senteza, J.: Hedonic Estimates of Regional Constant Quality House Prices. Journal of Real Estate Research 19(3) (2000) 235-253.
6. Rosen, S.: Hedonic prices and implicit markets: product differentiation in pure competition. Journal of Political Economy 1 (1974) 34-55.
7. Li, H., Shen, L.Y., Love, P.E.D.: ANN-Based Mark-up Estimation System with Self-explanatory Capacities. Journal of Construction Engineering and Management. 125(3) (1999) 185-189
8. Nguen, N., Cripps, A.: Predicting Housing Value: A Comparison of Multiple Regression Analysis and Artificial Neural Networks. Journal of Real Estate Research 22(3) (2001) 313-336

Local Neural Networks of Space-Time Predicting Modeling for Lattice Data in GIS

Haiqi Wang[1,2], Jinfeng Wang[1], and Xuhua Liu[3]

[1] Institute of Geographical Sciences and Nature Resources Research, Chinese Academy of Sciences, Beijing 100101, P.R. China
{wanghq, wangjf}@lreis.ac.cn
[2] College of Geo-resources and Information, University of Petroleum (East China), Dongying, Shandong 257061, P.R. China
[3] Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089-2564, U.S.A.
ruthlxh@gmail.com

Abstract. Lattice data have two different scale spatial properties: global dependence property and local fluctuation property. For lattice data space-time autoregressive modeling, to reduce influence of spatial fluctuation on prediction accuracy of neural networks, all regions are partitioned into several subareas by an improved k-means algorithm based on spatial contiguity relation. Some partition criteria are proposed to evaluate different partition schemes and the optimal scheme has the least spatial fluctuation and significant spatial dependent within each subarea. Each multi-layer perceptrons (MLPs) network is modeled respectively for each subarea, and the output nodes are the prediction values at time t of an attribute for all regions in a subarea, and the input nodes are observations before time t of this subarea itself and neighboring regions. As a case study, all local models are tested and compared with a single global MLPs network by one-step-ahead predicting of an epidemic dataset, and the results indicate that local NN model has better prediction performance than the global NN model.

1 Introduction

Three main types of geographic data in GIS defined by Cressie [1] include point pattern data, geostatistical data and lattice data. Lattice data refer to attributes attached to fixed, regular or irregular, polygonal regions such as districts or census zones in two-dimensional space. The main emphasis with lattice data space-time analysis is on detecting, modeling and predicting space-time patterns or trends of lattice attributes changed with time, while spatial topological structures, such as spatial arrangements and polygonal shapes and sizes, are simultaneously kept invariable.

From the perspective of space, just as the first law of geography referred to by Tobler [2], geographic objects take on interdependent patterns over space and furthermore, the degree of dependence is weakened with increasing of the distance between objects. In the context of lattice data, this dependence or association is referred to as spatial autocorrelation and it quantifies the correlation between the same attribute

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1192 – 1201, 2006.

variable at different spatial locations. At the same time, it has long been recognized, during the analysis of spatial dependence, that the assumption of structure stability over space may be highly unrealistic, especially when a large number of spatial observations are used. This fluctuation or rough, coexisted with overall spatial dependence, is represented in the form of local spatial clustering of similar values or local spatial outliers. Global spatial autocorrelation and local spatial rough of lattice data can be detected respectively by global and local version of Moran's *I* statistic.

This paper focuses on local neural networks space-time autoregressive modeling for lattice data. At First, to reduce the impact of local spatial fluctuation on predicting, all regions of the study area are partitioned by one modified k-means clustering algorithm based on spatial contiguity relation. Different partition schemes are evaluated and compared through a group of partition criteria, and among these criteria, global Moran's *I* quantifies global spatial autocorrelation and its significance for each subarea of regions, and the relationship between global and local Moran's *I* is utilized as an indicator of spatial fluctuation in each subarea. Second, each multi-layer perceptrons (MLPs) network is used respectively in modeling and predicting for each subarea. The output nodes are the predicting values at time *t* of an attribute variable for all regions in a subarea. The input nodes are observations before time *t* of the same attribute for the subarea itself and its neighboring regions. Finally, as a case study, all local models for all the subareas are trained, tested and compared with a single global MLPs network by modeling one-step-ahead prediction of an epidemic dataset which records weekly influenza cases of 94 departments in France from the first week of 1990 to the 53th of 1992.

2 Partition Criteria

The aim of partition, similar to regionalization that belongs to geographic terminology, is to group all initial regions into several classes using attributes in a way which is optimal according to one or more objective criteria [3]. In this research, three criteria are proposed to evaluate and select an optimal partition which serves local neutral networks modeling:

Dependence. If regions of a partitioned subarea have no spatial dependence with each other, it is inappropriate for NN modeling as output layer values of neural networks depend on the interactions of input layer nodes through hidden layers nodes. So an optimal partition must guarantee that there are real and significant spatial dependence among regions in a subarea.

Spatial autocorrelation of each subarea, as a single study area, can be measured by the global Moran's *I* and a test for significance of spatial autocorrelation can be based on *z*-score of Moran's *I* [4]. If Moran's *I* is nonzero and significant (for example significance level = 0.01), spatial dependence in a subarea exists. This criterion is an essential condition and the partition schemes unsatisfied with it must be discarded.

Contiuity. It means that only neighboring regions can be grouped into a subarea, and this criterion is also an essential condition and has been considered in the improved k-means algorithm in section 3.

Fluctuation. Although it is impossible to make each subarea have complete spatial stability through partitioning, the less fluctuation means the better predicting results of NN model [5].

Local fluctuation of a region i can be measured by local Moran's I_i [6]. If spatial weight matrix W, used in global and local Moran's I, is defined as contiguous form in which w_{ij} is one when region i has common borders with regions j and otherwise w_{ij} is zero, and furthermore W has been row standardized in which the sum of each row is one, the average of local Moran's I of all regions will be equal to global I. This relationship shows that if the underlying spatial process is stable throughout the whole area, then one would expect the local Moran's I to show little variation around their average, the global statistic. In other words, local statistic I with much difference from global Moran's I would indicate that the underlying process is highly fluctuant [6].

So the standard deviation between local Moran's I and global Moran's I within a study area can be defined as

$$\mathrm{Std}(I) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\text{local Moran's } I_i - \text{global Moran's } I)^2}\ , \tag{1}$$

where n is the number of regions in the study area. Here, Std(I) is regarded as an evaluation index to the fluctuation in one subarea. If Std(I) is relatively small, it is relatively stable and less fluctuant within a subarea of regions, and otherwise it is more fluctuant.

Another point that must be emphasized is that the modeling object of local NN is lattice space-time data, but Moran's I statistic involved in partition criteria above is only a pure spatial dependence index. So it is not appropriate for applying spatial Moran's I to measure space-time dependence. However, a space-time statistic, which can quantify global and local space-time autocorrelation of lattice data and also test significance of this dependence, is still not found in relevant literature. Thus a compromise method is proposed that, to each region, a spatial attribute variable is constructed and this synthetic variable is the function of different time-section observations of an attribute variable of this region. For example, to weekly flu cases of one county, the synthetic attribute variable of this county may be either the average, total, or maximum of weekly flu cases. The synthetic attribute is used to calculate Moran's I statistics in partition criteria.

3 K-Means Algorithm Based on Spatial Contiguity Relation

When the traditional clustering algorithms including k-means algorithm are used to partition spatial regions, these algorithms only utilize attributes and have not included spatial relations of regions. Then the clustering results can not guarantee that regions in one subarea (or one cluster) form a contiguous area on the map, and the map often shows that a single region of a subarea is surrounded by regions of another subarea [7]. Therefore, k-means algorithm is modified with consideration on spatial contiguity relations among regions. During partitioning, when judging one region which subarea it belongs to, not only should the distance be considered to the centroid of a subarea but also the common borders between this region and the regions in a subarea. Only when a region has neighbors in a subarea and has the shortest distance to this subarea, can it belong to this subarea.

3.1 Relevant Definitions

Lattice Object. Let the set of regions in the study area S have N spatial regions, $S=\{s_1, s_2,\ldots, s_N\}$ with a neighbor relation $R\subseteq S\times S$. Region s_i and s_j are neighbors if and only if $(s_i, s_j)\in R$, $i\neq j$. The neighbor relation R is given by a spatial contiguity matrix W, where $W(i, j)=W_{ij}=1$ if and only if $(s_i, s_j)\in R$ and otherwise, $W(i, j)=W_{ij}=0$ [8].

Associated with each s_i, there is a time-section observations vector of the studied attribute variable, $X_i\equiv X(s_i)=[x_{i1}, x_{i2}, \ldots, x_{iT}]$ and T is the length of time-section.

To each region s_i, a synthetic attribute Q is constructed and it is the function of X_i, $Q_i\equiv Q(s_i)=f(X_i)=f(x_{i1}, x_{i2}, \ldots, x_{iT})$. Q is used to calculate spatial Moran's I statistics of each subarea partitioned.

K-means Algorithm. Let the centroids of K subareas be $\{z_1, z_2,\ldots, z_K\}$ and each subarea z_j, like each x_i, be a T-D vector $z_j=[z_{j1}, z_{j2}, \ldots, z_{jT}]$ ($j=1, 2, \ldots, K$).

Define a set Z_j to each subarea z_j and Z_j contains spatial regions of the subarea z_j. During partitioning, the set Z_j is initially empty.

Define a 2-D distance matrix of $N\times K$, *Dist*. Its element is the distance from one region to the center of one subarea. In addition, define an $N\times K$ auxiliary logic matrix *DistMark*. An element of the logic matrix identifies whether its corresponding distance in the matrix *Dist* is participated in the searching process, which searches for the shortest distance from a region to a subarea in the matrix *Dist*. If an element value of *DistMark* is True, then the corresponding distance in *Dist* should be included in the search, otherwise excluded from the search.

3.2 Algorithm Procedure

Step 1: Utilize GIS software (for instance ArcGIS, GeoDa, etc.) to construct spatial contiguity matrix W [9], and give the partitioning number, K.

Step 2: Choose K spatial regions randomly from N regions as the initial centroids of K subareas, $\{z_1^{(0)}, z_2^{(0)},\ldots, z_K^{(0)}\}$.

Step 3: During iteration m, each set Z_j of K subareas is initially empty and the set of regions S is initialized to contain all the regions, $S=\{s_1, s_2, \ldots, s_N\}$. Calculate the distance, $Dist(s_i, z_j)$, between each region s_i and each subarea z_j according to the following equation (2) for all $i=1, 2, \ldots, N$ and $j=1, 2, \ldots, K$. Assign True to all the elements of the matrix *DistMark*.

$$Dist(s_i, z_j^{(m)}) \equiv Dist(i, j) = \left\| X_i - z_j^{(m)} \right\|^2 = \sum_{t=1}^{T} (x_{it} - z_{jt}^{(m)})^2 . \tag{2}$$

Step 4: Search the matrix *Dist* for the shortest distance from each region of the set S to each subarea. The corresponding element values in *DistMark* to the distances searched must be True. Let the shortest distance be *minDist*, and the region of *minDist* be s_{min_i} and the subarea be z_{min_j}.

Step 5: If the set Z_{min_j} of the subarea z_{min_j} is empty or the elements of Z_{min_j} have neighboring relation with the region s_{min_i}, that is $W(s_{min\_i}, Z_{min\_j})=1$, the set S and Z_{min_j} can be modified as $S=S-\{s_{min\_i}\}$, $Z_{min\_j}=Z_{min\_j}\cup\{s_{min\_i}\}$. Assign True to all the elements of *DistMark* and go to the next step;

If the elements of Z_{min_j} have no neighboring relation with the region s_{min_i}, set $DistMark(s_{min\_i}, z_{min\_j})$=False that indicates this distance will be excluded from the next search. Go back to **Step 4**.

Step 6: If the set S is not empty, go back to **Step 4**; otherwise, indicates all regions have already belonged to K subareas respectively and continues the next step.

Step 7: Update the centroids of K subareas $\{z_1^{(m+1)}, z_2^{(m+1)},\ldots, z_K^{(m+1)}\}$ by the following equation

$$z_j^{(m+1)} = \frac{1}{C(Z_j)} \sum_{s_i \in Z_j} X_i = \frac{1}{C(Z_j)} \sum_{s_i \in Z_j} (x_{i1}, x_{i2}, \ldots, x_{iT}) , \tag{3}$$

where j=1, 2,…, K and $C(Z_j)$ is the number of regions in the set Z_j at iteration m.

Step 8: If the algorithm has converged, $z_j(m)=z_j(m+1)$, terminate the iterations and each set Z_j contains regions of each subarea; otherwise set $m=m+1$ and go back to **step 3**.

Using this algorithm and giving different K values, we can get different partition schemes, and further an optimal scheme can be chosen through partition criteria.

At first, to the partition scheme of a certain K, global Moran's I of the attribute Q will be calculated respectively for each subarea in this scheme. If there are subareas with no autocorrelation or insignificant spatial autocorrelation, the scheme of this K will be discarded. Second, to all partition schemes with significant dependence, the fluctuation index Std(I) will be calculated respectively for each subarea in each scheme. The fluctuation of a partition scheme will be measured by the average of Std(I) of all subareas in this scheme. Therefore, an optimal scheme has significant spatial dependence and the least spatial fluctuation.

4 Neural Networks Modeling

4.1 Neural Networks Model

Based on the final partition scheme, we establish each NN model for each subarea. The multi-layer perceptrons (MLPs) networks are used in a variety of problems especially in predicting because of their inherent capability of arbitrary input-output mapping [10], so we focus on the MLPs model.

To a space-time predicting problem of lattice data, the inputs of a MLPs network are typically the past observations of an attribute variable of all regions and the outputs are the future values of this attribute variable of all regions. In fact, the MLPs network performs the following function mapping

$$X_t = f(X_{t-1}, X_{t-2}, \ldots, X_{t-p}) , \tag{4}$$

where X_t is the attribute vector of N regions at time t, $X_t=[x_{t1}, x_{t2}, \ldots, x_{tN}]$. Thus the MLPs network is equivalent to a space-time nonlinear autoregressive model. The parameter p is the size of moving window along time, and it decides the number of time-lagged observation vectors as the inputs.

The total input-output patterns of lattice data are divided into a training set and a test set, and the test set is used for measuring the prediction ability of the network. Two performance measures used are the average relative variance (ARV) and the dynamic

similarity rate (DSR). With the assumptions that the desired output at time t of N regions is a vector $X_t=[x_{t1}, x_{t2}, \ldots, x_{tN}]$ and that the model actual output is $\hat{X}_t=[\hat{x}_{t1},\hat{x}_{t2},\ldots,\hat{x}_{tN}]$, the DSR is defines as

$$\mathrm{DSR}=\sum_{i=1}^{N}(\hat{x}_{ti}-\overline{\hat{X}}_t)^2 \Big/ \sum_{i=1}^{N}(x_{ti}-\overline{X}_t)^2 , \tag{5}$$

where $\overline{\hat{X}}_t$ is the average of $\hat{X}_t$. Equation (5) is similar to the coefficient of determination in spatial interaction modeling [11]. DSR=0 means that the prediction is equal to the mean value. DSR=1 means that the dynamic behavior between the real vector and the predicted vector is completely identical but does not mean the two vectors are equal. In brief, the ARV measures the prediction accuracy and the DSR measures the prediction trend.

4.2 Boundary Effect of Local NN Model

To local MLPs model of each subarea for one-step-ahead predicting, the number of output nodes is relatively easy to specify. Corresponding to one region in a subarea is an output node in the output layer. Hence the number of output nodes is equal to the number of regions in a subarea.

The number of input nodes should be considered from both space and time. From the view of time, the number of input nodes is related to time-lagged observations of relevant regions and this is the parameter p mentioned above. From the view of space, the number of input nodes is not only related to the number of regions in a subarea but also related to the regions neighboring to a subarea and this can be called as boundary effect.

When all the regions are partitioned for local NN modeling, it does not mean that there is not any relation among regions of different subareas. There may be other forms of relationship, related to economy, traffic, population, etc., which can not be measured only based on spatial contiguity matrix. Thus, if we just consider the influence of the regions in a subarea on the outputs of local NN model and ignore the influencing factors of the regions surrounding a subarea, the relationship of different regions will be separated artificially, and it will violate the first theory of geography, and furthermore the predicting results of NN model will not be reliable. Therefore, when the NN model is established to predict results at time t of all the regions in a subarea, the inputs also include the observations before time t of the neighboring regions to this subarea besides the regions in this subarea.

If the number of regions in a subarea is n, and the number of neighboring regions is m, and the size of moving window is p, then the number of input nodes of local MLPs model is $(n+m)\times p$ and the number of output nodes is n.

5 Case Study

The sample lattice data set records weekly flu patients of 94 counties in France from the first week of 1990 to the 53th week of 1992, and the total period is 157 weeks. Number IDs of 94 France counties are shown in Fig. 1(a).

For each county, one synthetic attribute Q is constructed to be the average of weekly flu cases during 157 weeks. The spatial pattern of Q is displayed in the quantile map in Fig. 1(b), with the darkest shade corresponding to the highest quantile. To the variable Q, a positive and significant (significant level=0.05) global Moran's I of 0.128 indicates that average weekly cases of 94 counties are not random distribution but represent overall spatial dependence pattern. This statistic is computed based on a spatial contiguity (common border) matrix W.

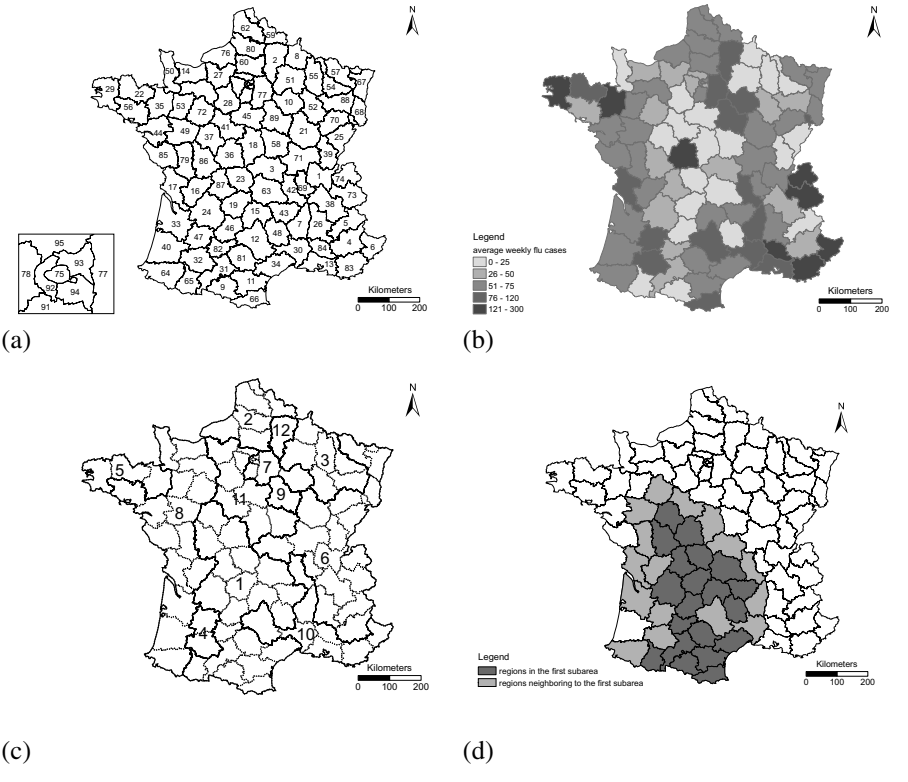


Fig. 1. (a) Number IDs of 94 counties in France; (b) Average weekly flu cases of 94 counties from the 1th week of 1990 to the 53th week of 1992; (c) The partition map of K=12 for 94 counties; (d) The first subarea and its neighboring regions

5.1 Partitioning

We partition 94 counties based on spatial contiguity matrix W and weekly flu cases of 157 weeks. Because the partitioning number K can not be predefined, we take sequentially an integer from 4 to 16 as K value of k-means clustering algorithm and get thirteen partition schemes of different K values. Through dependence calculating of partition criteria, K values with significant spatial dependence are 8, 9, 10, 12, 14, 16. Then the fluctuation index Std(I) is calculated respectively for these six partition schemes and listed in Table 1.

Table 1. Fluctuation index Std(*I*) of six partition schemes

K	Fluctuation index Std(*I*)	*K*	Fluctuation index Std(*I*)
8	0.6676	12	0.5251
9	0.6614	14	0.5884
10	0.6589	16	0.5345

Since the least value of Std(*I*) means the most stability, then the optimal partition scheme is *K*=12. The partition map of 12 subareas is displayed in Fig. 1(c). From the comparison of this figure with Fig. 1(b), the final partition scheme also reflects spatial distribution pattern of flu cases. For each subarea in the partition scheme of *K*=12, the number of regions, global Moran's *I* statistic, *z*-score of Moran's *I*, and the fluctuation index Std(*I*) are listed in Table 2. All the *z*-score values in Table 2, except that regions number is equal to 1, are either greater than 1.96 or smaller than -1.96, which indicates that each subarea in the final scheme has significant spatial dependence if significant level is 0.05.

Table 2. Relevant statistic of each subarea in the partition scheme of *K*=12

No.	Regions number	Global Moran's *I*	*z*-score	Std(*I*)	No.	Regions number	Global Moran's *I*	*z*-score	Std(*I*)
1	20	-0.339	-1.980	0.423	7	7	0.395	2.720	0.652
2	5	0.730	2.922	0.419	8	15	0.320	1.993	0.445
3	6	0.214	2.022	0.279	9	1			
4	2	-1	$-\infty$	0	10	8	0.533	2.149	0.666
5	3	-1.919	-4.014	0.342	11	8	0.407	2.216	0.403
6	18	0.251	1.973	0.678	12	1			

5.2 MLPs Modeling

Each MLPs model is established respectively for each subarea. For a subarea, the model utilizes flu cases at week *t*-1, *t*-2, …, *t*-*p* of all its counties and its neighboring counties to predict the cases at week *t* of all its counties. Because the infectious period of the influenza is about one week, the model input can only consider flu cases of previous week *t*-1, namely *p*=1. Hence, in a MLPs model, the number of input nodes is

Table 3. The number of input nodes and output nodes of each MLPs model for twelve subareas

No.	Input nodes	Output nodes	No.	Input nodes	Output nodes
1	36	20	7	17	7
2	9	5	8	28	15
3	15	6	9	6	1
4	10	2	10	22	8
5	8	3	11	24	8
6	32	18	12	7	1

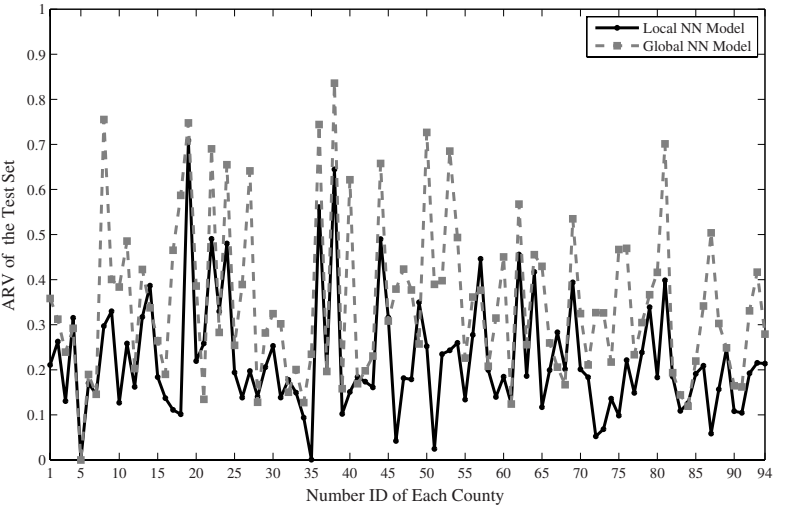


Fig. 2. The ARV of local and global NN model for 94 counties in France

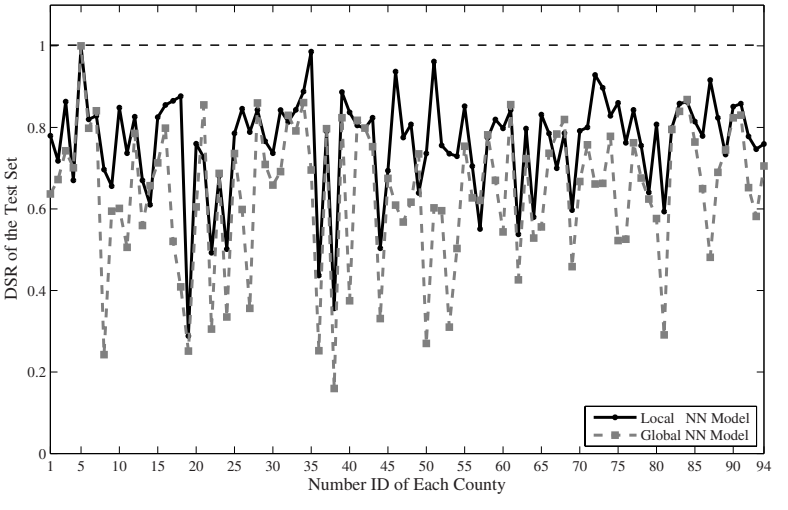


Fig. 3. The DSR of local and global NN model for 94 counties in France

equal to the number of regions in a subarea plus regions neighboring to this subarea. Fig. 1(d) illustrates the neighboring regions to the 1st subarea. Table 3 lists the number of input nodes and output nodes of each MLPs model.

In order to compare the performance of these local NN models, a global NN model for all 94 counties is simultaneously constructed. This model has 94 nodes in input and output layers, and the input nodes are the flu cases of week *t*-1 and the output are the predicting cases of week *t*.

Before the global and local models above are trained, 156 input-output pairs (X_{t-1}, X_t) are randomly divided into a training set and a test set in the proportion of 90% vs. 10%. The former has 140 pairs and the latter has 16 pairs. These NN models are trained through back-propagation algorithm, and the performance of each model can be quantified by ARV and DSR of the test set. Finally, two performance measures of local and global MLPs model for all 94 counties are shown in Fig. 2 and Fig. 3, which indicate that local NN model based on partitioning has better predicting capability than global NN model.

6 Discussions

From spatial properties of lattice data, an effective local model has been proposed for space-time predicting. Several issues are still worth further study. One involves the initial subareas of partitioning which are selected randomly in our study. A reasonable approach should combine the selection with spatial patterns, for instance considering the center of local clustering. Partition criteria should be another issue and different types of spatial and space-time processes, such as rainfall, price waves, public data, etc, may have different objective criteria for choosing an optimal partition. In addition, it may be more imperative to study feasible measures for quantifying global and local space-time dependence of lattice data and testing significance of this dependence.

References

1. Cressie, N.A.: Statistics for Spatial Data. 2nd edn. Wiley, New York (1993)
2. Tobler, W.: A Computer Movie Simulating Urban Growth in the Detroit Region. Economic Geography 46 (1970) 234–240
3. Haining, R., Wise, S., Ma, J.: Designing and Implementing Software for Spatial Statistical Analysis in a GIS Environment. Journal of Geographical Systems 2 (2000) 257–286
4. Anselin, L.: Spatial Econometrics: Methods and Models. Kluwer Academic Publishers, Dordrecht (1988)
5. Gilardi, N., Bengio, S.: Local Machine Learning Models for Spatial Data Analysis. Journal of Geographic Information and Decision Analysis 4(1) (2000) 11–28
6. Anselin, L.: Local Indicators of Spatial Association–LISA. Geographical Analysis 27(2) (1995) 93–115
7. Wise, S., Haining, R., Ma, J.: Regionalization Tools for the Exploratory Spatial Analysis of Health Data. In: Fisher, M., Getis, A. (eds.): Recent Developments in Spatial Analysis: Spatial Statistics, Behavioral Modeling, and Computational Intelligence. Springer-Verlag, Berlin Heidelberg New York (1997)
8. Hu, T.M., Sung, S.Y.: Data Fusion in Radial Basis Function Networks for Spatial Regression. Neural Processing Letters 21 (2005) 81–93
9. Anselin, L.: GeoDa 0.9 User's Guide. http://www.geoda.uiuc.edu (2003)
10. Zhang, G., Patuwo, B.E., Hu, M.Y.: Forecasting with Artificial Neural Networks: the State of the Art. International Journal of Forecasting 14 (1998) 35–62
11. Fotheringham, A.S., O'Kelly, M.E.: Spatial Interaction Models: Formulations and Applications. Kluwer Academic Publishers, Dordrecht (1989)

Modeling Meteorological Prediction Using Particle Swarm Optimization and Neural Network Ensemble

Jiansheng Wu[1], Long Jin[2], and Mingzhe Liu[3]

[1] Department of Mathematics and Computer, Liuzhou Teachers College, Guangxi, China
wjsh2002168@163.com
[2] Guangxi Research Institute of Meteorological Disaster Mitigation, Nanning, Guangxi, China
Jinlong01@163.com
[3] Institutes of Information Sciences and Technology, Massey University, Private Bag 11222, Palmerston North, New Zealand
m.z.liu@massey.ac.nz

Abstract. In this paper a novel optimization approach is presented. Network architecture and connection weights of neural networks (NN) are evolved by a particle swarm optimization (PSO) method, and then the appropriate network architecture and connection weights are fed into back-propagation (BP) networks. The ensemble strategy is carried out by simple averaging. The applied example is built with monthly mean rainfall of the whole area in Guangxi, China. The results show that the proposed approach can effectively improves convergence speed and generalization ability of NN.

1 Introduction

In the last few years, neural network approaches have been widely used in Meteorology discipline and climate analysis [1, 2]. As neural network approaches want of a rigorous theoretical support, effects of applications are strongly depend upon operator's experience. In the practical application, researchers determine appropriate network architecture and values of different parameters with trial and error due to short of prior knowledge. Therefore, applied effects of neural networks vary with operators. That is to say, even the same method is applied to solve the same problem, different operators can probably work out different results, which will cause over fitting, heavily degrade generalization ability of networks and limit applications of neural networks in meteorology study [3, 4].

Neural network ensemble is a learning paradigm where a number of neural networks are trained for the same task. It originate from Hansen and Salamon's work [5, 6], which shows that the generalization ability of a neural network system can be significantly improved through training many neural networks and then combining their results. This technology has been widely used in different fields such as face recognition [7], optical character recognition [8], signal classification [9], and medicine [10] and so on.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1202 – 1209, 2006.

Recently, a new evolutionary computation technique, the particle swarm optimization (PSO), is applied. Its development was based on observations of the social behavior of animals such as bird flocks, fish schooling, and swarm theory. Each individual in PSO is assigned with a randomized velocity according to its own and its companions' flying experience. The individuals, called particles, are then flown through hyperspace [11, 12].

In this paper, a novel optimization approach is proposed. PSO algorithm is applied to evolve neural network architecture and connection weights. The evolved neural network architecture and connection weights are input into a new neural network. The new neural network is trained using back-propagation (BP) algorithm. The output is obtained by simple averaging. The real date – monthly precipitation in Guangxi China is used to examine the proposed approach. Experimental results show that this approach is easy to operate and can lead to a high prediction accuracy.

2 Neural Network Ensemble Based on PSO

BP is a popular training algorithm. However, it has been proved that BP algorithm based on gradient descent is strongly dependent upon the choice of initial connection weights [13]. In addition, lots of issues are based on n-dimensional curve surfaces, which makes BP algorithm to converge slowly and fall in a local minima easily. Moreover, in the applications of neural networks, there still lacks of strict mathematical proof to determine the most appropriate network architecture. It normally is determined empirically, e.g., the numbers of neurons in the hidden layer are examined by increase or decrease progressively [14]. These deficiencies make outputs of training and test samples inconsistent and unpredictable, and limit applications of neural networks.

Position-speed relation model of PSO operates easily. The method of using PSO to evolve neural networks includes three steps: (i) using global searching ability of PSO to find an appropriate network architecture and connection weights; (ii) using BP algorithm to search peak value(s) in detail; (iii) obtaining results by simple averaging.

Mathematically, optimization problems of PSO-neural networks can be described as follows [15].

$$\begin{cases} \min\ E(w,v,\theta,\gamma) = (\mathbf{1}/N)\cdot\sum_{k=\mathbf{1}}^{N_{\mathbf{1}}}\sum_{t=\mathbf{1}}^{n}[y_k(t)-\hat{y}_k(t)]^{\mathbf{2}} < \varepsilon_{\mathbf{1}} \\ \quad \hat{y}_k(t) = \sum_{j=\mathbf{1}}^{p} v_{jk}\cdot f[\sum_{i=\mathbf{1}}^{m} x\cdot w_{ij} + \theta_j] + \gamma_t \\ \quad f(x) = \mathbf{1}/\exp(-x) \\ s.\ t \quad w\in R^{m\times p},\ v\in R^{p\times n},\ \theta\in R^{p},\ \gamma\in R^{n} \end{cases} \tag{1}$$

Where x is training samples, $\hat{y}_k(t)$, $y_k(t)$ are the desired output and real data, respectively. The fitness function is defined as follows:

$$F(w,v,\theta,r) = \mathbf{1}/(\mathbf{1} + \min\ E(w,v,\theta,r)) \tag{2}$$

Here we introduce our scheme:

Step 1: Initialize positions and speeds of a population. L individuals are randomly generated and each of them includes two parts: position and speed. The position of

each individual consists of network node link and connection weights. The hidden nodes are encoded as binary code string, 1 with connection and 0 without connection. The connection weights are encoded as float string, randomly generated within [-1, 1].
Step 2: Input training samples and calculate the fitness of each particle according to Expression (1). Initialize individual best position $p_{best}(t)$ and the global best position $p_{gbest}(t)$.
Step 3: Compare individual current fitness and the fitness of its experienced best position. If current fitness is better, we set current position to be the best position. Compare individual current fitness and the fitness of the global best position. If current fitness is better, we set current position to be the global best position.
Step 4: Equation of speed evolution for each particle can be written as follows:

$$v_{ij}(t+\mathbf{1}) = \omega \cdot v_{ij}(t) + c_1 r_1 (P_{best}(t) - x_{ij}(t)) + c_2 r_2 (P_{gbest}(t) - x_{ij}(t)) \tag{3}$$

$$\omega(t) = \omega_{\max} - ((\omega_{\max} - \omega_{\min}) / iter_{\max}) \cdot iter \tag{4}$$

Where ω_{max}, ω_{min} denote the maximum and minimum of inertia weights, respectively. While $iter$, $iter_{max}$ denote current iteration number and the maximum iteration number.
Step 5: According to Ref. [16], equation of network link can be written as follows:

$$x_{ij}(t+\mathbf{1}) = \begin{cases} \mathbf{0}, & r \geq \mathbf{1}/(\mathbf{1}+\exp(-v_{ij}(t+\mathbf{1}))) \\ \mathbf{1}, & r < \mathbf{1}/(\mathbf{1}+\exp(-v_{ij}(t+\mathbf{1}))) \end{cases} \tag{5}$$

Where r ranges from [0,1]. Equation of position evolution for each particle can be written as follows:

$$x_{ij}(t+\mathbf{1}) = x_{ij}(t) + v_{ij}(t+\mathbf{1}) \tag{6}$$

Step 6: Repeat step 2 ~ 5 until stopping criteria are satisfied, e.g., the best fitness is satisfied or the maximum iteration number is reached.
Step 7: Decode each particle and obtain L groups network architecture and connection weights. Thus, we can form L different neural networks. Train these networks with training samples until stopping criteria are satisfied.
Step 8: Input testing samples and output L results.
Step 9: Combine these L results with sample averaging.

The above-mentioned method can be summed up as follows: firstly, using PSO algorithm to get L groups network architecture and connection weights. Secondly, assigning these network architecture and connection weights into L neural networks and training them using BP algorithm. Finally, inputting test samples and averaging output results.

3 Data Pre-processing

In order to increase the accuracy of the method, pre-processing should be implemented in advance. There are many methods to pre-process data, such as data cleaning (cleaning incomplete, noisy, or inconsistent data), data integration (integrating different data sources), data normalization (scaling data value in a range, such as

[0, 1]), data reduction (reducing huge dataset to smaller representation), etc. This paper adopted data cleaning to pre-process meteorological data. Particularly, we use singular spectrum analysis (SSA) method [17] to reconstruct original precipitation series, and then use mean generating function (MGF) [18] to construct mean generating function matrix. This matrix is denoted as a self variable, while original precipitation series as a function. We use the partial least-square regression [19] to extract input factors and let original data be real output. Thus, we can build a neural network ensemble prediction method based on PSO algorithm.

4 Experimental Results and Discussion

This paper has investigated applying PSO and neural network ensemble to predict average precipitation on August in Guangxi (including Ziyuan, Guilin, Liuzhou, Bose, Wuzhou, Nanning, Yulin and Beihai). The data set is 48 from 1957-2004 and training sample is 38 (1957-1994) and testing sample is 10 (1995-2004). We use pre-process method (see Section 3) to clean the meteorological data and get 10 variables as network input. The original meteorological data is used as real output.

In order to measure effectiveness of the proposed method, we examine BP algorithm and PSO-based BP (PSO-BP) algorithm. For the purpose of comparison with other neural network models, such as the basic BP neural network, four types of errors, which are commonly found in many papers discussing these models, are also used here. Four types of errors are described as follows:

MSE (the mean squares error):

$$MSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2} \tag{7}$$

MAPE (the mean absolute percentage error):

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|y_t - \hat{y}_t|}{|y_t|} \tag{8}$$

MAE (the mean absolute error):

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t| \tag{9}$$

PR (Pearson Relative Coefficient):

$$PR = \frac{\sum_{t=1}^{n}(y_t - \overline{y_t})(\hat{y}_t - \overline{\hat{y}_t})}{\sqrt{\sum_{t=1}^{n}(y_t - \overline{y_t})^2}\sqrt{\sum_{t=1}^{n}(\hat{y}_t - \overline{\hat{y}_t})^2}} \tag{10}$$

In experiments, a simple 3-layer neural network is selected to handle the prediction problem. The number of neurons in the input layer is 10 and the number of neurons in the output layer is 1. BP parameters are set as follows: the number of neurons in the hidden layer range from 6 ~ 15. The learning rate is 0.9; the momentum factor is 0.7; the iteration times are 1000; the global error is 0.001. PSO-BP parameters are set as

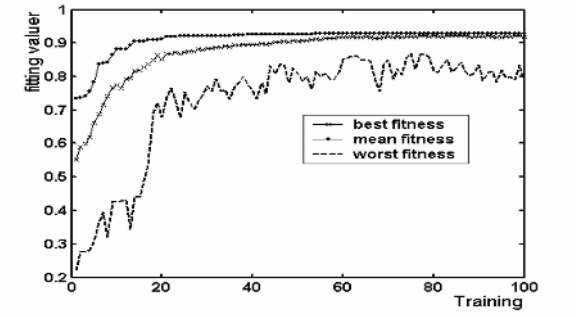


Fig. 1. Fitness values in PSO-BP approach

follows: the iteration times are 100; the population is 100; the minimum inertia weight is 0.1; the maximum inertia weight is 0.9. Other parameters in PSO-BP are the same as BP.

Figure.1 shows the curve of fitness in the training stage. One can see that the maximum, average and the minimum fitness and convergent speed are tending towards stability with increase of iteration number. Therefore, network architecture and connection weights are in near-optimal zone.

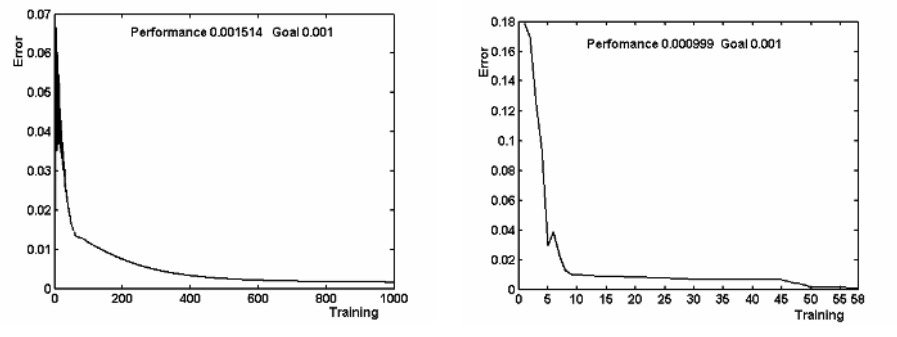


Fig. 2. Error convergence of BP (*left*) and PSO-BP (*right*)

Figure.2 shows a curve of error convergence for a BP network *(left)*. The architecture of the BP is 10 x 10 x 1, and illustrates a curve of error convergence for a PSO-BP network *(right)*. Clearly, the iteration number in PSO-BP is less than BP. Therefore, the convergent speed is greatly improved as well as the convergence reliability.

Figure.3 shows fitting of training sample. Both BP and PSO-BP are deal with data pre-processing. From Table.1 and Figure.3, we can see that learning results of PSO-

Table 1. shows the comparison of four types of error

Model	MAPE(/%)	MSE	MAE	PR
BP	9.58	19.34	16.33	0.898
PSO-BP	2.87	6.25	5.21	0.970

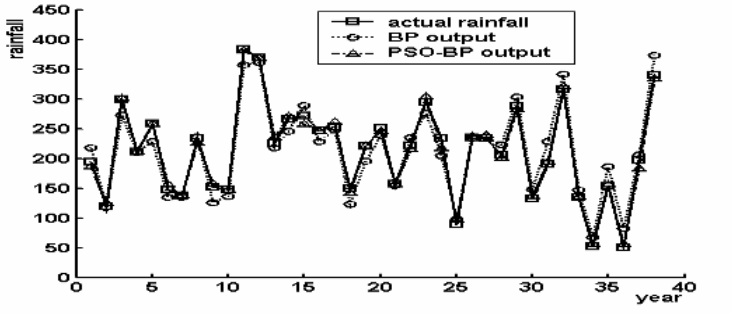


Fig. 3. Training results of BP, PSO-BP and real data

Table 2. gives prediction results of testing samples

year	Real data	BP Model			PSO-BP Model		
		Forest data	AE	APE	Forest data	AE	APE
1995	217.1	279.44	62.34	28.72	223.38	6.28	2.89
1996	273.5	329.24	57.74	20.38	283.19	9.69	3.54
1997	243.4	275.71	32.31	13.28	247.33	3.93	1.62
1998	116.9	98.21	18.69	15.98	125.29	8.39	7.18
1999	313.0	311.88	1.12	0.36	287.05	25.95	8.29
2000	148.4	203.92	55.52	37.41	157.76	9.36	6.31
2001	179.7	146.56	33.14	18.44	167.42	12.28	6.83
2002	209.5	211.57	2.07	0.99	190.60	18.90	9.02
2003	178.5	230.89	52.39	29.35	156.94	21.56	12.08
2004	125.1	157.50	32.40	25.90	133.92	8.82	7.05
Mean			34.57	19.08		12.52	6.48

BP are better than BP. The more important factor to measure performance of a method is to check its generalization ability.

The experimental result shows that generalization ability of PSO-BP outperforms BP under the same network input. Furthermore, we use the same method to train precipitation data on June and July from 1957-1994 and predict precipitation from 1995-2004. The experimental results also show that PSO-BP method is more quickly convergence and better generalization ability than BP method.

5 Conclusions

In this paper, a novel evolutionary neural networks approach, based on particle swarm optimization (PSO) algorithm, is presented for meteorological prediction. PSO algorithm is used to find *L*-groups particles with appropriate network architecture and

connection weights. These *L*- group particles form different neural networks. BP algorithm is further used to train these networks. Finally, these swarming networks are organized as an ensemble to give a combined output. Data pre-processing is adopted for improving data quality. SSA-MGF method reconstructs precipitation data and PLS method decrease dimension of feature space. The precipitation data sets have been used in our experimental study, which shows that the PSO-based ensemble has better generalization ability than the best individual. The results confirm that evolutionary learning should exploit collective information to improve generalization of learned systems.

Acknowledgement

J. Wu acknowledges the support from the Department of Education in Guangxi with Grant No. 200508234 and L. Jin acknowledges the support from the Ministry of Science and Technology in China with Grant No. 2004DIB3J122.

References

1. Dean, A.R., Brian, H.F.: Forecasting Warm-season Burn-off Low Clouds at the San Francisco International Airport Using Linear Regression and a Neural Network. Applied Meteor **41**(6) (2002) 629-639
2. Hsieh, W.W.: Nonlinear Canonical Correlation Analysis of the Tropical Pacific Climate Variability Using Neural Network Approach. Journal of Climate **14** (12) (2001) 2528-2539
3. Jin, L., Ju, W., Miao, Q.: Study on Ann-based Multi-step Prediction Model of Short-term Climatic Variation. Advances Atmosphere Sciences **17**(1) (2000) 157-164
4. Jin, L., Kuang, X.: Study on the Over-fitting of the Artificial Neural Network Forecasting Model. Acta Meteorologica Sinica **62**(1) (2004) 62-69
5. Hansen, L.K., Salamon, P.: Neural Network Ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence **12**(10) (1990) 993-1001
6. Sollich, P., Krogh, A.: Learning with Ensembles: How Over-fitting Can Be Useful. In D. S. Touretzky, M. C. Mozer and M. E. Hasselmo eds., Advances in Neural Information Processing Systems 8, Denver, CO,MIT Press, Cambridge, MA (1996) 190-196
7. Gutta, S., Wechsler, H.: Face Recognition Using Hybrid Classifier Systems, In Processing ICNN-96, Washington, DC (1996) 1017-1022
8. Mao, J.: A Case Study on Bagging Boosting and Basic Ensembles of Neural Networks for OCR, In Processing IJCNN-98, Anchorage, AK **3** (1998) 1828-1833
9. Sollich, P. and Intrator, N.: Classification of Seismic Signals by Integrating Ensembles of Neural Networks, IEEE Transactions Signal Processing **46**(5) (1998) 1194-1021
10. Li, N., Zhou, H. and Ling, J.: Speculated Lesion Detection in Digital Mammogram Based on Artificial Neural Network Ensemble. Advances in Neural Networks ISNN, Springer Press 2005(III): 790-795
11. Bonabeau, E., Dorigo, M. and Theraulaz, G.: Inspiration for Optimization from Social Insect Behavior. Nature **406**(6) (2000) 39-42
12. Kennedy, J. and Eberhart, R.C.: Swarm Intelligence. San Francisco: Morgan Kaufmann Publishers (2001)

13. Rumlhart, D.E., Hinton, G.E. and Williams, R.J.: Learning Representations by Back Propagating Errors. Nature **323**(11) (1986) 533-536
14. Reed, R.: Pruning Algorithms-A Survey. IEEE Transactions on Neural Networks **4** (1993) 740-747
15. Hopfield, J.J.: Neural Networks and Physical Systems with Emergent Collective Computation Abilities. In: Proceedings of the National Academy of Science (1982) 2554-2558
16. Kennedy, J. and Spears, W.: Matching Algorithms to Problems: an Experimental Test of the Particle Swarm and Some Genetic Algorithms on the Multimode Problem Generator. IEEE International Conference on Evolutionary Computation. Alaska, USA (1998)
17. Vautard, S.A.: A Toolkit for Noisy Chaotic Signals. Physical D **58** (1992) 95-126
18. Wei, F. and Cao, H.: The Mathematics Forecast Model and Application of Long Period Time, Beijing, the Meteorological Press (1990) 258-365
19. Wang, H.: The Model and Application of Partial Least-Squares Regression, the National Defense Science and Technology University Press, China (1999) 47-56

A Fast Cloud Detection Approach by Integration of Image Segmentation and Support Vector Machine

Bo Han[1,2], Lishan Kang[1], and Huazhu Song[3]

[1] School of Computer Science, Wuhan University,
Wuhan, Hubei 430072, P.R. China
hanboemail@yahoo.com
[2] Department of Computer and Information Science, Temple University,
Philadelphia, PA 19122, USA
[3] School of Computer Science and Technology, Wuhan University of Technology,
Wuhan, Hubei 430070, P.R. China

Abstract. We proposed a fast cloud detection approach for the geophysical data from Moderate Resolution Imaging Spectroradiometer (MODIS), a premium instrument aboard on NASA's satellite Terra to study clouds and aerosols. Previous pixel-based classifiers have been developed for remote-sensing instruments using various machine learning techniques, such as artificial neural networks (ANNs), support vector machines (SVMs). However, their computational costs are very expensive. Our novel approach integrated image segmentation and SVMs together to achieve the similar classification accuracy while using much less computation costs. It exploited the homogeneous property in local spatial sub-regions and used radiance information from sub-regions, rather than pixels, to build classifiers. The experimental results showed the proposed approach not only greatly speed up the classification training procedure, but also provide insights for domain experts to reveal different cloud types.

1 Introduction

Geophysical data collected through satellite remote sensing are increasingly used in many applications. The high volume datasets (acquired with TB per day), characterized by large variations of observed attributes and the knowledge behind them, create many challenges for data mining and machine learning community (Han et al. 2005). The challenge addressed in this article is using remotely sensed attributes to detect cloud presence, which plays a very important role in earth-atmosphere research. Cloud is the source of precipitations; affects the heat escaping from the earth and meanwhile screens the sunlight reaching the earth. Many important applications, such as retrieval of aerosol particles, climate monitoring and weather prediction, need to detect clouds. Because remote sensing data can reveal the fundamental optical characteristics of clouds - reflecting short-wave radiation, absorbing long-wave radiation, etc, many researchers have proposed approaches to use the satellite data for cloud detection.

Basically, these cloud detection methods are divided into two types: physical models and data mining models. By using domain knowledge, physical models mainly set

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1210 – 1215, 2006.

up a series of thresholds of albedo, brightness temperature, etc. for cloud detection (Ackerman et al. 1998). However, the complexity of the chemical and physical processes involved in the climate implies such models were insufficient to characterize cloud presence with satisfied accuracy. For examples, multilayered clouds made the reflection and emission of radiance complicated. With inadequate understanding of cloud properties, researchers recently applied data mining approaches in the field. Tian et al (2000), Li et al. (2003) and Bankert (2005) explored neural networks, maximum likelihood, decision trees and 1-nearest-neighbor classifiers to detect cloud. Comparing with physical models, the data mining approaches greatly improved classification accuracy. The state of the art accurate cloud classifier is using Support Vector Machine (SVM, Azimi-Sadjadi et al. 2000, Lee et al. 2004). The time complexity of constructing a SVM model is $O(N^2)$ (N is the number of training examples). Its time complexity for test is O(M) (M is the number of support vectors, usually M linearly correlates with N). For satisfying the timely update requirements in applications, Mazzoni et al. (2005) used a small number of subset support vectors to approximate the hyperplane normally created by all M support vectors. Their approach achieved the competitive classification accuracy while running much faster in test stage. However, the most expensive cost in the training stage is still unsolved. With the climate continues changes, SVM models should be timely updated and the model construction time becomes the biggest obstacle in cloud detection. Moreover, mining on the pixels from the compressed images has been used in other applications, however, the technique can not work well in our task since it will loss precision and it is hard to coordinate the localization properties among multiple compressed radiation images.

In this paper, we proposed an approach to integrate image segmentation with SVM classifiers. By exploiting spatial autocorrelation property, it firstly applied image segmentation to divide all training examples into different homogenous sub-regions. Then SVM models are learned based on radiance information from each sub-region, rather than from each pixel. Hence, the training dataset size has been greatly decreased and corresponding training time becomes much cheaper. We tested our approach on 10 scenes acquired by MODIS instrument abroad on satellite Terra. The experimental results showed our integration approach reduced the training time from few hours to few minutes while keep the similar classification accuracy. In addition, the results provide additional insights for researchers to reveal cloud types.

2 Integration of Image Segmentation and SVM Classifier

Previous classifiers using SVM were based on each pixel, so there are a huge number of training examples. However, both remote sensing radiance attributes and the cloud mask showed strong autocorrelation property. Figure 1 (a) and (b) gave an example image for a radiance attribute and the corresponding cloud mask (provided by NASA MODIS team). For some sub-regions, they showed similar or same optical attributes and finally they have the same cloud mask values (0 or 1). This encourages us to consider an approach to first detect homogenous sub-regions and then apply SVM classifier based on these sub-regions.

Image segmentation provides good ways for detecting spatially correlated homogenous sub-regions, such as clustering in attribute space, histogram thresholding,

etc. According to previously reported experimental results and our pilot study, we selected several discriminating radiance attributes for image segmentation.

Next, we applied k-means clustering to segment images. A homogeneous sub-region is defined as the pixels in the same clusters and also has the same class labels (cloud or cloud-free). Hence, for each cluster, we could obtain one or two sub-regions. Each sub-region will be randomly sampled tens of examples for training. For the cluster with pixel size less than a threshold (such as 30), we will include all its pixels for training. In this way, we decreased the examples with homogenous properties and keep the examples with heterogeneous properties. The training data size is greatly reduced while still sufficient for model learning.

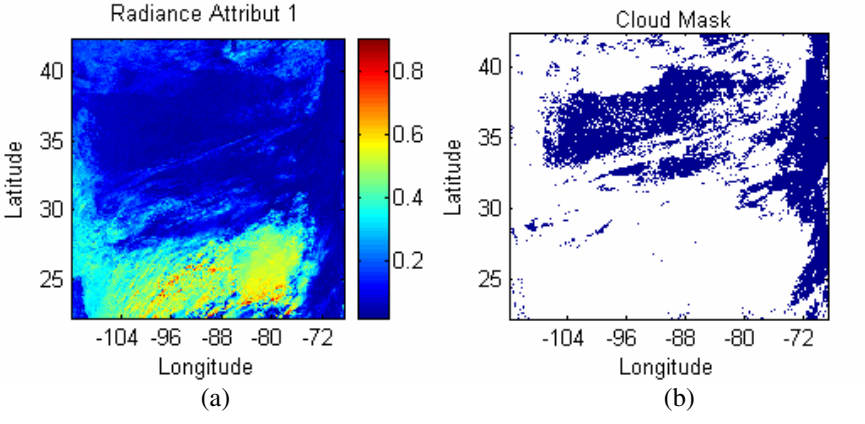


Fig. 1. (a) One radiance image (b) Cloud mask image (white - cloud; blue - cloud-free)

Using the reduced-set of examples, we can quickly construct a SVM classifier. SVM (Vapnik 1995) is optimized to find the decision hyperplane with the maximal separation margin between positive and negative examples. By representing each sub-region r_i with pair $(\mathbf{A}_i, y_i)$, where $\mathbf{A}_i = [a_{1i} \ldots a_{Ki}]$ is an attribute vector, $y_i = -1$ for negatives (cloud-free) and $y_i = +1$ for positives (cloudy). By using the Lagrangian function to solve a constrained optimization problem, the learned SVM classifier is described in (1),

$$SVM(A) = b + \sum_{j=1}^{N_S} \alpha_j y_j KF(A_j, A) \tag{1}$$

where N_S is number of support vectors selected from training examples, $\boldsymbol{A}_j$ denotes a support vector with K radiance attributes in a sub-region, α_i, $i = 1 \ldots N_S$, and b are model parameters obtained by optimization, and KF is an appropriate kernel function. For these parameters, we exhaustively searched the optimal combination of them with cross-validation method.

Finally, we applied (1) on each test example to obtain a soft output as its class label. To further improve the classification accuracy, we decide the class label of a pixel by averaging its soft output with those of neighborhood pixels. If the considered

pixel has a high confidence on one class, the average will not change its label. Otherwise, if the pixel is very close to decision hyperplane and the soft output has almost equal value to be positive class and negative class, the average method will encourage a homogenous sub-region, which is consistent with the autocorrelation property of spatial data.

3 Experimental Results

In the experiments, we used MODIS Level1B radiance product MOD02SSH. It includes 36 radiance bands information globally covering a wide spectral range. According to the previous reported experiments, we selected the band 1, 2, 6, 7, 20, 31 and 32 as classification attributes. Totally, we used 10 scenes in the experiments. Each scene includes 110026 pixels, including radiance information observed by MODIS in 5 minutes. If the 10 scenes data are merged, a standard cross-validation in which data are randomly divided into training and test samples is likely to produce overly optimistic accuracy estimates because of significant spatial correlation. To avoid this we used leave-one-scene-out cross-validation instead. In other words, we first applied image segmentation on 9 scenes and selected training examples from the resulted sub-regions. These examples were used to learn a SVM classifier. Then the data from the excluded scene were tested. The leave-one-scene-out process was repeated 10 times for each test scene, and the average accuracy is reported as an estimate of classification quality. The cloud detection results are compared with the cloud mask provided by MODIS product MOD35_L2.

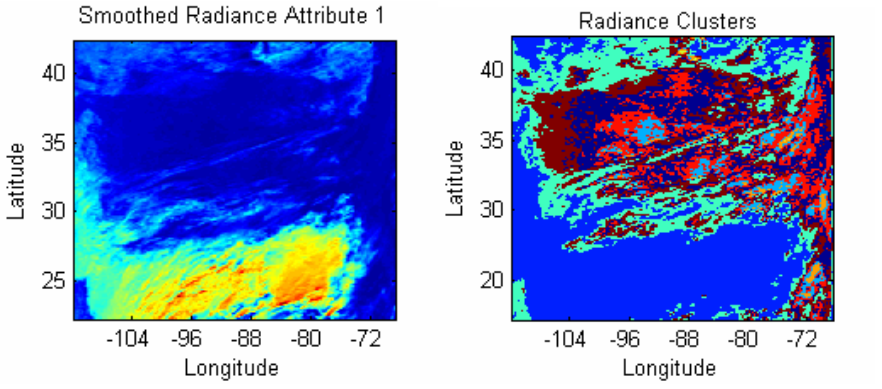
In the first step, we used the following 3*3 convolution mask for smoothing,

$$W = \frac{1}{10} * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{2}$$

The smoothed radiance image for figure 1 (a) is showed at figure 2 (a). It clearly shows smoothing procedure effectively eliminates impulsive noise while keeping the significance of the central pixel.

3.1 Image Segmentation by Using K Means Clustering

We applied K means clustering for image segmentation since it is simple and in linear time of training example size. K is selected ranging from 6 to 10. A pilot study shows K=8 achieve a best or near-best basis for classification. Figure 2 (b) shows the clustering results for the figure 3 with K=8. The clustering iterations generally converge after 6-7 rounds. After discussing with domain experts, we found the clusters correspond to different cloud types (cumulus, cirrus, etc.) or different cloud-free areas covered with different surfaces (land, ocean, etc.), which suggested that the classification based on segmented sub-regions are reasonable in domain, and also provided insights for curators to identify cloud properties or surface radiance characters.



(a) (b)

Fig. 2. (a) Smoothed radiance image (b) Image Segmentation by Clustering

Table 1. Comparison between pixel-based SVM classifier and sub-region-based SVM classifier

Test Scene	Pixel-based SVM Classifier		Sub-region-based SVM Classifier	
	Accuracy	Time (sec)	Accuracy	Time (sec)
1	0.8621	5638	0.8597	135
2	0.8496	5807	0.8535	129
3	0.7627	6642	0.7812	152
4	0.7870	6345	0.7866	165
5	0.6939	7854	0.6797	191
6	0.8853	5473	0.8929	143
7	0.7897	6328	0.7701	179
8	0.8512	5670	0.8608	148
9	0.8291	5995	0.8050	172
10	0.8937	5456	0.8854	133

3.2 SVM Classifier

In each cluster, 1-2 homogenous sub-regions are obtained according to corresponding cloud masks. For each cross-validation, we selected 1000-2000 examples from 9 scenes for training. With exhaustively search on SVM parameter, we found a good choice turned out to be a polynomial kernel with p=3, regularization parameter C=0.11. The soft classification outputs are averaged over a region by using convolution mask (2). Table 1 compared the training time and accuracy between the pixel-based SVM classifier and the proposed sub-region-based SVM classifier. The former achieves average accuracy 0.8204, in 6121 seconds as average. While the latter achieves average accuracy 0.8175, in 155 seconds as average.

4 Conclusions

The proposed cloud detection approach integrated image segmentation and SVM algorithm together. It used clustered homogeneous sub-region information to replace huge size of pixels as training examples. The experimental results based on MODIS data showed the approach achieved the similar classification accuracy while running around 40 times faster. In addition, it revealed insights for earth science researchers to understand cloud properties and surface radiance characters. This function encouraged us to further enhance our approach for cloud type detection in the future.

Acknowledgments

We thank NASA Langley Research Center for their data support. We are also grateful for insightful discussions with professors at Temple University.

References

1. Ackerman, S. A., Strabala, K. I., Menzel, W. P., Frey, R. A., Moeller, C. C., Gumley, L. E.: Discriminating Clear-sky from Clouds with MODIS. .J. of Geophy. RES-ATMOS. 103 (1998) 32141-32157
2. Azimi-Sadjadi, M.R., Zekavat, S.A.: Cloud Classification Using Support Vector Machines. Proceedings of the 2000 IEEE Geosci. and Remote Sensing Symp., Vol. 2, (2000) 669-671
3. Bankert, R.: Naval Research Laboratory Monterey GOES Cloud Classification (website) (2005) http://www.nrlmry.navy.mil/sat-bin/clouds.cgi.
4. Han,B., Vucetic,S., Braverman, A., Obradovic, Z.: Construction of an Accurate Geospatial Predictby Fusion of Global and Local Models. 8th Int. Conf. on Information Fusion. 2005.
5. Lee, Y., Wahba, G., Ackerman, S.: Cloud Classification of Satellite Radiance Data by Multicategory Support Vector Machines. J. of Atmos. and Oceanic Tech. 21 (2004) 159–169
6. Li, J., Menzel, W.P., Yang, Z., Frey, R.A., Ackerman, S.A.: High-spatial-resolution Surface and Cloud-type Classification from MODIS Multispectral Band Measurements. J. of Appl. Meteorology 42 (2003) 204–226
7. Mazzoni, A.H., Garay, M., Tang, B., Davies, R.: A MISR Cloud-type Classifier Using Reduced Support Vector Machines. Proceedings of the Eighth Workshop on Mining Scientific and Engineering Datasets, SIAM Int. Conf. on Data Mining (2005)
8. Tian, B., Azimi-Sadjadi, M.R., Vonder Haar, T.H., Reinke, D.: Temporal Updating Scheme for Probabilistic Neural Network with Application to Satellite Cloud Classification. IEEE Trans. on Neural Networks 11 (2000) 903–920
9. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, New York (1995)

Application of Support Vector Machines to Vapor Detection and Classification for Environmental Monitoring of Spacecraft

Tao Qian[1], Xiaokun Li[1], Bulent Ayhan[1], Roger Xu[1], Chiman Kwan[1], and Tim Griffin[2]

[1] Intelligent Automation, Inc., 15400 Calhoun Drive, Suite 400, Rockville, MD 20855, USA
{tqian, xli, bayhan, hgxu, ckwan}@i-a-i.com
[2] NASA Kennedy Space Center (KSC), Texas, USA

Abstract. Electronic noses (E-nose) have gained popularity in various applications such as food inspection, cosmetics quality control [1], toxic vapor detection to counter terrorism, detection of Improvised Explosive Devices (IED), narcotics detection, etc. In the paper, we summarized our results on the application of Support Vector Machines (SVM) to gas detection and classification using E-nose. First, based on experimental data from Jet Propulsion Lab. (JPL), we created three different data sets based on different pre-processing techniques. Second, we used SVM to detect gas sample data from non-gas background data, and used three sensor selection methods to improve the detection rate. We were able to achieve 85% correct detection of gases. Third, SVM gas classifier was developed to classify 15 different single gases and mixtures. Different sensor selection methods were applied and FSS & BSS feature selection method yielded the best performance.

1 Introduction

Electronic noses (E-nose) have found more and more applications in the past decade, such as food and cosmetic industries [1], patient's health diagnostics, environmental monitoring in the Space Shuttle and International Space Station [2], explosive detection, etc. A typical E-nose has an array of sensors that respond differently to different chemicals. Each sensor generates certain resistance values when it is exposed to vapors. The relative change in resistance value normally gives indication about the presence of vapors and the level of resistance change has certain nonlinear relationships with the vapor concentration level.

Recent theoretical advances and experimental results have drawn considerable attention to the use of kernel functions in data clustering and classification. Among them stands out the SVM. SVMs were first suggested by Vapnik in the 1960s [3] for classification and have recently become an area of intense research owing to developments in the techniques and theory coupled with extensions to regression and density estimation. SVM is a general architecture that can be applied to pattern recognition and classification, regression, estimation and other problems such as speech and target recognition.

In this paper, we summarize our studies on the application of SVM to E-nose data from JPL. Our main focus is to detect and classify 10 different vapors and also some

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1216 – 1222, 2006.

combinations of them. In Section 2, we will brief review the E-nose data and the basics of SVM. In Section 3, we summarize three methods to address the noise and drift issues. Section 4 describes the gas detection and gas classification results. With sensor selection, SVM could achieve over 85% correct classification and 85% for gas detection. Finally, we conclude the paper in Section 5.

2 E-Nose, Data, and SVM

The E-Nose consists of an array of vapor sensors. Popular vapor sensors are metal oxide semiconductor (MOS), surface acoustic wave (SAW), composite polymer (CP), conducting polymer, gas chromatography, mass spectrometry, etc. A typical time response of a signal sensor is shown in Fig. 1.

To the output signals of the sensors, we need to do a very important data preprocessing called feature extraction, which can get the features from signal data for the classifier. There are two kinds of features: transient response and steady states. The steady states are used for classification because of its robustness. However, most sensors require a long time to stabilize. To decrease the time, the transient response at specific time is used, which is much earlier than the time for the sensors to arrive at the steady states. The sensor measurements at time *T(*=30 or 90 in Fig. 1) can be used directly as features for classification. We also can perform some other preprocessing to the raw signal to eliminate the noise introduced by the sensors and environment.

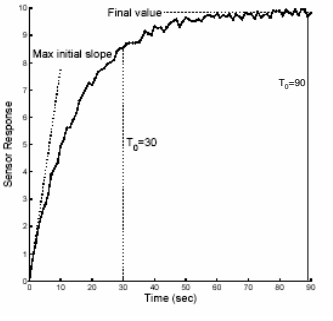


Fig. 1. Typical time response of a single sensor [4]

Jet Propulsion Lab. developed an E-nose with 32 sensors. Details of the E-nose can be found in [2]. According to references [5] [6], the two key elements in the implementation of SVM are the techniques of mathematical programming and kernel functions. The parameters are found by solving a quadratic programming problem with linear equality and inequality constraints. The flexibility of kernel functions allows the SVM to search a wide variety of hypothesis spaces.

Based on materials in [6], SVM has great potential to improve the performance of existing classification algorithms. Here the theory of SVM is briefly reviewed in a two-class classification problem, the classes being P, N for $y_i = +1, -1$ respectively. This can easily be extended to $k-$class classification by constructing k two-class

classifiers [7]. The geometrical interpretation of Support Vector Machine (SVM) is that the algorithm searches for the optimal separating surface, i.e. the hyperplane. That is, in a sense, equidistant from the two classes.

3 Data Preprocessing

Since the sensors have drift and noise, we used three different pre-processing methods to create three different types of sample data sets (Type I, II and II data sets).

Type I data set: The raw data was low-pass filtered first. Then we used sensor resistance value at first sample point as static R_0. Relative resistance change ΔZ_k is calculated with this R_0.

$$\Delta Z_k = \frac{Z_k^l - R_0^l}{R_0^l}, \; l \; is \; sensor \; number \tag{1}$$

Type II data set: Here the raw data was low-pass filtered first. The sensor resistance value before i^{th} gas event is used as dynamic R_i for sensor outputs of i^{th} gas events. Relative resistance change ΔZ_k for i^{th} gas events is calculated with this R_i. Using dynamic R_i is trying to reduce the baseline drafting in the sensor output.

$$\Delta Z_k = \frac{Z_k^l - R_i^l}{R_i^l}, \; l \; is \; sensor \; number, \; i \; is \; gas \; event \tag{2}$$

Type III data set: A Kalman filter was used to eliminate the high frequency noise in the sensor outputs. The low order polynomial fitting method was used to find the baseline drafting curve. By subtracting the baseline drafting curve from Kalman filter output, we can eliminate the baseline drafting effect in the sensor outputs.

$$\Delta Z_k = f(Z_k^l)_{Kalman} - g(Z_k^l)_{Polynomial-fitting} \tag{3}$$

4 Gas Detection and Classification

For the E-Nose data sets from JPL, we have two steps to determine the vapor types in the sample. In the first step, we used two-class classifier to determine if there is gas in the sample. The sample data set is generated as below:

- A total of 148 gas events and one sample per gas event
- 23,000 non-gas samples are reduced to 150 samples by K-mean clustering

We assigned 148 gas sample labeled as 1 (gas-event), 150 non-gas samples labeled as 0 (non-gas-event). Holdout method was used for SVM classifier performance evaluation. To evaluate the SVM performance accurately, we divided the whole data set into three parts equally. The first part was for SVM training. The second part was for SVM validation to tune the parameters. The third part was for SVM testing. The testing part is always kept out from SVM training and sensor selection procedure. It is only used to test SVM performance. After gas detection, SVM were used again for gas classification.

The main goal of this effort was to investigate the classification performance of SVM. Extensive studies for vapor classification of E-Nose based on data from NASA clearly demonstrated that the performance of SVM is better than other classifiers.

4.1 Gas Detection Results

Initially we used all sensor outputs in our gas detection algorithm. However, since sensor outputs are not consistent, they degrade the overall detection performance. Then we used a feature selection method to pickup the most reliable sensors for SVM in gas detection. The most common sequential search algorithms for feature selection are forward sequential selection (FSS) & backward sequential selection (BSS) [8] which procedure were shown in Fig. 2 [8].

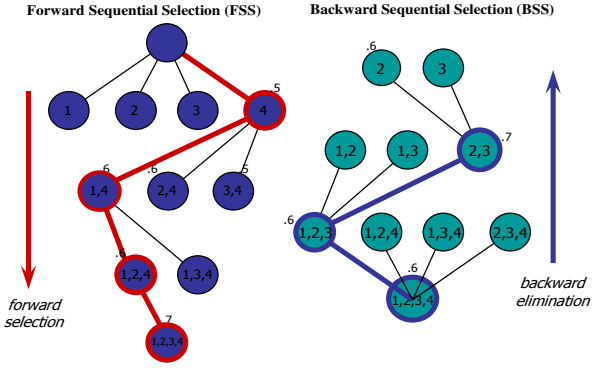


Fig. 2. Diagram for FSS and BSS algorithms

200 holdout iterations were performed. The results are summarized in Table 1. It can be seen that FSS & BSS feature selection methods give a significant improvement for the detection rate. The best detection rate is improved from 73% to 85.05% with the help of sensor selection. From Table 1, we also observe that by improving the baseline drifting compensation algorithms increase the gas detection rate.

Table 1. Gas detection rates for three different data type data sets

	Date Type I	Data Type II	Data Type III
All sensors	51.87%±4.26%	73.08%±4.11%	55.57%±4.45%
Selected sensors	60.02%±3.27%	85.05%±2.05%	77.73%±2.33%

4.2 Gas Classification

After gas is detected by the SVM gas detection classifier, we need to identify the type of gas in the samples. There are 10 different single gases and 5 binary mixtures in the JPL data set. The multiple classes SVM classifier is constructed to classify them. This

classifier includes 15x14/2=105 two-class SVMs and the final decision is decided by the maximum voting rule.

Because all sample numbers for each gas are less than 3 times of the input data dimension (3x32=96), Leave-One-Out method was used to evaluate the SVM performance. Same three different type data sets were used for evaluation.

Sensor selection based on the consistency/repeatability of the sensor response: Here sensor selection algorithm is based on the consistency of the sensor outputs under the same experimental condition. The idea of the algorithm is shown below:

- Calculate the variance of the sensor responses for 32 sensors.
- Pick up n sensors with lowest variances (n<32).
- Use maximum voting to find the best sensor subset for each data file.
- Calculate the frequency of each sensor appeared in whole sensor subset lists. Threshold is set to select the best sensors which appear mostly in the lists.

We selected the most reliable sensors from 32 sensors based on whole data set. Table 2 is the selection results for different thresholds.

Table 2. Selected sensor from 32 sensors based on whole data set

threshold	Index of selected sensors											
0.9	8	9	15	20	10	17	18	32	6			
0.8	8	9	15	20	10	17	18	32	6	25	16	19

12 selected sensors (threshold=0.8) were used for SVM classification. The simulation results are shown in Table 3. The best classification rate was 65.54%. This sensor selection algorithm gave 10% classification rate improvement.

Optimal weights using GA algorithm: Genetic algorithm (GA) can be used to optimize the weight for each sensor output based on the reliability of the sensor. This is similar to feature selection but more flexible since the weights can change continuously at any value larger than zero. Vapor identification error rate of SVM classifier was used as the objective function to optimize the weights of sensor outputs. The simulation results are shown in Table 3. GA algorithm improved the classification rates for all three data types by 10% ~ 13%.

Table 3. Gas classification rates using sensors selected by GA for three data type data sets

	Date Type I	Data Type II	Data Type III
All sensors	55.41%	29.73%	28.38%
Selected sensors based on consistency	65.54%	35.81%	23.65%
Selected sensors based on GA	68.24%	41.89%	38.51%

Sensor selection based on the forward sequential selection (FSS) & backward sequential selection (BSS): Based on our observation of the E-Nose data, we found different sensors have different sensitivity for different gases. For two different gases, we can select a set of sensors which can separate these two gases most efficiently. For

every 2-class SVM in multiple classes SVM classifier, an optimal subset of sensors is found using FSS & BSS method. We can get 105 sensor subsets for all 2-class SVM using FSS & BSS algorithm. When the testing data come, different selected features are fed in the different 2-class SVMs and maximum voting decides the final gas label.

Based on sensor selection simulation results above, we observed that Type I data have out-performed in the gas classification part. Here we only used the Type I data for FSS & BSS algorithm. To get the more realistic SVM performance evaluation, we have to generate different scenarios to get the FSS & BSS selected sensor subset. For each testing point, we take it out from data set. FSS & BSS sensor selection procedure is applied to the remaining data. Because for each testing data point, we need to have 105 2-class SVM classifiers, for whole data set, we need generate 14x148+105-14=2163 different sensor subsets using FSS & BSS feature selection algorithm. Each testing data point will be applied to 105 SVMs using selected features and final class label is obtained by maximum voting**.** Simulation result shows 85% correct classification by using FSS & BSS algorithm for Type I data.

From the simulation results above, we observed that selecting different sensors for different gas classifiers will boost the classification rate. This validated our assumption that different sensors have different sensitivity for different gases.

5 Conclusions

From this period research, we investigated different sensor selection methods and different classification algorithms. Using FSS & BSS sensor selection method, we can improve the gas detection and gas classification rate dramatically. The best classification rate was 85%.

Acknowledgement

This research was supported by NASA Kennedy Space Center under contract number NNK04OA02C. Encouragements and supports from Dr. Margaret Ryan and Dr. Hanying Zhou are deeply appreciated.

References

1. Hines, E. L., Llobet, E., Gardner, J. W.: Electronic Noses: a Review of Signal Processing Techniques. IEE Proc. Circuits Devices Syst. 146 (1999)
2. Ryan, M. A., Zhou, H., Buehler, M. G., Manatt, K. S., Mowrey, V. S., Jackson, S. P., Kisor, A. K., Shevade, A. V., Homer, M. L.: Monitoring Space Shuttle Air Quality Using the Jet Propulsion Laboratory Electronic Nose. IEEE Sensor Journal 4 (2004)
3. Burges, C.: A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 2, Kluwer Academic Publishers, Boston (1998) 121-167
4. Linnell, B., Young, R., Buttner, W.: Electronic Nose Vapor Identification for Space Program Applications.
5. Cristianini, N., Shawe-Taylor., J.: An Introduction to Support Vector Machines, Cambridge University Press (2000)

6. Osuna, E., Freund, R., Girosi., F.: Support Vector Machines: Training and Applications. AI Memo 1602, MIT (1997)
7. Hsu, C., Lin, C.: A Comparison of Methods for Multiclass Support Vector Machines, IEEE Trans. Neural Networks 13 (2002) 415-426
8. David, W., Richard, L. B.: Comparative Evaluation of Sequential Feature Selection Algorithms, Statistics Workshop (1995)

Artificial Neural Network Methodology for Three-Dimensional Seismic Parameters Attenuation Analysis

Ben-yu Liu[1], Liao-yuan Ye[1], Mei-ling Xiao[1], Sheng Miao[1], and Jing-yu Su[2]

[1] Institute of Public Safety and Disaster Prevention, Yunnan University, Kunming 650091, China
{liubenyu, lyye, mlxiao, msheng}@ynu.edu.cn
http://www.srees.ynu.edu.cn/structure/index.html
[2] Civil Engineering Department, Beijing University of Technology, Beijing 100022, China
jysu@bjpu.edu.cn

Abstract. With the accumulating of the strong earthquakes records, it becomes practicable to achieve the more accurate attenuation relationships. Based on the seismic records of West American, the Radial Basis Function (RBF) and Back Propagation (BP) artificial neural networks model are respectively constructed for three-dimensional seismic parameters attenuation relationship. The RBF model is nice fitting for the training data, although it has great errors on other tested points. While the BP model is not good than the RBF model for the training data, it possesses a better consecutive property in the whole area. It is a proper neural network model for the problem. After training with the selected records, the Neural Networks (NN) shows a good fitting with the training records. And it is easy to construct three-dimensional model to predict the attenuation relationship. In order to demonstrate the efficiency of the presented methodology, the contrast is discussed for the results of the BP model and three typical traditional attenuation formulae.

1 Introduction

Taking from the seismic ground motion attenuation relationships, the movement can easily be predicted if certain earthquake occurred. Traditionally, there are two ways to get this special relationship. One is make a theoretical attenuation model and analyze its influence coefficients and parameters; the other is statistical way from the stations seismic records. Since the complicated nature of the problem, the predicted results of most relationships can not accord with the stations records nice.

As we know, Artificial Neural Networks (ANN) are highly parametric functions of the input variables through processing units, whose high connectivity makes them suitable for describing complex input-output mappings without resorting to a physical description of the phenomenon. Some studies in this problem have been reported. Zheng Guanfen discussed the earthquake intensity attenuation using Back Propagation Neural Networks (BPNN) [1]. Wang Hushuang constructed a NN model to simulate the peak seismic parameters attenuation relation and a NN to relate the intensity

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1223 – 1230, 2006.

with peak seismic parameters [2]. By using the peak horizontal acceleration records acquired in Lancang-Gengma Earthquake in 1988, Cui Jianwen constructed a NN model to predict the peak ground acceleration attention relationship in Yunnan region [3].

This paper using the records collected in West America where is abundant in seismic records to study the three-dimensional (3D) attenuation relation of the ground seismic parameters. Up to now, it is very difficult to describe the 3D relationships in geophysics. The ANN model just presented a solution to this question. This paper only studied the three dimensions Peak Ground Acceleration (PGA), the peak ground velocity and peak ground displacement can be studied in the similar way. The 3D in the earthquake records are two horizontal directions and one vertical direction. In the paper, the horizontal directions are named as direction 1 and 2, the vertical direction as direction 3.

2 Experimental Statistical Model

The experimental models were simply regressed from the strong earthquakes records. As for the records in West America, the following models were famous.

Joyner-Boore [4] （West America, bedrock and hard soil, 1981）

$$\lg PGA = -1.02 + 0.249M - \lg(\sqrt{D^2 + 7.3^2}) - 0.0025\sqrt{D^2 + 7.3^2} \tag{1}$$

This model did not include the magnitude saturation which is proved in many strong earthquakes. Huo Junrong [5] revised this model in order to consider the magnitude saturation, as following,

$$\lg PGA = -0.935 + 1.241M - 0.046M^2 - 1.904\lg\left(D + 0.3268e^{0.6135M}\right) - 3 \tag{2}$$

In these two relations, the attenuation ratio is same when the magnitude is different. It means that the same magnitude space in the equations has the same variety even in different distance, which is different with the theoretical results and strong earthquakes observations. The fact is famous as magnitude saturation phenomenon. It says even with middle magnitude earthquake (such as M=5), the PGA in the near field may reach a high value; while with the more strong earthquakes, the PGA can't exceed an upper limit. The far field amplitude of the ground motion may attenuate with different ratio with respect to different magnitude [6]. Thus, Wang Guoxing presented the following equation,

$$\lg PGA = 4.053 - (2.797 - 0.251M)\lg(R + 8.84) - 3 \tag{3}$$

Where PGA is peak ground acceleration, R is the distance from the observation site to the focal, D is the direct distance from the observation site to the striking fault and M is the Richter Magnitude of the earthquake.

The relations presented by Boore D. M. [6, 7] also included the fault types, the shear wave velocity of the soil where is 30 meters below the observation station. And he made the solution more complicated.

3 ANN Analysis for the 3D Attenuation Relation

From the attenuation models presented above, we learned that the different statistical model implied different assumption of the relation that is why the given models demonstrated great discrepancy with the earthquake recordings. On the other side, the theoretical models are deduced from the very simplified assumptions, which is different greatly with the real seismic process and geology situation. How to minimize the assumptions in the models and reach a rather feasible attenuation relation for the seismic parameters is a valuable problem.

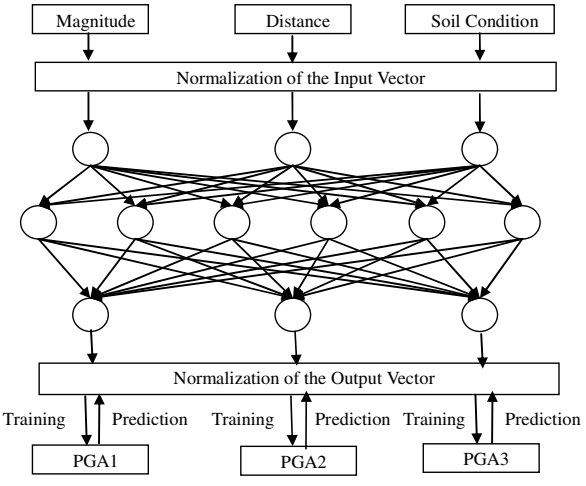


Fig. 1. The PGA, PGV and PGD should be constructed in three ANN models. This figure only demonstrated the architecture of the PGA. The PGV and PGD model can just be replaced the PGA with themselves. The number of perceptrons in the hidden layer was determined by networks itself according to the training effect.

Since ANN is self-studied, self-organized and self-adapted, it is interesting to use this methodology for seismic parameters analysis. The method can learn experiences from the real observation records, so the sufficient learning data is required. The ANN model needs no assumption which may suitable for this seismic problem.

When used for data fitting, the NN model such as BP and Radial Basis Function (RBF) are often employed. The RBF network is a linear basis function method, which can present 100 percent accuracy for the training data; while for the testing data may exhibit great discrepancy. The BP network is limited in the [-1, 1], so it can not behave in great difference. This paper takes the BP network as main model for the problem.

Figure 1 is the ANN model constructed in this paper for the seismic ground parameter analysis. The training samples are the records acquired after the strong earthquakes. The Magnitude, Distance and Soil Condition are selected as input part. The intensity of the observation station was excluded because it can not be used in the prediction procedure. The output parameters were the seismic parameters at the ob-

servation station, such as PGA, Peak Ground Velocity (PGV) and Peak Ground Displacement (PGD). Because the difference nature of the PGA, PGV and PGD, three ANN model were needed to be constructed for the problem.

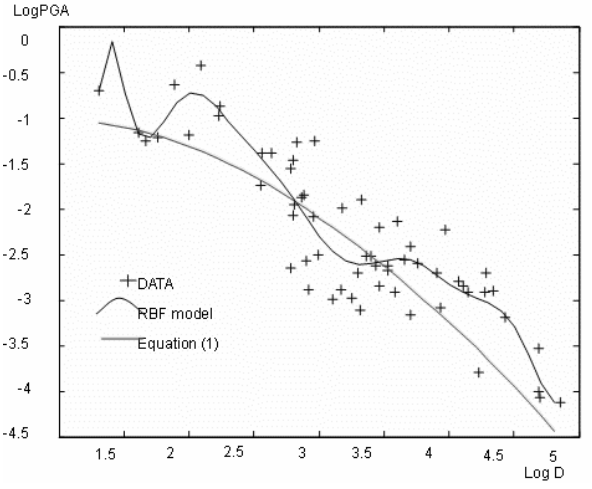


Fig. 2. RBF fitting with contrast to equation (1)

4 Data Processing

The data in this paper was coming from reference [9], which includes 63 items strong earthquakes of the West America. In order to avoid the deflection of one earthquake attenuation relation to the whole region relations, the records were selected. All the records in the earthquakes which had small data were accepted, while just part of the records was selected for those earthquakes which had excessive records. 186 PGA data were picked from the dataset. The magnitude changed from 4.3 to 7.7, the epicenter distance changed from 1 km to 214 km and the acceleration changed from 0.0081 to 1.2219 (g).

The RBF network was used to demonstrate its efficiency in data fitting. The training data was from 6.0 to 6.5 in magnitude. And the result of the equation (1) was drawn in figure 2. It can easily be pointed that the RBF network can fit the data with certain satisfaction. While in the place with insufficient data, it demonstrate sudden jump of the line, which means the unexpected results may occur.

Because of the situations demonstrated in figure 2. The BP network was chosen for the 3D seismic parameters prediction analysis. In order to present a strong impression this method, the results of equations (1) – (3) are listed in Figure 3 -5.

The data for the network training were also included in these figures. The magnitude of attenuation relation changed from 5.5 to 7.7, the epicenter distance changed from 5 km to 150 km. One of the horizontal direction results was listed in order to balance with other results.

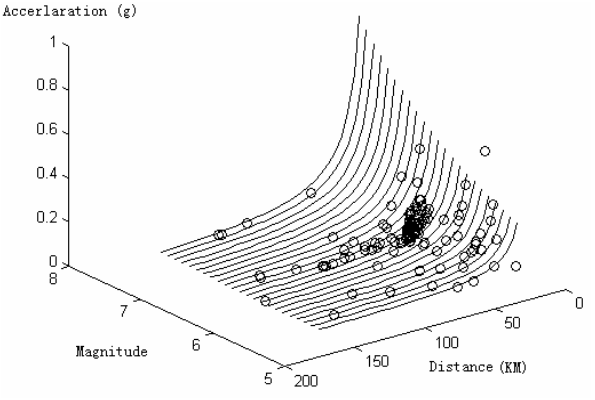


Fig. 3. The prediction results of equation (1)

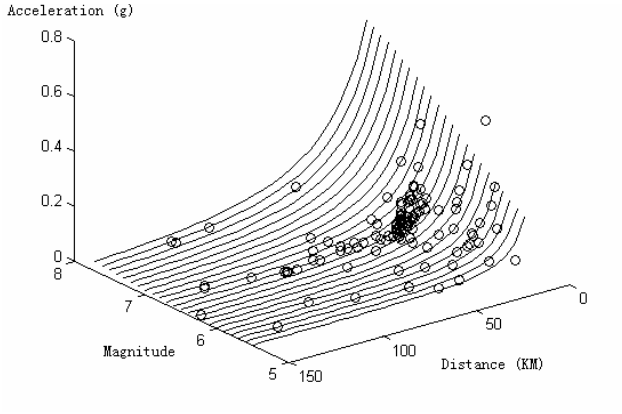


Fig. 4. The prediction results of equation (2)

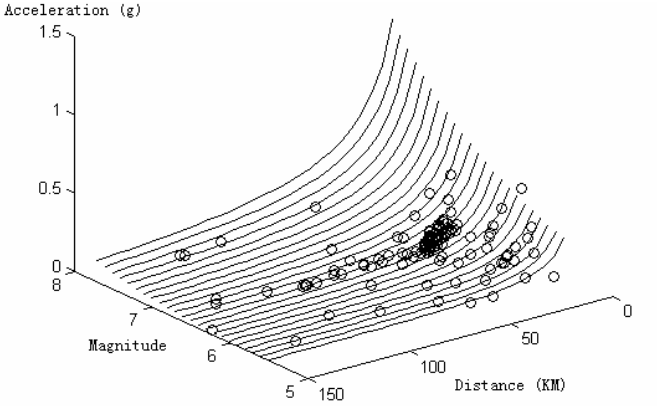


Fig. 5. The prediction results of equation (3)

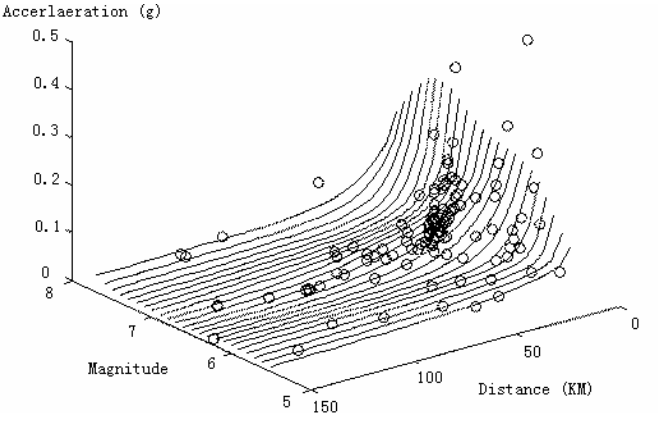


Fig. 6. Prediction of artificial neural network

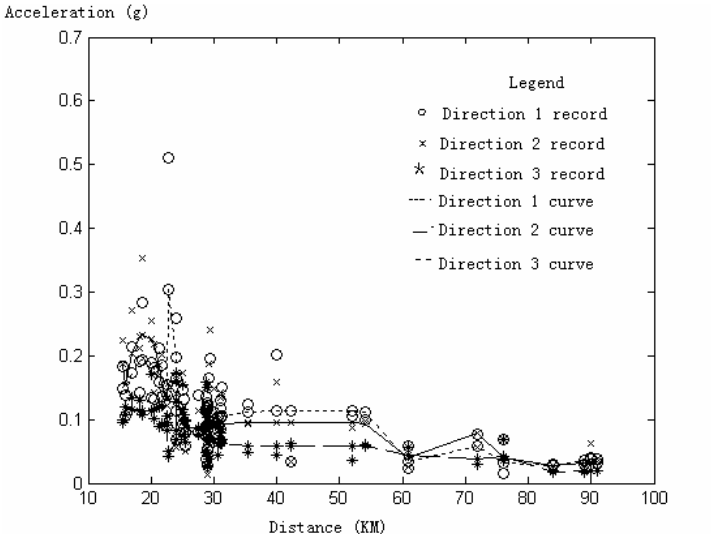


Fig. 7. The ANN prediction curves of the three dimensional attenuation of PGA when the magnitude equal to 6.6

Because most of the earthquake records are collected at middle distance with middle magnitude, the relationships introduced above at this area demonstrated favorable fitting results. But for the high magnitude data fitting, the results may be greatly different. The result of equation (3) is the maximum, the equation (1) in the middle, and equation (2) is the minimum. With the increasing of the magnitude, the acceleration is increased which failed the magnitude saturation fact. The ANN model demonstrates the acceleration decrease at the high magnitude, which may represent the magnitude saturation phenomenon. Of course, the insufficiency of the records may be one reason for the acceleration decrease. For the detail situation of the attenuation relation, the three former equations are changed steady while the ANN model is different.

Furthermore, the ANN method can easily get the 3D seismic parameter attenuation relationships. The target of PGV and PGD can be attained just change the training target from PGA to PGV or PGD, and no more assumptions were need for this methodology.

In order to demonstrate the advantage of ANN method, the earthquakes data which magnitude is 6.6 is selected. There are 55 records for the hard soil site in the dataset. The ANN data fitting and the original data were pointed in figure 7.

In figure 7, the fitting curves of the direction 1 and direction 2 are relative close. This is because they represent the two horizontal directions. In the near field, they are a little different, and while in the far field they are similar. The direction 3 is the vertical direction, which has smaller value than the horizontal direction and has the similar attenuation relationship with the horizontal directions.

5 Conclusions and Suggestions

From the contrast with the traditional seismic parameter attenuation relationships, this paper presented an ANN model to simulate this special relation. After training with the data acquired in West America, the model demonstrated favorable results. Unlike the traditional method, this model involve the magnitude saturation and length influence of the focal just by training the model with the proper data, and the three dimensional vectors and three parameters such as PGA, PGV and PGD can easily be acquired.

From the ANN analysis of this paper, it can easily be found out that the attenuation curves of the horizontal and vertical are different. The results of this paper can provide three dimensional attenuation relationships for engineering usage.

By employed the RBF and BP networks to the training data, this paper demonstrated that the BP network is more suitable for this problem. The architecture of the BP network was constructed in this paper. By contrast with the traditional models, it demonstrated that the ANN model is more suitable for the problem. With the sufficient training data, the model may present a reasonable relationship for the seismic parameters attenuation with displacement.

Acknowledgements

This research is financial supported by the National Science Foundation of China under the grant number 40564002, the National Science Foundation of Yunnan Province under the grant number 2004E007R and the Science and Engineering Foundation of Yunnan University under the grant number 2004Z005B.

References

1. Zhen, G.F., Tao, X.X.: Construct the Intensity Attenuation Relation Using ANN Method. Earthquake Engineering and Engineering Vibration 13(1) (1993) 60–66
2. Wang, H.S..: Intelligent Prediction of the Peak Seismic Parameters Based On ANN. Journal of Seismology 15(2) (1993) 208–216

3. Cui, J.W., Fan, Y.X., Wen, R.Z.: Establishment of Attenuation Law of Acceleration Peak Value by Using Neural Network. Earthquake Research 20(3) (1997) 296–306
4. Joyner, W.B., Boore, D.M.: Peak Attenuation and Velocity from Strong-Motion Records Including Records from 1979 Imperial Valley, California, Earthquake. Bulletin of Seismology Society of America 71(6) (1981) 2011–2038
5. Huo, J.R.: Near Field Ground Motion Attenuation Research. Ph. D. Thesis of the Institute Of Engineering Mechanics, China Earthquake Administration (1989)
6. Wang, G.X., Tao, X.X.: A New Two-Steps Method for Fitting Ground Motion Attenuation Relationship. Earthquake Engineering and Engineering Vibration 20(1) (2000)25–28
7. Boore, D.M., Joyner, W.B., Fumal, T.E.: Estimation of Response Spectra and Peak Accelerations from North America Earthquakes: an Interim Report. U. S. Geol. Surv. Open File Report (1993) 93–509
8. Boore, D.M., Joyner, W.B., Fumal, T.E.: Estimation of Response Spectra and Peak Accelerations from North America Earthquakes: Part 2, Interim Report. U. S. Geol. Surv. Open File Report (1994) 94–127
9. Hu, Y.X., Zhang, Y.M., Shi, Z.L.: Training Material for the Code of Evaluation of Seismic Safety for Engineering Sites. Engineering Earthquake Research Center (1994)

Estimation of the Future Earthquake Situation by Using Neural Networks Ensemble

Tian-Yu Liu[1,2], Guo-Zheng Li[1,2,⋆], Yue Liu[1], Geng-Feng Wu[1], and Wei Wang[3]

[1] School of Computer Engineering and Science, Shanghai University, Shanghai 200072, China
[2] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China
[3] Earthquake Administration of Shanghai Municipality, Shanghai 200062, China
gzli@staff.shu.edu.cn

Abstract. Earthquakes will do great harms to the people, to estimate the future earthquake situation in Chinese mainland is still an open issue. There have been previous attempts to solve this problem by using artificial neural networks. In this paper, a novel algorithm named MIFEB is proposed to improve the estimation accuracy by combing bagging of neural networks with mutual information based feature selection for its individuals. MIFEB is compared with the general case of bagging on UCI data sets, then, MIFEB is used to forecast the seismicity of strong earthquakes in Chinese mainland, computation results show that MIFEB obtains higher accuracy than other several methods like bagging of neural networks and single neural networks do.

1 Introduction

Earthquake forecasting is one of the challenging problems all over the world, where estimation of the future earthquake situation in Chinese mainland is an interesting topic especially for the Chinese researchers [1]. It is difficult to objectively give an overall and accurate estimation, because many direct factors arousing earthquakes cannot be measured, at the same time, the relationship between earthquakes and the observed precursors is strongly non-linear. In this paper, the relationship between the seismicity of strong earthquakes in Chinese mainland and those occurred at plate boundaries in the world will be studied, on which models will be built to forecast the the future strong earthquake situation in Chinese mainland.

Since artificial neural networks is a powerful modeling method, it has been widely used in the earthquake forecasting field. In the previous works on the topic of estimation of the future strong earthquake situation in Chinese mainland, Wang et al. [1] have applied single multi-layer perception neural networks to give an estimation. Nowadays, Ensemble learning is becoming an ad hoc topic

⋆ Corresponding author.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1231–1236, 2006.

in the neural networks community [2], which has been widely used to improve the generalization performance of single learning machine. Neural networks ensemble has been applied to the comprehensive earthquake forecasting and obtained satisfying results [3]. Therefore, using the bagging [4] of neural networks to estimate the situation of strong earthquakes in Chinese mainland is an interesting issue. Yet, there are irrelevant and redundant features in the data sets of earthquake estimation, In order to improve the accuracy of bagging of neural networks, a novel algorithm of MIFEB is proposed to use the feature selection methods [5, 6] to improve the prediction accuracy.

The rest of this paper are arranged as follows: In Section 2, the used data set for earthquake estimation is briefly described. In Section 3, MIFEB, a filter feature selection model with the mutual information criteria for bagging, is described in details. In Section 4, MIFEB is compared with the general bagging method on UCI data sets, and then, MIFEB is employed to estimate the strong earthquake situation in Chinese Mainland. At last, conclusions will be given in Section 4.

2 The Data Set for Estimation of Earthquake Situation

It is well known that the strong earthquakes in the world occurred mainly on the circum-Pacific zone and the Eurasian zone. According to the distribution of the strong earthquakes and the plates boundaries, the world can be divided into 16 districts of strong earthquakes as in Figure 1, where the digits are used to distinguish different districts, and all the districts are around the circum-Pacific and Eurasian zones.

The number of strong earthquakes Ms≥7.0 in one district is used as one feature, there are 16 districts, so there are 16 features for one instance of one

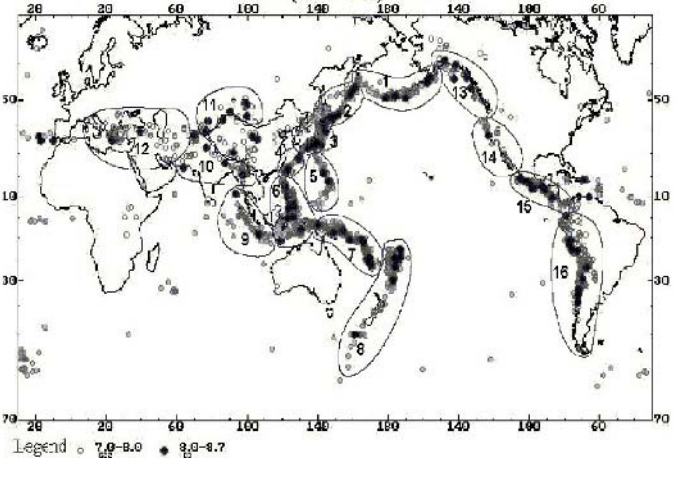


Fig. 1. The divided zones of strong earthquakes in the world

year. Since some records of the strong earthquakes in the world are not complete, the instances from 1925 to 2003 are collected as the input, the number of whole cases is 79. The output is a binary value, which is true if there occur strong earthquakes Ms≥7.0 in the next year in Chinese mainland, and false, otherwise. The real output values are collected from 1926 to 2004.

The data set is collected from several earthquake catalogues published by the government including 1) The Chinese ancient catalogues of strong earthquakes; 2) The Chinese current catalogues of strong earthquakes; 3) The global catalogues of strong earthquakes. Detailed information can be found in [1].

3 Computational Methods

The estimation problem of strong earthquakes has been treated by using single neural networks [1]. Nowadays, ensemble learning is a state of arts technique. Yet, there are redundant features in the earthquake data sets which will hurt the performance of learning machines. To remove the irrelevant features for the individuals of the used ensemble learning method, we propose a novel algorithm MIFEB (Mutual Information based Feature sElection for Bagging) which uses the filter feature selection method with the mutual information criteria for bagging of neural networks.

Mutual information describes the statistical dependence of two random features. The mutual information between two features R and S can be defined in terms of the probability distribution of intensities as follows:

$$I(R:S) = \sum_{r\in R}\sum_{s\in S} p\{r,s\} \lg \frac{p\{r,s\}}{p\{r\}p\{s\}}, \tag{1}$$

where $p\{r,s\}$ is the combined probability distribution of intensities of two features R and S. $p\{r\}$ and $p\{s\}$ respectively are the individual probability distribution of intensities of R and S. The mutual information criteria has been widely used as the filter feature selection model [7].

Algorithm MIFEB

Suppose $T_r(x^1, x^2, \cdots, x^D, C)$ is the training set and p is the number of individuals of ensemble.

T_r and p are input into the procedure and ensemble model L is the output.

Step 1. Generate a training subset T_{rk} from T_r by using Bootstrap sampling algorithm, the size of T_{rk} is three quarters of the size of T_r.

Step 2. Employ the mutual information formula (1) on the training subset T_{rk} and the target set of T_{rk} to obtain the value vector MI.

Step 3. Rank the vector MI in descending order and sum MI to obtain $SUM(MI)$.

Step 4. Select all of the first features as the optimal subset, whose total values should be greater than $RMI * SUM(MI)$. RMI is a pre-defined ratio which is greater than 0 but less than 1.

Step 5. Generate the optimal training subset $T_{rk-optimal}$ from T_{rk} according to the optimal features obtained in Step 4.
Step 6. Train the individual model L_k on the optimal training subset $T_{rk-optimal}$ by using neural networks.
Step 7. Repeat from Step 2 to Step 6 until p models are set up, p is 20 in this paper.
Step 8. Ensemble the obtained models L by the way of majority voting method.

4 Computational Results

In this section, the novel algorithm of MIFEB will be validated on the UCI data set, and then applied to the estimation of earthquake situation.

4.1 Experiments on UCI Data Sets

Eight data sets are selected from the UCI machine learning repository [8] and listed in Table 1, in which the number of instances ranges from hundreds to thousands and the number of features ranges from 9 to 35. To make them suitable for our algorithms, the nominal values are changed to be numerical in all data sets. Then, all the attributes are transformed into the interval of [-1, 1] by an affine function.

Table 1. The properties of the used UCI data sets

Data set	Number of classes	Number of features	Number of instances
all-hyper	5	29	3772
backup	19	35	683
breast-cancer-W	2	9	699
glass	6	9	214
proc-C	5	13	303
proc-H	2	13	294
soybean-l	19	34	307
statlog-h	2	13	270

The hold out method is used to validate the results. Experiments will be repeated fifty times on each data set. In this paper, we use an improved multi-layer perception neural networks, weight decay based neural networks in Bayesian frame, which adds a regularized term to the objective function and is insensitive to the setting of parameters to some degree [9]. The parameters for the neural networks are the same on all data sets, namely the number of hidden nodes is 10. The number of individuals for bagging is 20 and The RMI for mutual information algorithm is 0.9 in our experiment.

Experimental results of the accuracy obtained by the different bagging methods are shown in Table 2, from which we can see that the mean accuracy values obtained by MIFEB are improved in various degree on different data sets which ranges from 1% to 6%, the average value improved on all the data sets is 3.1 percent. At the same time, the values of standard deviation are also reduced in some degree.

Table 2. Statistical prediction accuracy on the UCI data sets

Data set	MIFEB	Bagging
all-hyper	97.93 ± 0.16	96.76 ± 0.43
backup	91.47 ± 1.62	89.90 ± 2.06
breast-cancer-W	94.01 ± 0.92	91.23 ± 1.71
glass	65.78 ± 4.01	61.75 ± 5.12
proc-C	54.16 ± 2.40	49.93 ± 3.65
proc-H	79.70 ± 2.71	73.89 ± 3.52
soybean-l	84.30 ± 3.38	83.25 ± 3.57
statlog-h	78.98 ± 2.87	74.88 ± 3.87
Average	80.79 ± 2.25	77.69 ± 2.99

4.2 The Earthquake Situation Forecast

For the estimation problem of the strong earthquake situation, to validate MIFEB, and compare MIFEB with bagging of neural networks and single neural networks, 4 instances of the last four years are selected out of all 79 instances as the test sample, the other 75 instances are used as the training sample, where each instance has 16 features. The parameters for the three algorithms are set as: the number of hidden nodes is 8, the number of individuals for bagging is 7 and the RMI for mutual information algorithm is 0.9. Computational results on the test sample by using three different neural networks based learning methods are shown in Table 3, where the highest level of earthquakes occurred in Chinese mainland in that year is listed in the 2nd row, the label is determined as in Section 2 and listed in the 3rd row.

From Table 3, it can be seen that both MIFEB and bagging of neural networks could obtain 100% accuracy on the four instances, while single neural networks misclassified one instance.

Since the number of the total instances is small, only 4 are selected as the test sample, to validate the statistical performance of the three learning methods, the computation is repeated 20 times, the average estimation accuracy of MIFEB, bagging of neural networks and single neural networks are 90%, 85%, and 75% respectively, which shows that MIFEB could build the best model on the relationship of earthquakes occurred between in Chinese mainland and the world.

Table 3. Results on the test sample of the earthquake situation data set

Year	2001	2002	2003	2004
Highest Level	8.1	5.9	6.8	6.7
Label	1	0	0	0
Single	1	0	1	0
Bagging	1	0	0	0
MIFEB	1	0	0	0

5 Conclusions

To improve the estimation accuracy of the strong earthquake situation in Chinese mainland, a novel algorithm of MIFEB (Mutual Information based Feature sElection for Bagging) is proposed in this paper. Experimental results on UCI data sets and the earthquake situation data set show that MIFEB obtained better results than bagging of neural networks and single neural networks did. From the view of the estimation of earthquake situation, the computational results imply that there are strong nonlinear relationship between the earthquakes in Chinese mainland and the world, and this relationship can be modeled by these neural networks based methods, of which MIFEB is the best method. From the view of learning methods, the computation results imply that there are redundant features in the earthquake and other real world data sets, and feature selection can really improve the accuracy of bagging of neural networks.

Acknowledgement

This work was supported by NSFC (20503015) and SHMEC NSF (05AZ67). The authors would like to thank the anonymous reviewers for their valuable advices.

References

1. Wang, W., Cao, X., Song, X.: Estimation of the Earthquakes in Chinese Mainland by Using Artificial Neural Networks (in Chinese). Chinese Journal of Earthquakes **3(21)** (2001) 10–14
2. Dietterich, T.: Machine Learning Research: Four Current Directions. The AI Magazine **18** (1998) 97–136
3. Liu, Y., Li, Y., Li, G.Z., Zhang, B., Wu, G.: Constructive Ensemble of RBF Neural Networks and Its Application to Earthquake Prediction. In: Wang, J., Liao, X., Yi, Z. (eds): Advances in Neural Networks - ISNN 2005. Lecture Notes on Computer Science, vol. 3496, Springer-Verlag, (2005) 532–537
4. Breiman, L.: Bagging Predictors. Machine Learning **24** (1996) 123–140
5. Kohavi, R., George, J.H.: Wrappers for Feature Subset Selection. Artificial Intelligence **97** (1997) 273–324
6. Guyon, I., Elisseeff, A.: An Introduction to Variable and Feature Selection. Journal of Machine Learning Research **3** (2003) 1157–1182
7. Liu, H., Yu, L.: Toward Integrating Feature Selection Algorithms for Classification and Clustering. IEEE Trans. on Knowledge and Data Engineering **17** (2005) 1–12
8. Blake, C., Keogh, E., Merz, C.J.: UCI Repository of Machine Learning Databases. Technical Report, Department of Information and Computer Science, University of California, Irvine, CA (1998) http://www.ics.uci.edu/ mlearn/MLRepository.htm.
9. Foresee, F.D., Hagan, M.T.: Gauss-newton Approximation to Bayesian Regularization. In: Proceedings of the 1997 International Joint Conference on Neural Networks. (1997) 1930–1935

An Expert System Based on BP Neural Networks for Pre-splitting Blasting Design

Xiaohong Li[1], Xinfei Wang[1], Yongkang Dong[2], Qiang Ge[1], and Li Qian[1]

[1] Key Lab of Ministry of Education for the Exploitation of Southwest Resources & the Environmental Disaster Control Engineering, College of Resources & Environmental Sciences, Chongqing University, Chongqing 400030, China
[2] North Construction Branch, Chongqing Expressway Co. Ltd, Chongqing 400042, China
xhli@cqu.edu.cn, wangxinfei781231@163.com, yongkd@sohu.com, haorenhaoxin@126.com, gogo-1212@163.com

Abstract. The issue of pre-splitting blasting design is investigated in this paper. According to the principle of expert system and neural networks, Visual C++ 6.0 and SQL Sever 2000 are employed to develop hybrid expert system of pre-splitting blasting design based on BP neural networks. The proposed expert system is a coupling system of engineering database and three-layered BP neural networks, which can be applied into cutting excavation of the expressway. The experiments show that the system can enhance the reliability of in-site pre-splitting blasting design scheme, efficiency and quality of construction.

1 Introduction

Landform of China's western region is complex with high and steep mountains. The expressway to be constructed must pass through lots of massif, and the quantity of cutting excavation is great [1]. Taking optimum pre-splitting blasting design scheme in the different geological conditions is a key problem in construction of expressway in mountainous area [2]. The main goal of optimization of pre-splitting blasting design scheme is to reduce the cost and get the better effect by adjusting parameters and method [3]. So far most of the design schemes for pre-splitting blasting are obtained by practical experience, thus it restricts the development of pre-splitting blasting technology and its application in engineering, while the development of computer technology, artificial intelligent technology, application of expert system and neural networks accelerates the development of pre-splitting blasting technology. According to the principle of expert system and neural networks, it is feasible to develop hybrid expert system of pre-splitting blasting design based on BP neural networks.

2 Framework of Expert System of Pre-splitting Blasting Design (BDES) for Cutting Excavation of Expressway

An expert system is a computer program that mimics the human reasoning process, which relies on logic, belief, rules of thumb, opinion, and experience. Unlike Artificial Neural Networks, Expert Systems suffer from serious limitations, mainly their

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1237 – 1241, 2006.

hypersensitivity to incomplete and noisy data. Moreover, some human knowledge cannot be expressed explicitly in terms of rules. These and several other limitations have stimulated the exploration of Artificial Neural Networks for data modeling. Expert Systems work in sequential manner, whereas Artificial Neural Networks are parallel data manipulators, and sometimes viewed as a reverse approach to Expert Systems. Other differences relate to information processing, connectivity, self-learning ability, fault tolerance, and relation to neurobiology. In order to capture the desirable features of both systems, Expert Systems and Artificial Neural Networks are integrated into one hybrid system. [4]

The BDES is also a hybrid system, consists of user interface, infer processor, blasting design system, dynamic database, neural network system and engineering database management system, as shown in Fig.1 [1]

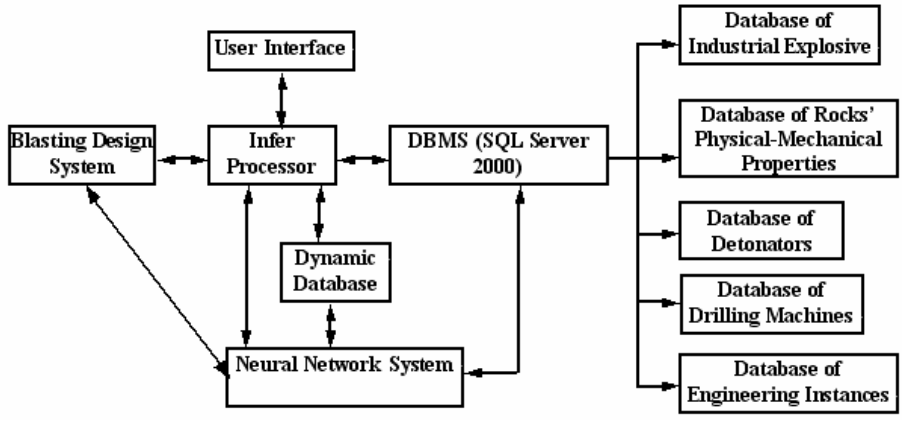


Fig. 1. System framework of BDES

Medsker has explained the five different hybrid strategies. These are: (1) stand-alone models; (2) transformational models; (3) loose coupling models;(4) tight-coupling models; and (5) fully integrated models. [5] The BDES used loose-coupling strategy. And BDES communicate through the dynamic database and data files.

Vassiliou et al. and Jarke and Vassiliou have described four ways to integrate expert systems and databases: (1) elementary data management within ES; (2) generalized data management within ES;(3) loose coupling of ES and existing DBMS; and (4) tight coupling of the ES and existing DBMS. [5] The engineering database is linked to the BDES by loose coupling. The Engineering database consists of database of industrial explosive, database of physical-mechanical properties of rocks, database of detonators, database of drilling machines and database of engineering instances.

The main purpose of BDES is to provide optimized parameters and related information to engineer manager and technician, to accumulate the knowledge and engineer instances, and to make the management and design scheme of pre-splitting blasting more efficiently.

3 Neural Network System

According to the character of pre-splitting blasting design, the typical three-layered BP network is selected here. Structure of three-layered BP network is shown in Fig.2.[6]

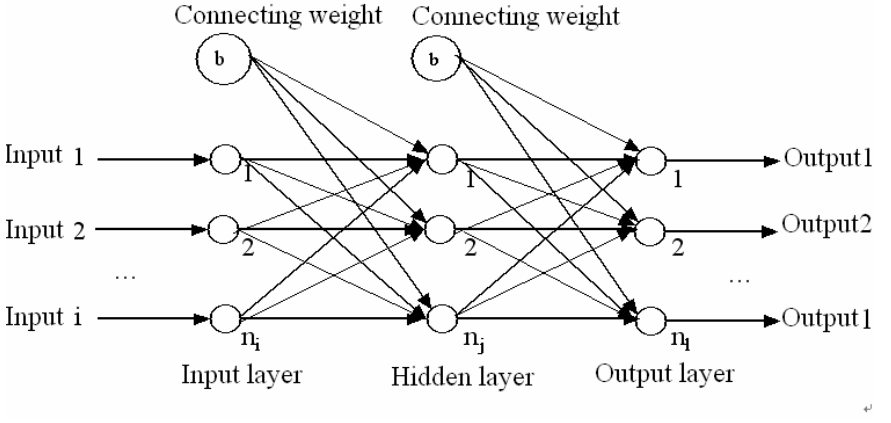


Fig. 2. Structure of three-layered BP network

The neural network of BDES is trained by the engineering instance. And so the practical experience knowledge is dug out and stored in the network. Accumulations of engineering instance enhance accumulation of experience knowledge on pre-splitting blasting technology.

4 Engineering Application of BDES

BDES is used to carry on the field experiment in Qinggangbang Slope of the Yuwan Expressway. Under the typical blasting condition (as listed in Table 1) of this slope, BDES optimizes the pre-splitting blasting parameters (as shown in Table 2), and predicts the blasting result (see, Table 3) under the typical blasting condition and the optimized the pre-splitting blasting parameters[1].

Table 1. Typical condition of blasting design in Yuwan Expressway

Time	2003-3-24
Structure of Rock mass	Integrated Structure
Crack growth degree	Light growth degree
Number of free face	2
Water-contained situation of Rock mass	No water
Type of Rock	Sandstone
Type of explosive	No.2 AN rock explosive
Type of Drilling Machine	YT-20 Drilling Machine
Geometrical parameters of excavation	$10.0\times5.0\times2.0m^3$

Table 2. Parameters for typical condition of blasting design in Some Freeway

	Type of blast-holes	Pre-splitting hole	Cushioning hole	Main blasting hole
Parameters of blasting-holes	Num.	16	9	36
	Diameter /mm	42	42	42
	Length /m	2.2	2.2	2.2
	Hole space /m	0.61	1.00	1.10
	Row space /m	0.72	0.95	1.05
	Loading quantity /kg	0.25	0.68	0.78
	Loading length /m	0.25	0.68	0.80
	Filling length /m	0.28	0.61	0.72
	Loading destiny /kg·m^{-1}	1.00	1.01	0.97
	Uncoupling coefficient	1.68	1.68	1.18
Detonating parameters	Minimum delay time between pre-splitting hole and cushioning-hole /ms			75
	Delay time /ms			30

Table 3. Blasting effect predictions

Average degree of fragment /cm	62.25
Block rate/%	1.87
Toe rate/%	2.15
Half holes rating/%	84.32
Levelness of slope/cm	3.77
Width of pre-splitting crack /cm	5.04
Connectivity of pre-splitting crack /%	97.5

According to the comparison between calculating results and field experiment results, the blasting parameters optimized by BDES is valid.

5 Conclusions

The expert system for pre-splitting blasting design has been developed and applied to cutting excavation of the expressway. The proposed expert system is a coupling system of neural network and engineering database. It is worth noting that engineering database of the system should be adjusted in the pre-splitting blasting experiment, and the neural network must be trained by the pre-splitting blasting engineer instances of the engineering database. An example has been given to show the validity of the developed system.

Acknowledgement

This work was funded by National Natural Science Foundation Project (No. 50334060).

References

1. Wang, X.F.: Research on Pre-Splitting Blasting Design Expert System and Its Application in Cutting Excavation of Expressway. Unpublished Master Dissertation, Chongqing University (2003)
2. Kang, H.M., Li, X.H.: The Application of Pre-splitting Blasting to Excavation on High Slope in Soft Rock Section. Journal of Chongqing University 1 (2001) 47-50
3. Qiu, X.D., Wang, X.F.: Application of Pre-splitting Blasting Technology in the Cutting Slop. Underground Space 1 (2003) 1–3
4. Basheer, I.A., Hajmeer, M.: Artificial Neural Networks: Fundamentals, Computing, Design, and Application. Journal of Microbiological Methods 43 (2000) 3–31
5. Sonar, R.M.: Integrating Intelligent Systems Using an SQL-Database. Expert Systems with Applications 17 (1999) 45–49
6. Wang, X.F., Li, X.H.: Application of BP Neural Network into Prediction of Rockburst in Tunneling, Progress in Safety Science and Technology, Vol. 4. Science Press, Beijing New York (2004) 617–621

Surface Reconstruction Based on Radial Basis Functions Network

Han-bo Liu[1], Xin Wang[2], Xiao-jun Wu[1], and Wen-yi Qiang[3]

[1] Department of Control and Mechatronics Engineering,
Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, P.R. China
liu_hanbo@hit.edu.cn
wuxj@hitsz.edu.cn
[2] Department of Mechanical Engineering and Automation,
Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, P.R. China
wang_xin@hit.edu.cn
[3] Department of Control Science and Engineering,
Harbin Institute of Technology, Harbin, P.R. China

Abstract. A new method for arbitrary 3d-object reconstruction in unknown environment is proposed in this paper. The implicit surface is reconstructed based on radial basis functions network from range scattered data. For the property of locality of radial basis function, the method is fast and robust with respect to large data. Furthermore, an adapted K-Means algorithm is used to reduce RBF centers for reconstruction. Experiment results show that the presented approach is helpful in speed improvement and is a good solution for large data reconstruction.

1 Introduction

It is long cherished that arbitrary 3D objects can be reconstructed fast and completely from scattered point data from calibrated cameras or laser scanner, since Boissonnat first addressed in the mid-1980s [1]. Additionally, rapidity and robustness to noises are essential to industrial application. But most of the reconstruction systems remain in need of several minutes or hours to get a 3D model.

One solution is to cut the data without a loss in accuracy and object features. Some of researchers, such as Carr [2], Beatson [2] and Schall [3], believe that the same input data may be able to be approximated to the desired accuracy using significantly fewer centers. That means dense laser scanned data can be represented by significantly fewer centers than the total number of data points. [4] mentioned curtly that surface reconstruction results can be drastically improved if an appropriate preliminary filtering and denoising method is applied to original scattered data. But there is still not a universal and effective method to reduce the centers now.

In this paper, we propose a fast method in which an adapted K-Means algorithm is used to reduce the number of reconstruction centers. Furthermore, the RBF neural network is brought into the 3D reconstruction field.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1242 – 1247, 2006.

2 Reconstructing Surface with RBF Network

There are two main approaches to reconstruct surfaces from 3D scattered data. One is Delaunay-based method, which involves the computation of a Delaunay complex or of dual structures using the scattered data. But a drawback is geometry method can not deal with noisy and under-sampled data. Some researchers [5, 6] now focus on this area. The other is volumetric method, which approximates the scattered data by a scalar function [7, 8]. Partition of Unity (PU) [9] and Moving Least Squares (MLS) are much popular methods now. For the sake of industrial application where noises are unavoidable, we use PU+RBF network reconstruction method based on the latter.

2.1 Surface Representation

In general, surfaces can be expressed in discrete, implicit, or explicit form [10]. Discrete form, such as a collection of polygons and sampled points, can approximate smooth surfaces, but is verbose and has fixed resolution. Let $f : U \subset R^2 \to R$ .The explicit form of surface is the subset consisting of points $(x_1, x_2, f(x_1, x_2))$ for $(x_1, x_2) \in U$ an open subset of R^2 .The implicit surface in R^3 is expressed as $f : R^3 \to R \mid f(x, y, z) = constant$. As a general rule, the constant is equal to zero. Moreover, complicated editing operations are easy to do on implicit surface, such as union, difference and intersection. Implicit form is usually adopted for surface reconstruction now [11] [12]. Our aim is to model the whole object with a single implicit function which is continuous and differentiable.

2.2 RBF Network

The implicit surface S is defined by a zero level set of radial basis functions network $F(x)$, which is greater than zero outside the surface and less than zero inside the surface. It is expressed in the following form:

$$F(x) = \sum_{i=1}^{n} w_i \phi(X, c_i, r_i) + P_k(x). \tag{1}$$

Where $\phi(x)$ is a basis function, $X = (x, y, z)$ is the input vector, c_i and r_i are the centers and radii of the basis, $P_k(x)$ is a degree k polynomial for affine transform, n is the number of centers and w_i are unknown weight coefficients to be determined.

The interpolant will be chosen from $BL^{(2)}(R^3)$, the Beppo-Levi space of distributions on R^3 with square integrable second derivatives. So ω_i must satisfy the orthogonality condition,

$$\sum_{i=1}^{n} \omega_i = \sum_{i=1}^{n} \omega_i x_i = \sum_{i=1}^{n} \omega_i y_i = \sum_{i=1}^{n} \omega_i z_i = 0. \tag{2}$$

So we can easily regard the form as summing a collection of weighted radial basis functions. The RBF network model of the surface is shown in Fig. 1.

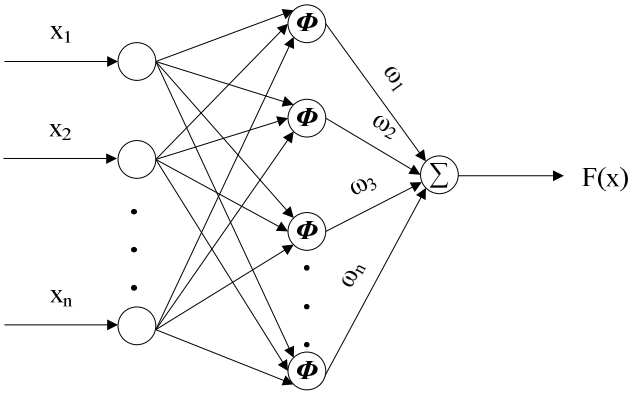


Fig. 1. RBF neural network model of implicit surface function

The first layer is composed of input nodes, whose number is the number of interpolation centers; the hidden node is nonlinear radial basis function; the output layer is a single linear weighted sum of the output of hidden node.

The basis function can be not only non-compactly supported functions, such as the thin-plate spline $\phi(r) = r^2 \log(r)$, the Gaussian $\phi(r) = \exp(-cr^2)$, the multiquadric $\phi(r) = \sqrt{r^2 + c^2}$ and harmonic spline $\phi(r) = r^3$ [2], but also can be compact supported functions proposed by Wendland, Wu or Buhmann. We select the triharmonic spline as the basis function in our study.

To accelerate the training and converging speed of RBF network for complex objects, we use PU method to divide the global data set into several smaller overlapping subdomains and SVD approach to solve equations together. We can only use SVD method to obtain the network weights quickly in some small subdomains.

3 K-Means Algorithm for RBF Center Reduction

For construction from large scanned data, it is in great need of centers reduction. We adapt the K-Means algorithm to reduce the number of centers, which is described in details below.

3.1 K-Means Algorithm

K-Means is a classical algorithm for clustering [13]. But it doesn't completely suit RBF reconstruction since the centriod condition points may not locate on the surface. Therefore, we adapt the K-Means algorithm of RBF network for reconstruction.

3.2 Adapted K-Means Algorithm

Let fix the notation in common use: $D = \{x_i\}_{i=1}^{l}$ be a data set with $x_i \in R^3$, $W = \{w_k\}_{k=1}^{K}$ with $w_k \in R^3$ be a clustered set, and $K << l$. The Voronoi region R_k of the clustered center w_k is the set of all vectors in R^3 for which w_k is the closest vector. The Voronoi set V_k of the clustered center w_k is the set of all vectors in D.

$$R_k = \{x \in R^3 \mid k = \arg\min_{j=1\ldots K} \| x - x_j \|\}.$$

$$V_k = \{x_i \in D \mid k = \arg\min_{j=1\ldots K} \| x_i - x_j \|\}.$$

The adapted K-Means algorithm can be stated as fellows:

- Step1: Initialize the clustered set W with k clustered centers. Instead of randomly selecting, we keep the distance between initial centers greater than L. Results show that the distribution quality of clustered centers is improved greatly.
- Step 2: Compute the Voronoi set V_k of each center w_k.
- Step 3: Move each center to the mean of V_k.
- Step 4: Return the centers w_k if no vector is changed in step 3. Otherwise, go to step 2.
- Step5: Select the vector $x \in D$ as the clustered center, where x is nearest to w_k. This can ensure the computed cluster centers all on the surfaces.

4 Results

Fig.2 shows the duck reconstruction from 302 scattered points in comparison between only in RBF method and in the adapted K-Means RBF method. The left picture is the 302 centers (top row) and clustered 178 centers (bottom row) used in reconstruction from 302 scattered points; the middle one is the reconstruction from 302 centers only in the RBF network method; the right is from clustered 178 centers in the adapted K-Means RBF method.

Fig.3 shows the reconstruction of sculpture and squirrel models from scattered points in two methods based on RBF network. The top left sculpture model is constructed from scattered 25,382 centers while the top right is constructed from clustered 15,139 centers in the adapted K-Means RBF method from the same input data; the bottom left squirrel model is reconstructed from scattered 40,627 centers while the bottom right is constructed only from clustered 20,234 points in the adapted K-Means RBF method. Results demonstrate that the adapted K-Means RBF network method can reconstruct not only the complex objects, but also preserve the details well.

Objects are usually composed by simple shapes in industrial field, such as sphere, torus and cube. The reconstruction results show that the number of reconstructing centers can reduce fewer than half and the models are identical with the sampled objects in the adapted K-Means RBF network method.

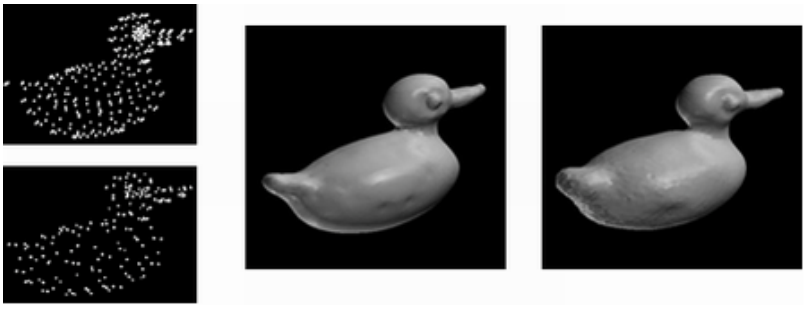


Fig. 2. The duck reconstruction in RBF method and the adapted K-Means RBF method

Fig. 3. Sculpture and squirrel construction in RBF and the adapted K-Means RBF method

5 Conclusions

Our reconstruction method provides us a simple and robust method to create 3D models. Moreover, we propose an adapted K-Means RBF algorithm for subdomains to reduce the interpolation centers, in contrast to the conventional RBF reconstruction methods, where all data are selected as centers so that the speed can not satisfy the rapidity requirement in industry. Experimental results show that our method reduces

the quantity of centers greatly and saves time substantially without loss of reconstruction precision. For the real-time purpose, we will focus more on the probability distribution of the data to reduce centers while preserving the geometry details well in the future work.

References

1. Boissonnat, J.D.: Geometric Structures for Three-dimensional Shape Representation. ACM Trans. Graphics. 3(4) (1984) 266-286
2. Carr, J.C., Beatson, R.K., Cherrie, J.B., et al.: Reconstruction and Representation of 3D Object with Radial Basis Functions. Proceedings of SIGGRAPH. ACM Trans. Graphics, (2001) 67-76
3. Isgro, F., Odone, F., Saleem, W., Schall, O.: Clustering for Surface Reconstruction. 1st International Workshop on Semantic Virtual Environments. (2005) 156-162
4. Schall, O., Samozino, M.: Surface from Scattered Points: a Brief Survey of Recent Developments. 1st International Workshop on Semantic Virtual Environments. (2005) 138-147
5. Dey, T.K. and Goswami, S.: Provable Surface Reconstruction from Noisy Samples. In: Jack, S., Jean-Daniel, B. (eds.): Proceedings of 20th ACM Symposium on Computational Geometry. ASM2004, (2004) 330-339
6. Kolluri. R., Shewchuk, J.R., and O'Brien, J.F.: Spectral Surface Reconstruction from Noisy Point Clouds. In: Jack, S., Jean-Daniel, B. (eds.): Proceedings of 20th ACM Symposium on Computational Geometry. ACM2004, (2004) 11-21
7. Tobor, I., Reuter, P., Schlick, C.: Multi-Scale Reconstruction of Implicit Surfaces with Attributes from Large Unorganized Point Sets. Proceedings of SMI. IEEE Computer Society, (2004) 19-30
8. Ohtake, Y., Belyaev, A., Seidel, H.: A Multi-scale Approach to 3D Scattered Data Interpolation with Compactly Supported Basis Functions. Proceedings of SMI. IEEE Computer Society, (2003) 153-161
9. Ohtake, Y., Belyaev, A., Alexa, M., Turk, G., et al.: Multi-level Partition of Unity Implicits. ACM Trans. Graphics. 22(3) (2003) 463-470
10. Bolle, R.M. and Vemuri, B.C.: On Three-dimensional Surface Reconstruction Methods. IEEE Trans. PAMI. 13(1) (1991) 1-13
11. Ohtake, Y., Belyaev, A., et al.: 3D Scattered Data Approximation with Adaptive Compactly Supported Radial Basis Functions. Proceeding of SMI. IEEE Computer Society, (2004) 31-39
12. Turk, G. and O'brien, J.: Modeling with Implicit Surfaces That Interpolate. ACM Trans. Graphics. 21(4) (2002) 855-873
13. Camastra, F. and Verri, A.: A Kernel Method for Clustering. IEEE Trans. PAMI. 27(5) (2005) 801-804

Determination of Gas and Water Volume Fraction in Oil Water Gas Pipe Flow Using Neural Networks Based on Dual Modality Densitometry

Chunguo Jing[1,2], Guangzhong Xing[1], Bin Liu[1], and Qiuguo Bai[2]

[1] Yanshan University, Qinhuangdao 066004, China
jingchunguo@163.com
[2] Northeastern University at Qinhuangdao, Qinhuangdao 066004, China

Abstract. The models of dual modality densitometry were developed. It can be used for measuring the gas volume fraction and water volume fraction in oil water gas pipe flow. The models are complex. In order to solve models, it often uses simplified models. This reduces measurement precision. The method of measuring gas and water volume fraction using neural networks was presented. The simulation data was gotten using Geant4. The radial basis function networks were trained and tested on computer simulation data. The results show that networks predicted gas volume fraction fit true gas fraction well and water volume fraction has some deviations.

1 Introduction

It is important for petroleum industries to measure gas volume fraction (GVF) and water volume fraction (WVF) in pipe flow. Many methods can be used to measure GVF and WVF, such as capacitance, microwave and gamma radiation etc[1]. The gamma ray has higher measuring precision for it can not be affected by the oil packed in water or water packed in oil. The dual-energy gamma densitometry and dual modality densitometry (DMD) are two main methods. The dual-energy desitometry[2] is easily modelled and can be used to calculate GVF and WVF directly from models, but it is difficult to select a suitable dual-energy source. The principle of dual modality densitometry is to detect the transmitted and scattered radiation intensities, but modelling of scattered radiation is far more complex. In general, it adopts a simplified model that leads to lower precision. The methods to predict GVF and WVF using neural networks were presented, which saves the trouble of solving models.

2 The Principle of Dual Modality Densitometry

The measurement GVF and WVF using DMD has been studied by many researchers. Tjugum S. A. and Johansen G. A. in their paper[3] have developed a simplified model and multipile voxel model for scattered radiation. Figure 1 gives a simplified measurement geometry.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1248 – 1253, 2006.

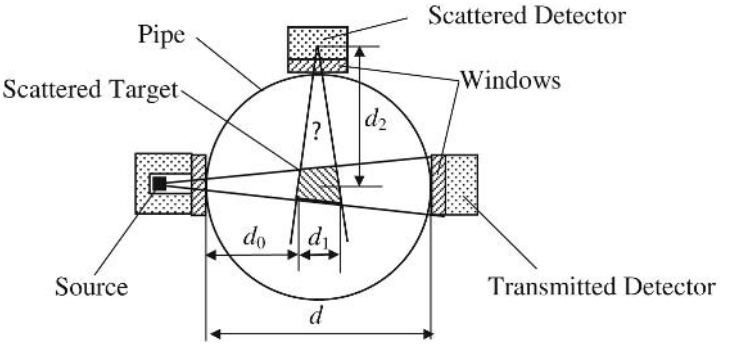


Fig. 1. A simplified measurement geometry

In figure 1, the gamma source is ^{241}Am. The diameter of pipe is d. The solid angle which the scattered detector is to the scattered target is θ. The average width of scattered target is d_1. The distance from the center of scattered target to scattered detector is d_2. The distance from the center of scattered target to the window of gamma source is d_0. The fluid in pipe is the mixture of oil, water and gas and the GVF is α. According to the theory of gamma ray attenuation, the intensities detected by the transmitted detector and the scattered detector are

$$I_t = \varepsilon I_0 e^{-\mu(1-\alpha)d} \tag{1}$$

$$I_s = \varepsilon \frac{\Omega(\theta)}{4\pi} I_0 \frac{\mu_\sigma}{\mu}(1-e^{-\mu(1-\alpha)d_1})e^{-\mu(1-\alpha)d_0}e^{-\mu'(1-\alpha)d_2} \tag{2}$$

where I_0 is the intensity of incident radiation, ε is efficiency factor of detectors, μ and μ' are the linear attenuation coefficients of the mixture flow at the energy of incident and scattered radiation respectively. The radiation energy is reduced in the Compton scattering process, and the attenuation coefficient of scattered radiation is thus higher. $\Omega(\theta)$ is solid angle of the scattered detector to scattered target. Because the linear attenuation coefficient of gas is too small, formulae (1) and (2) does not calculate the effect of attenuation of gas.

The generation of scattered radiation from scattered target is

$$\frac{\mu_\sigma}{\mu}(1-e^{-\mu(1-\alpha)d_1}) \tag{3}$$

Where μ_σ is the linear attenuation coefficient of Compton scattering. The attenuation of radiation from the source to the scattered target is

$$I_0 e^{-\mu(1-\alpha)d_0} \tag{4}$$

The attenuation from the scattered target to the scattered detector is

$$e^{-\mu'(1-\alpha)d_2} \tag{5}$$

It is difficult to calculate GVF and WVF from formulae (1) and (2). The simplified model was used to measure GVF and WVF, but these methods enlarge the errors.

3 Simulation Geometries

In order to study the measurement GVF and WVF using DMD, the computer simulation models were defined by using Geant4 software[4]. Geant4 is a toolkit for the simulation of the passage of particles through matter and was developed by CERN. Geant4 simulate particles interaction with matter through Monte Carlo methods. A simulation geometry constructed by Geant4 shows in figure 2.

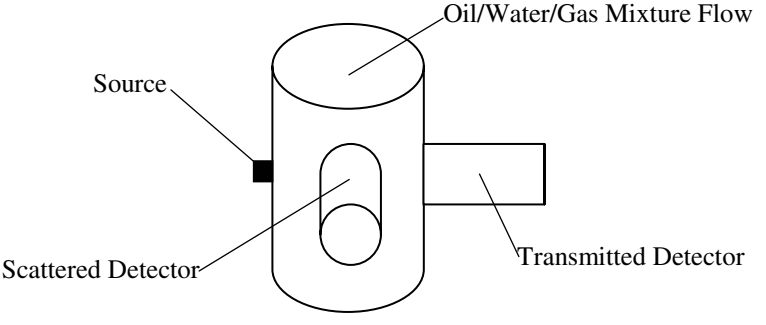


Fig. 2. The simulation geometry constructed by Geant4

In figure 2, the diameter of pipe is 100mm, the detectors are the Φ40mm×40mm NaI crystal. Oil is cetene (molecular formula, $C_{16}H_{34}$, density, 0.7733g/cm^3) instead of crude oil, water's molecular formula is H_2O and density is 1.0g/cm^3, gas is methane (molecular formula, CH_4, density, 0.717mg/cm^3) intead of natural gas. The radiation source energy is 59.5keV. The numbers of simulation event are 100,000. The flow in pipe is the mixture of oil, water and gas by giving the fractional mass of each component. The intensities of transmitted radiation and scattered radiation decayed by different mixture matter were recorded by Geant4. The simulation data was used to train and test the radial basis function neural networks.

4 Neural Networks and Results Analyse

It has been demonstrated that a neural network can approach any nonlinear function based on inputs and outputs. It can be applied to the recognition and function approximation of a wide range of situation, pattern and individual features of different systems. Bishop and James[5] have shown that it is possible to determine oil, water and gas fraction in multiphase pipelines using neural networks based on dual-energy densitometry. The aim of this study was to apply the technique to determine GVF and WVF in mixture flow using DMD.

The back-propagation (BP) neural network is the most widely applied neural network technique. In general, the BP neural network encounters local minimum, slow convergence speed and convergence instability. The radial basis function neural

network (RBF) is similar to the BP. It can be trained very quickly because the algorithm uses a fixed Gaussian function. In this paper RBF neural network was used. A RBF neural network architecture used for predicting GVF and WVF is shown in figure 3. The input layer consists of two neurons which are the intensities of transmitted and scattered radiation. The output layer is also two neurons predicting GVF and WVF. The hidden layer nodes are called RBF units, determined by a parameter vector called center and a scalar called width. The Gaussian density function is used in the hidden layer as an activation function. The overall input-output mapping is as follows:

$$y_i = w_0 + \sum_{j=1}^{h} w_{ji} \exp(-\left\| \boldsymbol{X} - \boldsymbol{C}_j \right\|^2 / \beta_j^2) \tag{6}$$

where $\boldsymbol{X}$ is the input vector, $\boldsymbol{C}_j$ is the j[th] center of RBF unit in the hidden layer, h is the number of RBF units, w_o and w_{ji} are the bias term and the weight between the hidden and output layers respectively, and y_i is the i[th] output in the m dimensional space. Once the centers of BRF units are established, the width of the i[th] center in the hidden layer is calculated by (7).

$$\beta_i = \left[\frac{1}{h} \sum_{j=1}^{h} \sum_{k=1}^{n} (\left\| c_{ki} - c_{kj} \right\|) \right]^{1/2} \tag{7}$$

where c_{ki} and c_{kj} are the k[th] value of the center of i[th] and j[th] RBF units.

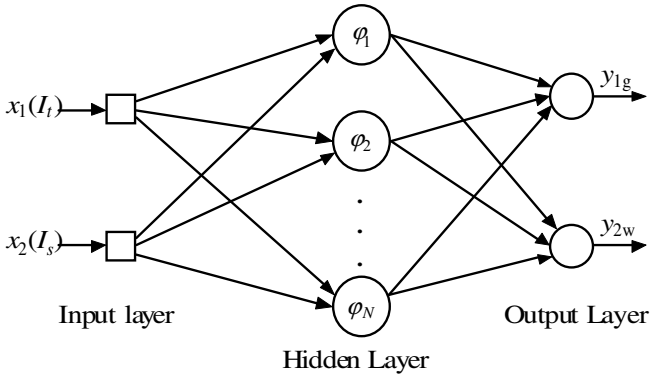


Fig. 3. A architecture of a RBF neural network

The training set and test set were gotten by computer simulation. Figures 4 and 5 are the comparison of the predicting GVF to true GVF and the predicting WVF to true WVF. From figures 4 and 5 it can be seen that the predicting GVF fit true GVF well and the predicting WVF has some errors to true WVF. The reasons are the linear attenuation coefficient of gas is small, when GVF changes, the detected transmitted and scattered radiation intensities change obviously. The water linear attenuation coefficient is close to oil, the features that WVF changes are not obvious. Table 1 is statistical results in the course of trying to predict GVF and WVF. Although the errors between true WVF and the predicting WVF emerge, the mean square error (MSE) is lower.

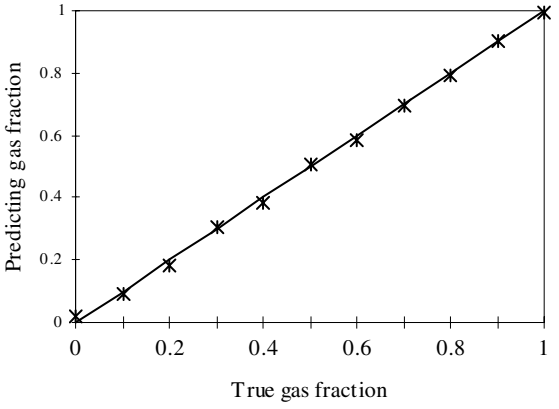


Fig. 4. The comparison of the predicting GVF to true GVF

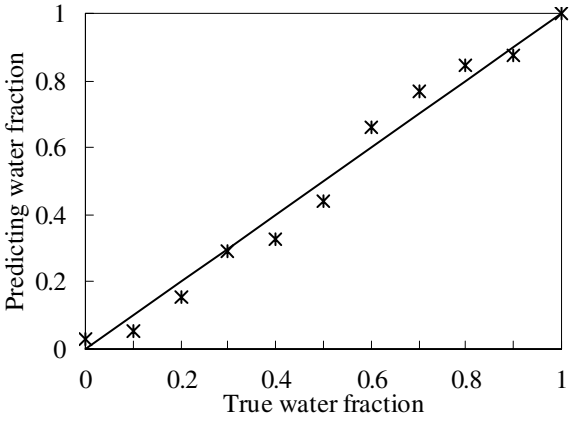


Fig. 5. The comparison of the predicting WVF to true WVF

Table 1. The statistical results of predicting GVF and WVF

Performance	WVF	GVF
MSE	0.001850085	0.000130955
Min Abs Error	0.000539245	0.001850492
Max Abs Error	0.071016457	0.019572849

5 Conclusions

The models of DMD are complex. In order to calculate WVF and GVF in mixture fluid using DMD, the soft-sensing method based on RBF neural networks was presented. The simulation geometries were constructed by using Geant4. The RBF

networks were trained and test on similation data. The predicting results show that GVF is accurate close to true GVF and WVF has some errors. The overall MSE is lower.

References

1. Thorn, R., Johansen, G.A., Hammer, E.A.: Recent Developments in Three-phase Measurement. Measurement Science and Technology 8 (7) (1997) 691-701
2. Bai, Q., Jing, C., Shu, D.: The Mathematical Method in Measuring Water-rate and Gas-rate of Petroleum in Means of γ-ray. Nuclear Electronics and Detection Technology 22(3) (2002) 225-227
3. Tjugum, S.A., Johansen, G.A., Holstad, M.B.: A Multiple Voxel Model for Scattered Gamma Radiation in Pipe Flow. Measurement Science and Technology 14(10) (2003) 1777-1782
4. GEANT4 Collaboration. GEANT4 User's Guide. http://geant4.web.cern.ch/geant4/
5. Bishop, C.M., James, G.D.: Analysis of Multiphase Flows Using Dual-energy Gamma Densitometry and Neural Networks. Nuclear Instruments and Methods in Physics Research A327 (1993) 580-593

Application of Neural Network in Metal Loss Evaluation for Gas Conducting Pipelines

Wei Zhang[1], Jing-Tao Guo[1], and Song-Ling Huang[2]

[1] Department of Mechanical Engineering, Tsinghua University, 100084 Beijing, China
[2] Department of Electrical Engineering, Tsinghua University, 100084 Beijing, China

Abstract. An application based on BP neural network is conducted on evaluating the state of metal loss in gas conducting pipelines. We employ network to quantify the defects of metal loss in outer pipe spool according to the $w_{\mathbf{m}}$, $l_{\mathbf{m}}$ and $h_{\mathbf{m}}$ of MFL (magnetic flux leakage), which are the width, length and depth of the area of abnormal magnetic data. After quantifying, we obtain w, l and d, which are deemed to the width, length and depth of the defect. The data to train our networks are developed from MFL data which are picked by Pipe Pig moving in service pipe with a constant velocity parallel to pipe axis.

1 Classification of Defects in Pipelines

The most frequent cause of deterioration in pipeline integrity is loss of metal from the pipe wall due to corrosion or mechanical damage. Whether due to the former or latter, the level of MFL near where the metal loss will be higher than ordinary. This is a conclusion drawn from theoretical analysis[1], and experiments support this opinion strongly[2].

Box is the smallest rectangle encapsulating the area of deterioration. The edges of the box should be parallel or perpendicular to the axis of pipe. l is the length of box edges which are parallel to axis. w is the length of box edges which are perpendicular to axis. $d$is the depth of metal loss, namely the maximum value of a deterioration below ordinary plane without metal loss.

As show in Fig.1, the defect on the curved face of pipe spool can be deemed to be on a plane, since the diameter of the pipe is much larger than its thickness. And we can draw a box as defined above. Conventionally, metal loss is divided into 7 types according to their geometrical length (l) and width (w). The defects are never beyond the seven types (Table 1 and Fig.2).

1.1 Magnetic Flux Leakage (MFL) Technology in Metal Loss Inspection

To decide to which kind certain defect belongs, w and l of box is necessary. But in reality, these data can't be obtained easily because pipes are usually buried deep in earth. The most usually used method to detect pipes is MFL technology.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1254–1260, 2006.

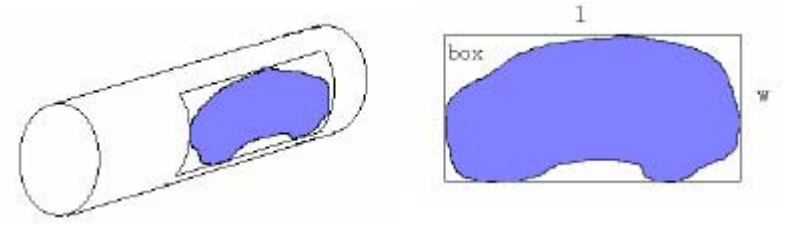


Fig. 1. Defect and its box on outer spool wall

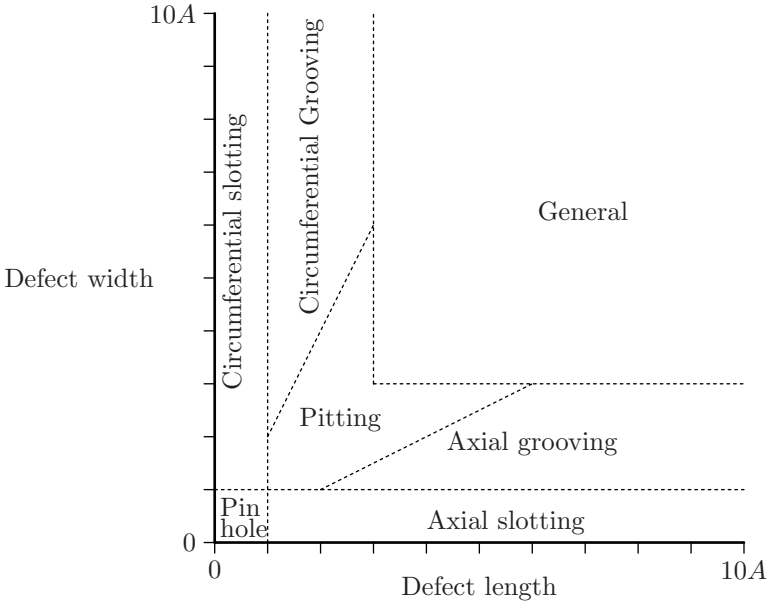


Fig. 2. Conventional classification of defects according to their box's length and width (horizontal axis indicates defect length in a coordinate plane and vertical width)

Table 1. Conventional classification of defects according to their box's length and width. A is the thickness of spool wall.

Defect type	Criterion
General	$w \geq 3A$ and $l \geq 3A$
Pitting	$(A \leq w < 6A$ and $A \leq l < 6A$ and $0.5 < \frac{l}{w} < 2)$ and not $(w > 3A$ and $l > 3A)$
Axial grooving	$A \leq w < 3A$ and $\frac{l}{w} \geq 2$
Circumferential Grooving	$\frac{l}{w} \leq 0.5$ and $A < l < 3A$
Pinhole	$0 < w < A$ and $0 < l < A$
Axial slotting	$0 < w < A$ and $l \geq A$
Circumferential slotting	$w \geq A$ and $0 < l < A$

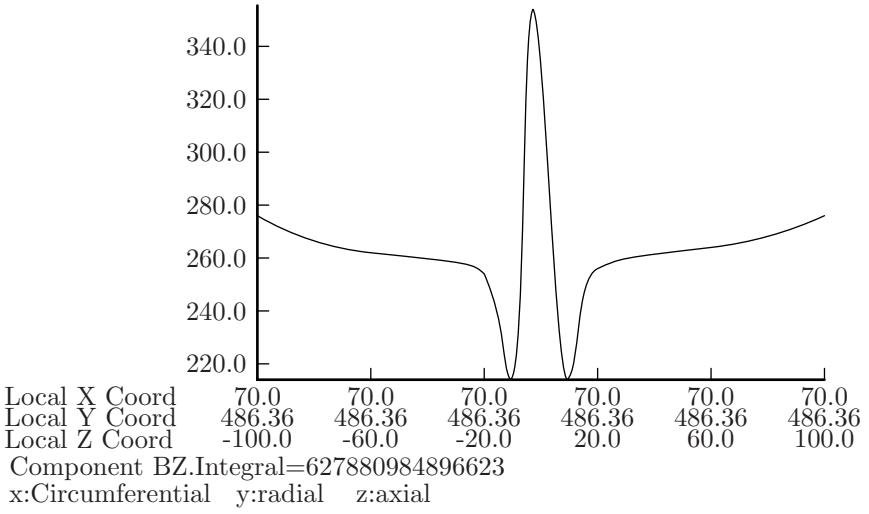


Fig. 3. MFL data of one channel

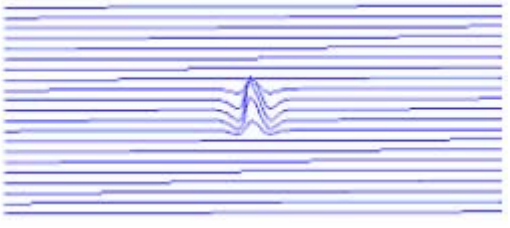


Fig. 4. MFL data map of channels

The MFL data are collected by Pipe Pig. Pipe Pig is a machine like capsule equipped with 400 (maybe larger or smaller according to the diameter of pipe) combinations of stimulator and sensor. These combinations are uniform-spaced on a circle, which is just smaller than the circle of pipe wall. So when the Pig moves in pipe, we can accept a spool of data. The data picked up by a sensor is called a channel (Fig 3), and the data of many channels can form a map of lines (Fig 4). Certainly, these lines are almost uniform where the pipe wall is perfect and are curved where defects.

1.2 Traditional Method of Obtaining Defects' Length, Width and Depth

Magnetic field data of a defect can be seen as a raised face. So the part above ordinary level is a swell. We can draw a smallest 3-D box enclose the swell, and the box's length ($l_{\mathbf{m}}$) is along the channel, the width ($w_{\mathbf{m}}$) is perpendicular to the channel and in the plane of channels, and the height ($h_{\mathbf{m}}$) is perpendicular to the plane of the channels.

Since the raised face of MFL data is caused by metal loss in certain box of area, $w_{\mathbf{m}}$, $l_{\mathbf{m}}$ and $h_{\mathbf{m}}$ should be functions of w, l and d, their relationship can be shown as

$$\begin{cases} w_{\mathbf{m}} = W_{\mathbf{m}}(w, l, d), \\ l_{\mathbf{m}} = L_{\mathbf{m}}(w, l, d), \\ h_{\mathbf{m}} = H_{\mathbf{m}}(w, l, d). \end{cases} \tag{1}$$

But we aim to inference the formula such as below

$$\begin{cases} w = W(w_{\mathbf{m}}, l_{\mathbf{m}}, h_{\mathbf{m}}), \\ l = L(w_{\mathbf{m}}, l_{\mathbf{m}}, d_{\mathbf{m}}), \\ d = D(w_{\mathbf{m}}, l_{\mathbf{m}}, d_{\mathbf{m}}). \end{cases} \tag{2}$$

The functions $L(.)$, $W(.)$ and $D(.)$ should be determined. Traditional methods are either based on analysis according to physical theorem or based on experiences. Employing physical theorem and numerical analysis, experts obtained nonlinear relations (for the reason of commercial, the detail of the relation between geometric and magnetic data I s neglected).

2 Using Neural Network to Evaluate the Defects

2.1 Construction of Network

We adopt BP network to build the relationship between $w_{\mathbf{m}}$, $l_{\mathbf{m}}$, $h_{\mathbf{m}}$ and w, l, d. In the BP network, the amount of nodes in both input layer and output layer are 3. The input vector is $(w_{\mathbf{m}}, l_{\mathbf{m}}, h_{\mathbf{m}})^T$ and output vector $(w, l, d)^T$.

The hidden layer has 6 *tan-sigmoid* neurons whose transfer function is $f(x) = \frac{2}{1+\mathrm{e}^{-2x}} - 1$. And the output lay has 3 pure linear function neurons whose transfer function is $g(x) = x$.

Let n be the number of input neuron, and m hidden neuron. $[w_{ji}]_{m\times n}$ links input and hidden neuron. $(\theta_i)_m$ is offset of hidden layer. The weight between hidden and output neuron is $(u_j)_m$, and r is output neuron's offset. Equations 3 to 9 [3] [5] show the steps to train BP network.

Calculate inputs of hidden neuron

$$a_{jk} = \sum_{i=1}^{n} w_{ji} x_{ik} - \theta_j. \tag{3}$$

$(i = 1, 2, ..., n, j = 1, 2, ..., m$, and the same in formulas below)

Calculate output of hidden neuron

$$b_{jk} = f(a_k). \tag{4}$$

Calculate input of output neuron

$$c_k = \sum_{j=1}^{m} u_j b_k - r_k. \tag{5}$$

Calculate output of output neuron

$$y_k = g(c_k). \tag{6}$$

Correction term of weight links hidden and output neuron

$$\begin{cases} \delta_k = (y_k - \overline{y}_k) f(c_k), \\ \Delta u_{jk} = \eta b_{jk} \delta_k. \end{cases} \tag{7}$$

Correction term of weight links input and hidden neuron

$$\begin{cases} \delta_{jk} = f(a_k) \sum_{j=1}^{m} u_j \delta_k, \\ \Delta w_{ji} = \eta \delta_{jk} x_{ki}. \end{cases} \tag{8}$$

Update weight

$$\begin{cases} w'_{ji} = w_{ji} + \Delta w_{ji}, \\ u'_j = u_j + \Delta u_j. \end{cases} \tag{9}$$

Before training and calculating, data for input should be normalized. Table 2 shows 8 samples of real data groups. The range of $w_{\mathbf{m}}$, $l_{\mathbf{m}}$, w, l is $[0, 250]$, $h_{\mathbf{m}}$ $[0, 180]$, and d $[0, A]$. A is the thickness of pipe, here A equals to 15. We linearly map these data to $\widehat{w}_{\mathbf{m}}$, $\widehat{l}_{\mathbf{m}}$, $\widehat{h}_{\mathbf{m}}$ and $\widehat{w}$, $\widehat{l}$, $\widehat{d}$ which are in the range of $[0, 1]$

Table 2. Conventional classification of defects according to their box's length and width. A is the thickness of spool wall.

$w_{\mathbf{m}}(mm)$	$l_{\mathbf{m}}(mm)$	$h_{\mathbf{m}}(Gauss)$	$w(mm)$	$l(mm)$	$d(Gauss)$
12.5	11.3	50.30	10.0	10.4	3.92
20.2	11.5	70.78	16.0	8.20	4.93
22.7	35.4	72.43	16.5	32.6	5.21
19.3	50.3	97.33	18.1	47.5	7.97
78.5	30.0	73.22	75.3	26.5	5.43
35.3	53.7	96.09	30.1	49.1	8.43
120	42.5	120.3	109	40.2	10.3
210	123	132.3	198	118	12.4

Table 3. The data in Table 2 is normalized to the range of $[0, 1]$ linearly, and the units are inconsequential

$\widehat{w}_{\mathbf{m}}$	$\widehat{l}_{\mathbf{m}}$	$\widehat{h}_{\mathbf{m}}$	$\widehat{w}$	$\widehat{l}$	$\widehat{d}$
0.0500	0.0452	0.2794	0.0400	0.0416	0.2613
0.0808	0.0460	0.3932	0.0640	0.0328	0.3287
0.0908	0.1416	0.4024	0.0660	0.1304	0.3473
0.0772	0.2012	0.5407	0.0724	0.1900	0.5313
0.3140	0.1200	0.4068	0.3012	0.1060	0.3620
0.1412	0.2148	0.5338	0.1204	0.1964	0.5620
0.4800	0.1700	0.6683	0.4360	0.1608	0.6867
0.8400	0.4920	0.7350	0.7920	0.4720	0.8267

Table 4. Columns whose head followed by *(A)* is defects' actual size,followed by *(N)* is size obtained by BP network, and followed by *(F)* is size obtained by traditional formula

$w(A)$	$l(A)$	$d(A)$	$w(N)$	$l(N)$	$d(A)$	$w(F)$	$l(F)$	$d(F)$
102	212	1.34	103	215	2.3	113	213	2.5
242	120	4.53	250	122	5.2	264	130	7.3
230	23.3	12.5	234	24.2	13.2	242	26.6	18.3
21.4	123	14.7	22.5	124	14.6	21.5	128	16.4
12.3	23.3	7.35	12.5	24.5	7.93	12.5	23.3	7.63
12.3	54.5	9.44	11.3	55.5	10.2	12.5	5.1	9.37
31.1	11.3	1.31	32.2	11.9	1.42	31.5	10.9	1.32
1.41	123	13.3	1.41	125	13.6	1.11	159	31.3

(Table 3). Certainly, the data output by network should be mapped back to their original ranges.

We train the BP network by 160 data groups, the training goal is 0.001 and epoch is 2000.

2.2 Test

Use the data group other than the 160 data groups above to test the network.

Compare to traditional method of evaluation of metal loss in pipe wall, BP neural network is better for very large and very small defects. These also happen when the defect is long and narrow.

3 Conclusion

The mechanics in evaluation of defect on metal gas conducting pipe is not exact, so employing formula to present the relation between defect and MFL is inaccuracy and fussy. Neural network can build the relation ignoring physical theorem. Given enough training data, the network can output famous result. The deficiency of network lies in its dependence on training data. Its accuracy can hold only in the range of training data.

Acknowledgement

This work is supported by National Science Foundation 90410019.

References

1. Huebener, R.P.: Magnetic Flux Structures in Superconductors. Springer-Verlag, Berlin Heidelberg New York (1979) 315–333
2. Lukyanets, S., Snarskii, A., Shamonin, M., Bakaev, V.: Calculation of Magnetic Leakage Field from a Surface Defect in a Linear Ferromagnetic Material: An Analytical Approach. NDT and E International. 36 (2003) 51–55

3. Hecht-Nielson, R.: Theory of the Back-Propagation Neural Network. IJCNN. (1989) 583–604
4. Zhang, G.P.: Neural Networks for Classification: A Survey. IEEE Transac-tions on Systems, Man, and Cybernetics-Part C: Applications and Reviews. 30 (2000) 451–462
5. Branch, D.W., Wheeler, B.C., Brewer, G.J., Leckband, D.E.: Long-term Stability of Microstamped Substrates of Polylysine and Grafted Polyethylene Glycol in Cell Culture Condition. Biomaterials. J. 22 (2001) 1035–1047

Soft Sensor Using PNN Model and Rule Base for Wastewater Treatment Plant

Yejin Kim[1], Hyeon Bae[2], Kyungmin Poo[3], Jongrack Kim[1], Taesup Moon[1], Sungshin Kim[2], and Changwon Kim[1]

[1] Department of Environmental Engineering, Pusan National University
San 30, Jangjun-Dong, Kumjung-Gu, Busan, 609-735, Korea
{yjkim, cwkim, jong93, tsmoon}@pusan.ac.kr
[2] School of Electrical & Computer Engineering, Pusan National University
San 30, Jangjun-Dong, Kumjung-Gu, Busan, 609-735, Korea
{baehyeon, sskim}@pusan.ac.kr
[3] Water Environment & Remediation Center, Korea Institute of Science and Technology
39-1, Hawolkokdong, Seongbukku, Seoul, 136-791, Korea
kmboo@kist.re.kr

Abstract. The biological wastewater treatment plant, which uses microbial community to remove organic matter and nutrients in wastewater, is known as its nonlinear behavior and uncertainty to operate. In spite of strong needs of automatic monitoring of nutrients, it is thought that tremendous expense may be required to install equipments related with remote control system, especially on-line sensors for monitoring organic and nutrient concentrations in the treatment processes. In this research, as a cost-effective tool for replacing expensive on-line sensor, PNN(Polynomial Neural Network) models were developed to estimate the NOx-N and ammonia concentrations by only using on-line values of ORP, DO and pH at the wastewater treatment plant. Developed PNN model could estimate the NOx-N and ammonia profile well. However, the error was increased at the first anoxic period of the first sub-cycle and NOx-N accumulation was occurred at the sub-cycle. To deal with those errors, the rule-base-compensator was developed based on operational knowledge.

1 Introduction

The biological wastewater treatment plant, which uses microbial community to remove organic matter and nutrients in wastewater, is known as its nonlinear behavior and uncertainty to operate. Therefore, operation of the biological wastewater treatment process much depends on observation and knowledge of operators. Recently, small-scale sewage treatment plants (SSTPs) have been gradually focused because large-scale wastewater treatment plants were mostly constructed and operated in Korea. Nearly 330 SSTPs may be constructed in those areas in the near future and they should be incorporated by unmanned remote management system. It is thought that tremendous expense may be required to install equipments related with remote control system, especially on-line sensors for monitoring organic and nutrient concentrations in the treatment processes. The minimum numbers of sensors are recommended to reduce initial investment and management cost.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1261 – 1269, 2006.

Therefore, many researches have worked on developing cheaper alternative sensors to replace the expensive on-line nutrient sensors for wastewater treatment plants. Lee *et al.* [1] used artificial neural network to develop the soft sensor. However, this sensor required too many input variables such as ORP(t), pH(t), DO(t), ORP(t-30), pH(t-30) and so on, which made the model structure and input data processing much complicated. They also used two models applied for separately anoxic and aerobic period. The model structure was so complicated that it was difficult to modify the model. As other important research results, there were some cases of developing soft sensor to predict some variables which takes long time to measure manually [2][4]. However, they used input variables as influent measurements which also take some hours to measure. Therefore, the fast sensing of target output variables were difficult. As other cases, there was some cases of using on-line sensor measurements to detect target variables [3][5]. Their soft sensors were made by artificial neural network (ANN) and each variable had each ANN model. However, more models make more variables to be set and modified.

In this research, as a cost-effective tool for replacing expensive on-line sensor, the soft sensor was developed to calculate NOx-N and ammonia concentrations which were the key variables in the wastewater treatment process. This on-line soft sensor was made using on-line values of ORP, DO and pH obtained from conventional sensors. Sequencing batch reactor (SBR) that was considered as the suitable and flexible process for SSTP was selected as a target treatment process. The polynomial neural network (PNN) [6] was applied to make soft sensor. PNN could filter unimportant input variables during training procedure that could prevent over-training. Also it had relatively simple structure that could be easily programmed and updated. In addition, the errors between model estimation results and real measured data could be reduced using the *rule-base*-compensator representing human operator's knowledge. This could significantly improve the soft sensor's performance.

2 Methods

2.1 Polynomial Neural Network

PNN based on the GMDH algorithm is a useful method to model the system from many observed data and input variables. It is widely employed for the modeling of dynamic systems, prediction, and artificial intelligent control because of the advantages in data handling. Fig. 1 shows the model of four input variables and the outputs of node13 and node15 are represented as follows:

$$\begin{aligned} y_{13} &= f_1(x_1, x_2) = \omega_{01} + \omega_{11}x_1 + \omega_{21}x_2 + \omega_{31}x_1x_2 + \omega_{41}{x_1}^2 + \omega_{51}{x_2}^2, \\ y_{15} &= f_2(x_3, x_4) = \omega_{02} + \omega_{12}x_3 + \omega_{22}x_4 + \omega_{32}x_3x_4 + \omega_{42}{x_3}^2 + \omega_{52}{x_4}^2. \end{aligned} \tag{1}$$

where y_{ij} is the jth node of ith layer. The final output z is represented by the polynomial equation of y_{13} and y_{15} as

$$z = f_3(y_{13}, y_{15}) = \omega_{03} + \omega_{13}y_{13} + \omega_{23}y_{15} + \omega_{33}y_{13}y_{15} + \omega_{43}{y_{13}}^2 + \omega_{53}{y_{15}}^2. \tag{2}$$

The least square estimator (LSE) is applied for estimating the parameters at an each node to minimize an objective function. If there is m number of data, the output equations at each node are formed as follows:

$$\begin{bmatrix} y_{ij}^1 \\ y_{ij}^2 \\ \vdots \\ y_{ij}^m \end{bmatrix} = \begin{bmatrix} x_{0(1)} & x_{1(1)} & x_{2(1)} & x_{1(1)}x_{2(1)} & x_{1(1)}^2 & x_{2(1)}^2 \\ x_{0(2)} & x_{1(2)} & x_{2(2)} & x_{1(2)}x_{2(2)} & x_{1(2)}^2 & x_{2(2)}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{0(m)} & x_{1(m)} & x_{2(m)} & x_{1(m)}x_{2(m)} & x_{1(m)}^2 & x_{2(m)}^2 \end{bmatrix} \begin{bmatrix} \omega_0 \\ \omega_1 \\ \vdots \\ \omega_5 \end{bmatrix}, \tag{3}$$

where x_0=1, $\mathbf{y}_{ij}=\mathbf{\Phi\omega}$, and y_{ij} is the jth node of ith layer.

Statistical learning networks have no loops. Therefore the network is a tree of interconnected functions that implements a single input/output function. In the literature, there are described several composition schemes for network functions and corresponding estimation algorithms [6]. The parameters are estimated by

$$J(\boldsymbol{\omega}) = \left\| \mathbf{y}_{\mathbf{ij}} - \mathbf{\Phi\omega} \right\|^2 = \sum_{k=1}^{m} \left[y_{ij}^k - \sum_{l=0}^{5} \omega_l \phi_{kl+1} \right]^2 \tag{4}$$

$$\boldsymbol{\omega} = (\mathbf{\Phi}^T \mathbf{\Phi})^{-1} \mathbf{\Phi}^T \mathbf{y}_{\mathbf{ij}} . \tag{5}$$

All of parameters are estimated in this step and then the polynomial equation is completely determined for the node's output in each layer.

Input Selection. In Fig. 1, DPNN has 4 input variables and one output variable and nodes are 6 at the first layer that are determined by the combination 4C2 because two input variables are connected into each node. Among the 6 nodes, three nodes are selected based on a performance indicator. The selected three nodes are used for the next layer inputs. The number of nodes at the second layer is three that are constructed by the combination 3C2. At the final layer, the best performance node is selected from three nodes of the output layer. Finally, the selected nodes determine the best model. In Fig. 3, node13 and node15 at the first layer, and node23 at the second layer are selected for the final model. Therefore, the best inputs are chosen as x1, x2, and x4. The output of node13 is composed with the coefficient and variable matrix as follows:

$$\begin{aligned} \text{node13} &= \omega_{01} + \omega_{11}x_1 + \omega_{21}x_4 + \omega_{31}x_1x_4 + \omega_{41}x_1^2 + \omega_{51}x_4^2 \\ &= [\omega_{01} \quad \omega_{11} \quad \omega_{21} \quad \omega_{31} \quad \omega_{41} \quad \omega_{51}][1 \quad x_1 \quad x_4 \quad x_1x_4 \quad x_1^2 \quad x_4^2]^T . \end{aligned} \tag{6}$$

The variable x_4 is commonly selected at the node13 and node15. Therefore, x_4 is considered as the important variable for the performance.

2.2 Target Treatment Plant and Data Collection

Target plant was pilot-scale SBR working volume of 60 L equipped with on-line pH, DO, ORP sensors. The control panel for blower and mixer manipulation also was installed. Measured values obtained from sensors were stored every 30 seconds.

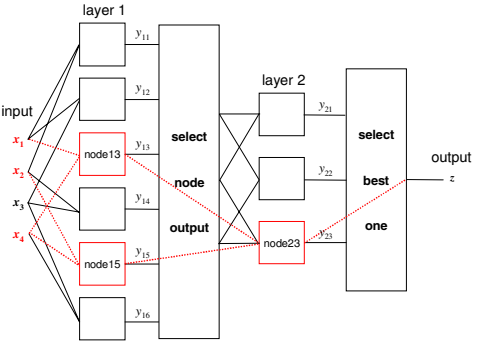
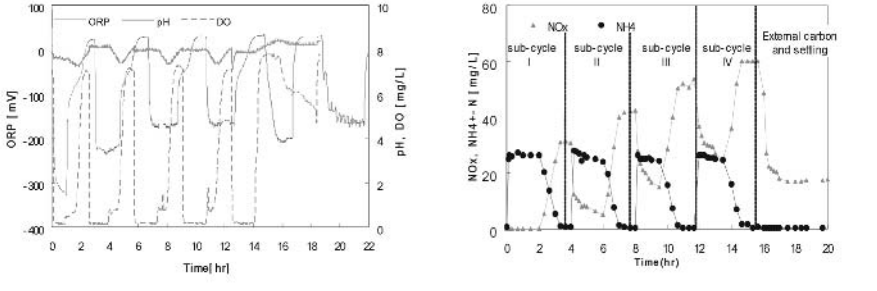


Fig. 1. Selection variable for the best performance

NH_4^+-N and NOx-N were analyzed with Auto Analyzer 3 (Bran+Luebbe) after filtering with 0.45 μm filter. MLVSS concentration was maintained about 7000 mg/L and temperature was controlled as 30. The ammonia loading rate was manipulated as 0.175, 0.25 and 0.4 kgNH_4^+-N/m^3 with C/N ratio of 3. Fig. 2 shows that operation data at the ammonia loading rate of 0.175 kg/m^3-day with C/N ratio of 3. The NOx-N concentration was gradually accumulated every sub-cycle until addition of external carbon, suggesting that denitrification was limited by the lack of carbon source. The denitrification capacity in each sub-cycle was maintained constant and the profiles of ORP, pH, and DO showed similar pattern.



(a) Temporal profile of ORP, pH, DO (b) Temporal profiles of NH4+-N and NOx-N

Fig. 2. Operation data at ammonia loading rate of 0.175 kg NH4+-N /m3-day (C/N ratio 3)

2.3 Model Development for Soft Sensor

Poo [8] found that the *DO-lag-time* that was defined as the required time for the DO level to reach 3 mg/L had proportional relationship with influent loading rate. Bae *et al.* [9] developed the *N-fuzzy-inference-system* for evaluating ammonia loading rate based on the *DO-lag-time*. Prior to application of the PNN model, the classification of loading rate was performed using the *N-fuzzy-inference-system.* Our research concentrated on NOx-N and ammonia estimation with PNN model and the *rule-base-*

compensator as inner area of bold line in Fig. 3. The training data set for PNN model was the first and second sub-cycle data at each experimental condition. The anoxic period among the first sub-cycle was not used for training because of absence of NOx-N.

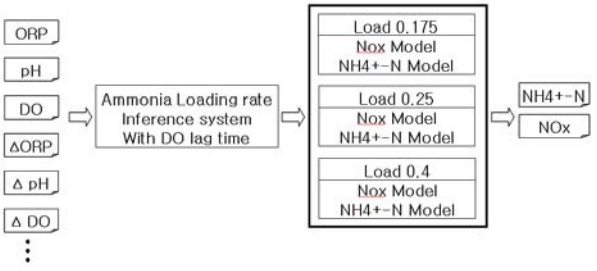
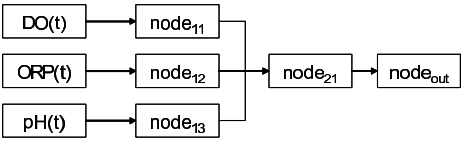


Fig. 3. Schematic diagram of soft sensor

3 Results and Discussion

3.1 PNN Model Development

Fig. 4 shows the sample structure of NOx-N estimation model developed for ammonia loading rate of 0.4 kg/m^3 at C/N ratio 3. Each node had an equation to calculate output value. The coefficients of the polynomials were automatically determined via the model training procedure.



(a) PNN model structure

$$node_{11} = -0.7482 + 1.1350 * DO(t)$$
$$node_{12} = 1.2292 + 0.0075 * ORP(t)$$
$$node_{13} = -27.7693 + 3.6693 * PH(t)$$

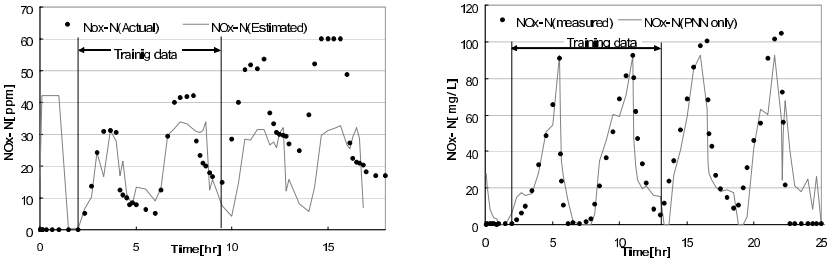
$$node_{21} = -0.1635 - 1.27 * node_{11} - 0.3023 * node_{12} - 1.1798 * node_{13} + 0.1554 * node_{11}^2 - 0.7004 * node_{12}^2 + 0.0901 * node_{13}^2 - 1.0793 * node_{11} * node_{12} - 1.3708 * node_{11} * node_{13} - 0.4021 * node_{12} * node_{13} - 0.4749 * node_{11} * node_{12} * node_{13} + 0.0843 * node_{11}^3 - 0.2042 * node_{12}^3$$

$$NOx\text{-}N\text{-}N = 0 + 25.6506 + 27.92276 * node_{21}$$

(b) PNN model equation

Fig. 4. NOx-N estimation model for ammonia load 0.4kg/m^3 with C/N ratio of 3

The results of model application and measurements are shown in Fig. 5. The PNN model was used to predict NOx-N concentration at the loading rate of 0.4 and 0.175 with C/N ratio of 3. However, it could not predict the NOx-N accumulation because the ORP, pH and DO patterns as input variables of PNN model were not varied according the NOx-N accumulation. To overcome this difficulty a rule-base compensator was proposed as following section.



(a) (b)

Fig. 5. NOx-N estimation results. (a) Load 0.175, (b) Load 0.4.

3.2 Development of Rule-Base-Compensator

Rule 1 : Compensation for no denitrification at first anoxic period

During the first anoxic period, ORP decreased because of the oxygen depletion. In this case, PNN model could not estimate NOx-N concentration. The pattern of ORP, DO and pH values were similar to those at other sub-cycle in Fig. 3. In order to estimate NOx-N concentration at first anoxic stage, the first rule was used as below.

IF "state" is "ANO1", THEN NOx-N(t)=NOx-N_last_fin

NOx-N_last_fin = The final effluent concentration of NOx-N at the last cycle.
ANO1 = The first anoxic stage of 1st sub-cycle

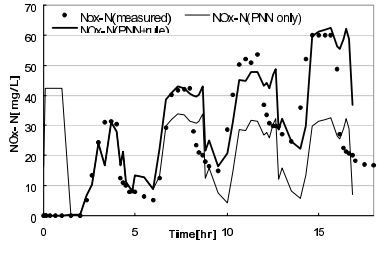
Rule 2 : Compensation for NOx-N accumulation at each sub-cycle

The second rule was the compensation for NOx-N accumulation at C/N ratio of 3 caused by the lack of carbon source in influent. The NOx-N concentration increased with sub-cycle repetition. Poo [8] presented that the minimum ORP value was higher than -300 mV when denitrification was not completed. Based on this operating knowledge, following rule was suggested.

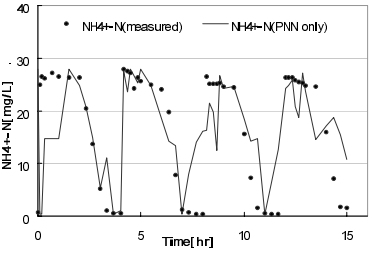
IF ORP_{ANO_MIN} > -300mV, THEN NOx-N(t)=NOx-N_last+NOx-N(t_pnn)

NOx-N_last = estimated NOx-N concentration at the last of previous sub-cycle
NOx-N(t_pnn) = estimated NOx-N concentration at time t by PNN only

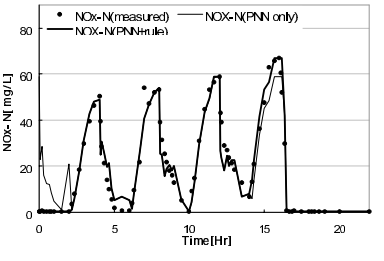
Estimated NOx-N and ammonia concentration are compared with measured value in Fig. 6. The *rule-base-compensator* made the model produce reliable estimation results during NOx-N accumulation. The RMSE (Root Mean Square Error) of each case are listed at Table 1. It should be noted that the RMSE was decreased significantly about 0.5 ~ 15 times by incorporating the *rule-base-compensator*. Relatively high RMSE were presented at ammonia estimation model. However, the model could estimate well the pattern of ammonia profile as shown in Fig. 6. It was said that the high RMSE was originated from the time difference between measured and estimated data.



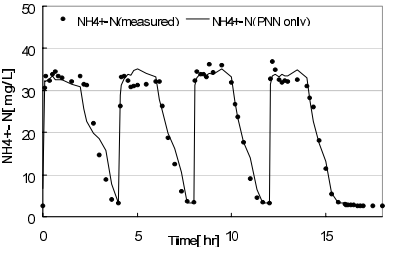
(a) Load 0.175, NOx-N



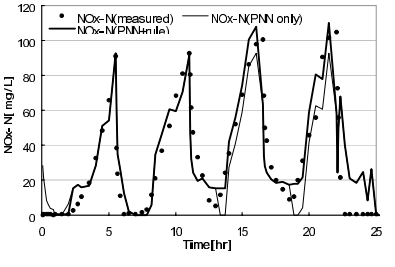
(b) Load 0.175, NH4+-N



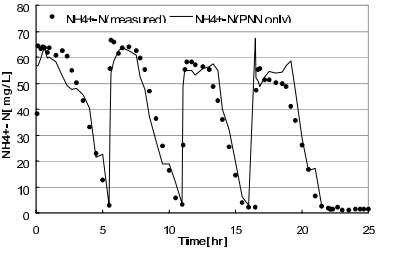
(c) Load 0.25, NOx-N



(d) Load 0.25, NH4+-N



(e) Load 0.4, NOx-N



(f) Load 0.4, NH4+-N

Fig. 6. Measured and estimated concentration of NOx-N and ammonia

Table 1. RMSE of the soft sensor estimation results

Output var.	Model	Loading rate[kg NH_4^+-N/m^3]		
		0.175	0.25	0.4
NOx-N	PNN only	19.131	10.661	15.649
	PNN + Rule	8.433	5.254	1.231
NH4+-N	PNN only	9.184	2.543	11.222

4 Conclusions

The soft sensor based on PNN model which can estimate NOx-N and ammonia concentration in SBR was developed in this research. Developed PNN model could estimate the NOx-N and ammonia profile well. However, the error was increased at the first anoxic period of the first sub-cycle and NOx-N accumulation was occurred at sub-cycle. Therefore, the *rule-base-compensator* was developed to solve these problems based on operational knowledge. The incorporation of the *rule-base-compensator* strengthened the PNN model accuracy that the developed model could estimate both of NOx-N and ammonia at the operating condition of various influent loading rates and C/N ratios. It was concluded that developed PNN model with the *rule –base-compensator* could be an alternative for nutrients sensor.

Acknowledgements

Dr. Haejin Woo is acknowledged for her great contribution to this work who passed away due to cancer during her carrying out this project. This work was supported by grant No. (R01-2003-000-10714-0) from the Basic Research program of the Korea Science & Engineering Foundation and grant No. (KRF-2004-042-D00114) from the Korea Research Foundation.

References

1. Lee, D. S., Park, J. M.: Neural Network Modeling for On-line Estimation of Nutrient Dynamics in a Sequentially-operated Batch Reactor. Journal of Biotechnology. 75 (1999) 229-239
2. Choi, D. J., Park, H.: A Hybrid Artificial Neural Network as a Software Sensor for Optimal Control of a Wastewater Treatment Process. Wat. Res. 35(16) (2001) 3959-3967
3. Charef, A., Ghauch, A., Baussand, P., Martin-Bouyer, M.: Water Quality Monitoring Using a Smart Sensing System. Measurement. 28 (2000) 219-224
4. Zyngier, D., Araujo, O.Q.F., Coelho, M.A.Z., Lima, E. L.: Robust Soft Sensors for SBR Monitoring. Wat. Sci. Tech. 43(3) (2001) 101-105
5. Luccarini, L., Porra, E., Spagni, A., Ratini, P., Grilli, S., Longhi, S., Bortone, G.: Soft Snsors for Control of Nitrogen and Phosphorus Removal from Wastewaters by Neural Networks. Wat. Sci. Tech.45(4-5) (2002) 101-107
6. Graybill, F. A.: Theory and Application of the Linear Model. Duxbury Press. CA (1976)

7. Kim, S., Vachtsevanos, G. J.: An Intelligent Approach to Integration and Control of Textile Processes. Information Sciences. 123 (2000) 181-199
8. Poo, K. M., Jun, B. H., Im, J. H., Ko, J. H., Woo, H. J., Kim, C. W.: ORP/DO Based Control and Remote Monitoring System for Nitrogen Removal in SBR: 4th International symposium on advanced environmental monitoring, Jeju Hyatt hotel, Korea (2002)
9. Bae, H., Jun. B. H., Kim, Y. J., Kim, S.: Remote Process Management System Based Upon Artificial Intelligence for the Process Control of the SBR Type of the Piggery Wastewater Treatment Plant : IWA Specialty Symposium on Strong Nitrogenous and Agro-Wastewater, Seoul, Korea (2003)

Knowledge Acquisition Based on Neural Networks for Performance Evaluation of Sugarcane Harvester

Fang-Lan Ma[1,2], Shang-Ping Li[3], Yu-Lin He[2], Shi Liang[1], and Shan-Shan Hu[1]

[1] College of Mechanical Engineering, Guangxi University,
Postfach 530004, Nanning, China
lan_mfl@163.com
[2] College of Mechanical Engineering, Chongqing University,
Postfach 400044, Chongqing, China
[3] Mechanical Department, Guangxi Engineering Institute,
Postfach 545006, Liuzhou, China
spli@vip.sina.com

Abstract. Expertise acquisition is always the obstacle and bottleneck in the development of intelligent design system. In order to generalize and accumulate the expertise and experience of simulation analysis and experiments, the intelligent design system of sugarcane harvester is introduced. In the intelligent system of sugarcane harvester, the neural network is applied to overcome the difficulty of knowledge acquisition (KA). In this study, the application of neural network in the system is illustrated, including data predisposal, generation and management of the knowledge. An example is given to explain the application as well. The research shows using neural network can simplify the procedure of knowledge acquisition. It can also evaluate and forecast the performance of sugarcane harvester in design phrase. And it is beneficial to enhance the development success rate of the digital product and to lessen the development cost of physical prototype.

1 Introduction

In the south of China, many academies and colleges have developed the sugarcane harvesters since 1970s or the sugarcane harvesters are introduced from other countries to imitate a new one. However, the physical prototypes are still developed with the traditional pattern so that the development cost is higher and the development cycle is longer. In our studies [1-4], the working mechanisms and the laws of motion of the major components, such as the equipments of holding and cutting, the cleaning element of sugarcane harvester, are simulated and analyzed with the virtual design method and experimental researches. Meanwhile, the design experience is enriched through simulations and experiments. In order to make good use of the experience and design knowledge, the knowledge base (KB) and the intelligent system are presented to help the designers to develop the sugarcane harvester effectively.

With the development of computer technology and artificial intelligence, the expert systems, which have the ability of solving problem similar to the experts, are applied widely[5,6]. However, its development is limited by KA. The automatic KA methods can be divided into machine learning, induction, analogy and artificial neural network

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1270 – 1276, 2006.

[7, 8]. The artificial neural network can improve the capacity and quality of the stored knowledge through new samples' training. Also it can extract the similarities and discrepancy among the samples. These make it become one of the focal points of KA. Therefore, the intelligent design system of sugarcane harvester is integrated with the artificial neural network to build KB. The compatibility and associability of neural network are used to acquire and represent the knowledge and the feedforward propagation algorithm is applied to reuse it. So the performance of sugarcane harvester can be evaluated and forecasted in the design phase.

2 Intelligent Design System Structure of Sugarcane Harvester

In the intelligent design system of sugarcane harvester (Fig.1), the design requirements of users are first of all input into the system and mapped into the functions of sugarcane harvester. The reasoning system searches for knowledge in KB, matches them with the requirements and outputs design plan and major parameter values. Second, the structural and parameterized design is carried out on CAD system based on the plan and parameter values. And then, the dynamic and kinetic characteristics of the parameterized model output from CAD system are simulated and analyzed on CAE system. Finally, these characteristics are input into the performance evaluation module to evaluate the performance of sugarcane harvester according to the evaluation indexes of KB. If the characteristics can meet the needs of the performance requirements, the digital prototype is achieved. Otherwise, it needs to return to the upper module or the uppermost module to choose and design the plan of the parameterized model once more.

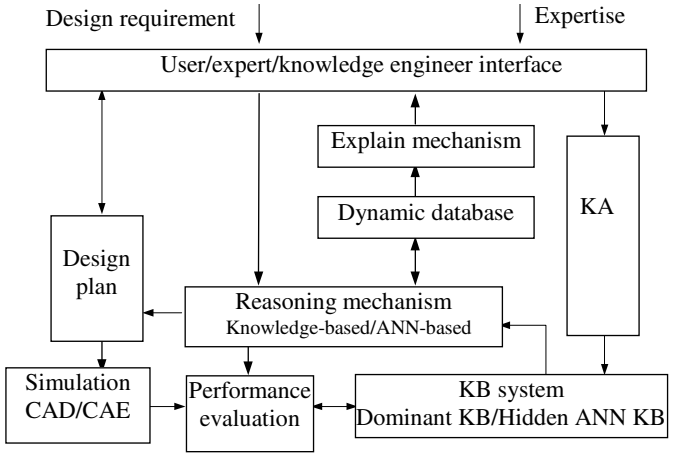


Fig. 1. Structure of the intelligent design system based on neural network

According to the above phases of the product design, the intelligent design system of sugarcane harvester is divided into three modules, that is, the expert system for design, the engineering analysis module and the performance evaluation module. In

the expert system, the reasoning mechanism searches for knowledge in KB according to the performance requirements of product to output the initial design plan of sugarcane harvester. The plan is regarded as the bases to guide and assist the structural and detailed design later. The engineering analysis module is utilized to analyze the dynamic and kinetic characteristics and the performance optimization. The analysis results are output in the forms of curve and data. The performance evaluation module is integrated with several technologies, such as the artificial neural network, fuzzy comprehensive evaluation, to evaluate the cutting performance, the cleaning performance, and others of sugarcane harvester. These modules are independent as well as relative with each other. The expert system provides the engineering analysis module with the structural and functional parameters about the performance of design plan. Meanwhile, the engineering analysis module services the expert system to test the reasonability and reliability of the plan. Based on the above results, the performance evaluation module assesses the plan further. This module can accumulate experience through the analysis procedure. So the capacity of KB is enriched and the intelligent level of the intelligent design system is enhanced. In this paper, it mainly describes the application of neural network in KA of the performance evaluation module.

3 Analysis on Performance Evaluation Module Based on Neural Network

In the intelligent design system of sugarcane harvester, it makes use of the self-learning ability of neural network to exploit the inherent laws among the data achieved from the experiments and simulation analysis to automate the procedure of KA. The neural network can be taken as a calling function embedded into the intelligent system to become a tool of KA and the performance evaluation.

3.1 Data Predisposal and Normalization

Data predisposal means to select the indexes of the sample data and to build the sample database. For the convenience of modification, expansion, and visualization, the sample data is stored in database. Therefore, data predisposal includes the selections of structure and contents of the sample data. Different data indexes reflect different performances of sugarcane harvester, so it needs to build different neural network evaluation models. The major performance parameters of the whole-stalk sugarcane harvester are the proportion of ragged root, the remainder ratio of sugarcane leaf, the productivity ratio, and others.

(1) Index of the proportion of ragged root. The factors affecting the cutting performance are the driving speed of machine, the rotational speed and dip angle of cutting dish, and others. The proportion of ragged root is usually taken as the major evaluation index to measure the quality of harvesting. It is judged through the length and width of the split, the height of convex fracture and the number of splits on the root of sugarcane. Therefore, the performance evaluation model on the proportion of ragged root based on neural network can take the driving speed, the rotational speed, and the dip angle as its input vectors, and the proportion of ragged root as its output.

(2) Index of the remainder ratio of sugarcane leaf. The cleaning equipment of sugarcane harvester depends on the rotating cleaning element to strike and rub against the sugarcane to strip the leaves away. So the effects on the cleaning element are about the reacting force and the stress on the root. These influence directly the remainder ratio of sugarcane leaf, the scrape of the sugarcane, and the service life of the element. The factors affecting the reacting force and the stress includes material of element, the installed spiral angle, the number of rows of element, the installed front angle, and the intersection depth between the element and sugarcane. Thus, these factors can be regarded as the inputs of neural network, and the reacting force and the stress as the outputs to measure the remainder ratio of the cleaning equipment.

(3) Index of productivity. The productivity is one of the whole performance indexes of sugarcane harvester. It is related to the whole links of harvesting sugarcane. In general, it is weighed by the indexes of the proportion of ragged root, the ratio of holding sugarcane, the remainder ratio of sugarcane leaf, transportation, and so on.

In the simulation and experiment of sugarcane harvester, the performance testing is combined with the orthogonal experiment method. Thus, the training samples of neural network are composed of the orthogonal experimental data. The experimental conditions and results are normalized to make the training of neural network easy to be convergent. The transformation pattern is showed as Fig.2. The input vectors come from the levels of orthogonal table. And the output vectors are between 0 and 1, which are transformed from the fuzzy evaluation results of the experimental results.

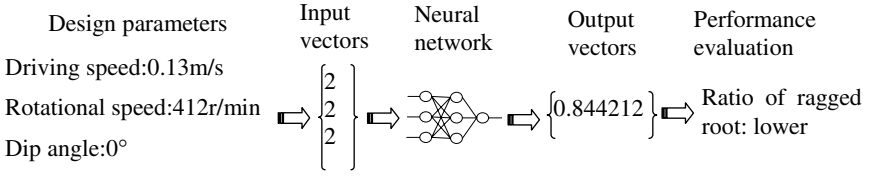


Fig. 2. Transformation pattern of data disposal of neural network

3.2 Modeling of BP Neural Network

The procedure of KA includes the selection of training samples, the structural parameter design of neural network, and the training algorithm. The training samples used to train the neural network integrated in the system come from the simulation and experiments. The neural network is feedforward three-layer network including the input, hidden and output layers, whose activation functions are Sigmoid function. The improved back propagation (BP) algorithm is applied to train the neural network. After the neural network is trained until the error function tends towards stability, the knowledge is contributed in the connecting weights.

3.3 Knowledge Management of Neural Network

The knowledge of neural network includes the initialization knowledge and training knowledge. The initialization knowledge includes the structural parameters of

network, such as the number of layers and nodes, and the training samples. The knowledge acquired from training neural network is mainly memorized in the connecting weights and biases among the nodes of neural network. In order to make use of the knowledge effectively, the connecting weights and biases are stored in database in the form of data file. In this way, the neural network doesn't need to be trained again when it is used to evaluate the performance. And the system directly calls the data stored in database so that the training time of neural network is shortened. However, the knowledge is stored in the form of the structural parameters of neural network, so it can't be revised arbitrary. The regeneration and maintenance of the knowledge can be modified only when the structure and contents of samples or the structure of neural network are changed and the neural network is trained again.

4 Application Example

In order to achieve automatic KA and performance forecast based on BP neural network, the orthogonal experiments are carried out on the cutting experiment platform to obtain training samples. From the orthogonal experiments, the effects of the driving speed, the rotational speed, and the dip angle on the length and width of the split, and the height of convex fracture can be found out. The orthogonal experiments are conducted by the orthogonal table $L_{25}(5^6)$. And the factors are showed in Table 1. The experimental results are in Table 2. The fuzzy comprehensive evaluation method is used to evaluate the proportion of ragged root. The fuzzy evaluation results are also showed in Table 2. In Table 2, when the evaluation result is between 90 and 100 score, it means the proportion of ragged root belongs to lower grade. It is the same to 80~90, 70~80, 60~70, 50~60. They respectively belong to low, average, high, and higher grades.

Table 1. Factors and levels of orthogonal experiment

Driving speed (m/s)(A)	Rotational speed (r/min)(B)	Dip angle (°)(C)
1 (0.13)	1 (412)	1 (0)
2 (0.29)	2 (550)	2 (2)
3 (0.46)	3 (687)	3 (4)
4 (0.66)	4 (825)	4 (6)
5 (0.79)	5 (962)	5 (8)

The neural network applied to evaluate the cutting performance takes the factors in Table 2 as the input vectors, and the fuzzy evaluation results of proportion of ragged root as the output vector. The structure of neural network is 3×3×1(the number of neurons of the input layer, the hidden and output layers). The training samples in Table 2 are normalized with the transformation pattern (showed in Fig.2) and distributed between 0 and 1. The learning rate, the momentum parameter and the overall error of training the neural network are 0.6, 0.8, and 0.0001. After the training

processes are 454, the error function is tended to be stable. And the final training results are in Table 2. Now the connecting weights and biases of the neural network can reflex the inherent law of the training samples. Also these parameters and networked structure can be taken as the actual parameters and structure of the cutting performance forecast model. According to the connecting weights and biases, the knowledge about the performance of the sugarcane harvester is acquired.

Table 2. Results of orthogonal experiments and fuzzy evaluation & BP neural network

No.	A	B	C	Experimental results				Fuzzy evaluation	Neural network	Grade
				Height* (mm)	Length* (mm)	Width* (mm)	Number*			
1	1	1	1	0.55	20.49	0.82	1.67	83.22	84.4212	lower
2	1	2	2	0.62	16.21	1.08	1.7	84.14	83.4104	lower
3	1	3	3	0.34	12.74	1.18	1.6	85.64	83.5784	lower
4	1	4	4	0.53	6.86	0.87	1.44	88.68	84.6516	lower
5	1	5	5	1.17	4.04	0.48	1	89.54	85.7008	lower
6	2	1	2	1.77	15.07	1	1.5	81.93	81.5186	lower
7	2	2	3	1.74	13.26	1.03	1.29	83.16	81.8173	lower
8	2	3	4	2.37	17.15	1.52	2.5	75.41	80.6559	lower
9	2	4	5	2.28	0.38	0.83	1.5	83.20	79.9453	average
10	2	5	1	0.42	20.06	1.21	1.22	83.63	86.2767	lower
11	3	1	3	3.16	17.89	1.22	0.89	75.76	74.4009	average
12	3	2	4	2.98	18.63	0.55	0.86	75.64	76.3033	average
13	3	3	5	3.88	15.38	0.93	0.88	73.84	73.6581	average
14	3	4	1	0.65	20.57	1.08	0.9	84.64	87.9746	lower
15	3	5	2	1.55	12.32	0.85	0.7	85.05	87.3122	lower
16	4	1	4	4.03	18.3	2.22	1.11	70.37	72.3522	average
17	4	2	5	4.51	17.86	0.73	0.71	72.84	75.2807	average
18	4	3	1	0.84	17.02	1.1	0.9	85.57	89.1280	lower
19	4	4	2	0.91	3.25	0.44	0.25	91.13	87.9884	lower
20	4	5	3	2.54	7.2	0.5	0.5	82.42	85.8578	lower
21	5	1	5	5.24	14.22	1.1	1	76.20	74.5393	average
22	5	2	1	1.13	18.11	1.53	1.11	83.66	81.2564	lower
23	5	3	2	0.83	14.14	1.11	1.14	86.28	84.4238	lower
24	5	4	3	2.25	4.67	0.43	0.4	84.65	83.5211	lower
25	5	5	4	3.26	2.5	0.088	0.25	79.22	79.445	average

∗Means the height of the convex fracture, the length, the width and the number of the splits.

The acquired knowledge is reused through the feedforward calculation of neural network to evaluate the performance of sugarcane harvester. For examples, in Table 3, 5 samples not to be taken as training samples are considered as the new design conditions to forecast the performance of sugarcane harvester. These new conditions are also evaluated with fuzzy comprehensive evaluation method. From Table 3, it can be seen that the forecast results are in coincided with the fuzzy evaluation results. It also illustrates using neural network can extract the similarities and discrepancy among samples. The compatibility and associability of neural network can make the knowledge memorized in the connecting weights and biases easy to be acquired and reused.

Table 3. Forecast results of BP neural network

No.	A	B	C	Fuzzy evalustion	Neural network	grade
26	3	4	4	80.91	81.6921	lower
27	3	5	4	83.19	84.3110	lower
28	2	4	4	81.47	83.5762	lower
29	4	4	4	77.95	79.6996	average
30	5	4	4	76.88	78.9937	average

5 Conclusion

For the sugarcane harvester, the expertise and design experience usually come from the simulation and experimental researches. The experience is complex and various, and is often difficult to be acquired and accumulated in the procedure of design. The application of neural network in the intelligent design system of sugarcane harvester can overcome the difficulty of knowledge acquisition to some extend. Meanwhile, the neural network has the ability of associative memory. With the ability, the intelligence and reasoning efficiency of the intelligent design system can be enhanced. Also, the intelligent system of sugarcane harvester can realize the whole links from design to analysis, evaluation and to redesign because the system integrates the expert system technology with neural network. It can meet the need of practical requirements and convenient the designer to develop a new product.

Acknowledgement

This work is supported by Natural Science Foundation of China (50365001).

References

1. Meng, Y.M., Li, S.P., Liu, Z.S., et al.: Visual Virtual Design Platform for Sugarcane Harvester. Journal of Computer-Aided Design&Computer Graphics 14 (2002) 1096-1100
2. Deng, J.L., Li, S.P., Yang, J.Q., et al.: Simulation Study on Virtual Prototyping of a Small-size Sugarcane Combine Harvester. Journal of Agricultural Machinery 33 (2002) 54-56
3. Ren, X.Z., Li, S.P.: Device Design of Cutting Top of Sugarcane. Journal of Agricultural Machinery 35 (2004) 172-174
4. Ma, F.L.: Virtual Experimental Analysis on Cleaning Element in Brush Shape of Sugarcane Harvester. Nanning, Guangxi University (2002)
5. Jiang, S.P., Chen Y.: Design of Knowledge Base of Expert System in ANN-Based Mechanical Design Process. Chinese Mechanical Engineering 13 (2002) 1034-1037
6. Wang, R., Zheng, X.D., He, D.N., et al.: Artificial Neural Network Based Expert System for Diagnosing of Blanking Parts' Defects. Journal of Shanghai Jiao Tong University 35 (2001) 977-980
7. Castro, J.L., Castro-Schez, J.J., Zurita, J.M.: Use of a Fuzzy Machine Learning Technique in the Knowledge Acquisition Process. Fuzzy Sets and Systems 123 (2001) 307-320
8. Su, M.C., Chang, H.T.: Application of Neural Networks Incorporated with Real-valued Genetic Algorithms in Knowledge Acquisition. Fuzzy Sets and Systems 112 (2000) 85-97

Application of Artificial Neural Network in Countercurrent Spray Saturator

Yixing Li, Yuzhang Wang, Shilie Weng, and Yonghong Wang

Key Laboratory for Power Machinery and Engineering of Ministry of Education, China
Shanghai Jiao Tong University, Shanghai 200030, China
dragon_sjtu@yahoo.com.cn

Abstract. This paper presents the application of artificial neural network (ANN) in saturator. Phase Doppler Anemometry (PDA) is utilized to investigate the distribution of water droplets diameter and velocity in the saturator. The data obtained from experiment is used as input-output of ANN. Before using ANN method, some prerequisites have to be processed, including the selection of the number of input and output variables, hidden layer neurons, the network architecture and the normalization of data etc. The results indicate that the trained ANN can provide accurate prediction values which agree with real experimental data closely.

1 Introduction

HAT (humid air turbine) cycle, first put forth by Rao [1], can enhance gas turbine performance greatly, particularly with high efficiency, more specific work output, as well as lower emission of NOx. HAT cycle, therefore, has aroused considerable attention in the urgent situation of energy crisis and anti-pollution. The saturator, key component in the HAT cycle, introduces water droplets under boiling point into the compressed air steam with minimal energy loss [2]. Absorbing the low quality energy of water droplets in low temperature for adding air humidity and temperature, HAT cycle will have more power output. So it's important to research the behavior of the saturator to improve the thermodynamic performance. However, since the air-liquid two-phase flow in saturator combines the heat and mass transfer simultaneously, as well as with severe turbulent flow, the atomization mechanism is too hard to be known just with theoretical methods. Some prior practices [3, 4] have tried to use the conventional methods to set up theory models of saturator, but the results are not satisfactory because of lots of limitations in the assumptions of those theories. Therefore, it's necessary to further study about thermodynamic assessment of saturator with experimental method.

With the support of National Nature Science Foundation of China, an experimental countercurrent spray saturator model system is built. The state-of-the-art laser measurement equipment PDA (Phase Doppler Anemometry) is used in experimental research. Mean diameter and velocity of water droplets at one location in the flow field can be measured with PDA which has high level of accuracy [5]. Analyzing experimental data at different operating conditions, we can learn the characteristics of heat and mass transfer process between water droplets and humid air.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1277 – 1282, 2006.

However, there has one disadvantage in PDA that diameter and velocity of droplets in a single point can be measured at one time. Only a few points can be measured within the limited experimental time, so the information about the whole flow field can't be obtained. Additional tools and technologies should be applied for the imperfection of PDA [6]. The data obtained from the experiment is severe non-linearity, so the selected tools and methods must have the ability of full analyzing such data with high accuracy. Finally, because of the merit in solving these problems efficiently [7, 8], the artificial neural network (ANN) method is considered as the most suitable method. This paper presents the application of ANN to evaluate thermodynamic parameters in saturator.

2 Experimental Facilities

Figure 1 is the schematic diagram of the experimental system. The body of counter-current saturator model consists of transparent glass so that the detecting laser of PDA can get through it. Figure 2 shows the experimental facilities with PDA. The detailed descriptions are presented in the literature [9].

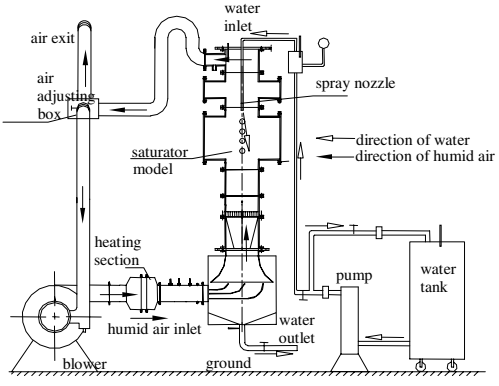


Fig. 1. Experimental system

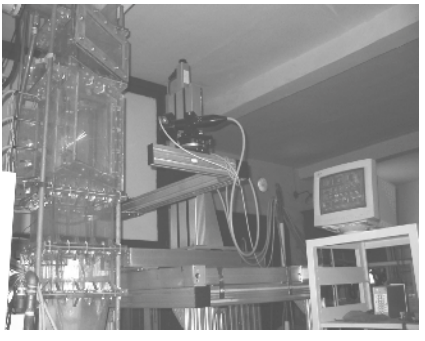


Fig. 2. Experimental facilities with PDA

PDA can measure both distributions of droplets diameter and velocity in the saturator. By changing some input parameters, such as temperature and mass flow rate of water, we can get experimental data at different operating conditions. The results then can be used as the input-output data-set of ANN.

3 Application of Artificial Neuron Network

During the past 10 years, the application of ANN in engineering is becoming more and more popular. The ANN method has been applied successfully in some fields [10]. The advantages of ANN, compared with conventional data process methods, are fastness, simplicity and the ability of learning from examples. It can deal with non-linear problems with robustness and fault tolerance, so less engineering efforts are

required. ANN operates as a "black box" model without requiring detailed information about the system. In addition, it can learn the relationship between the input and output data. The most useful neural network model for steady state systems is 'feed-forward' model, in which the information is propagated in one direction [11]. The structure of neural network usually consists of three parts: input layer, some hidden layers and output layer.

3.1 Selection of Inputs and Outputs Variables

To construct a neural network, inputs and outputs variables should be determined on the basis of the real process of the saturator. Input variables should just include the independent parameters that have considerable influence on the results of outputs. In this study, the interested outputs are the diameter and velocity of droplets in the flow field. According to the spray process and some prior trial experiments, four main parameters are considered as crucial input parameters in ANN: inlet water/air ratio, water inlet temperature, air inlet humidity and the height level of measurement section in the saturator (To simplify the problem, it's assumed that all droplets in one height level are the same diameter). The temperature of inlet air is 23 $^{\circ}C$ constantly in the experiment. Finally the number of inlet and outlet variables is four and two, respectively. Four input variables are inlet water/air ratio, water inlet temperature, air inlet humidity and the height, while output variables are diameter and velocity of droplets.

3.2 Decision of Number of Hidden Layer and Neurons

In this work, the number of input and output variables is few, so a single hidden layer in ANN is enough. However, it is not easy to choose the approximate number of neurons in the hidden layer, because there are no definite rules to determine it only with continuous trials and comparisons. Too few neurons could impair the neural network and prevent correct mapping of input to output. Too many neurons could increase training time and cause over-fitting [12]. After trial and training several times, the optimal number of neurons in hidden layer is chosen to be twelve.

3.3 Other Important Parameters of ANN

For three layers of ANN above, there has no transfer function in the input layer. Logistic sigmod (logsig) transfer function shown in Eq. (1) is chosen in the hidden layer [13].

$$f(w) = \frac{1}{1+e^{-w}} \tag{1}$$

Where w is the weighted sum of the input. Pure linear (purelin) transfer function is used in the output layer. That is shown in Eq. (2).

$$f(z) = z \tag{2}$$

where z is the input of output layer, namely the output of hidden layer.

Because the output range of logsig transfer function is limited between 0 and 1, the necessary normalization process is adopted:

$$\overline{A} = \frac{A - A_{\min}}{A_{\max} - A_{\min}} \tag{3}$$

where A is the raw data, $A_{\min}$ and $A_{\max}$ are the minimum and maximum value in data set. Thus $\overline{A}$ is the normalized value within the range [0, 1]. Before training network, the goal and epochs of ANN is chosen 0.001 and 500 respectively. The BP learning network has adopted Levenberg-Marquardt algorithm. Figure 3 shows the schematic of ANN structure used in present study.

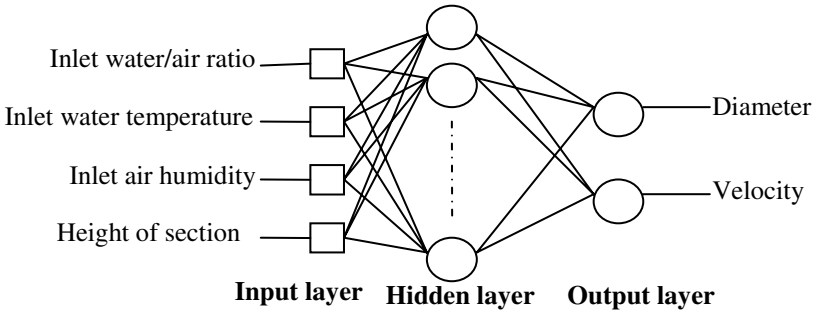


Fig. 3. Schematic of three-layer neural network used in present work

4 Results and Discussion

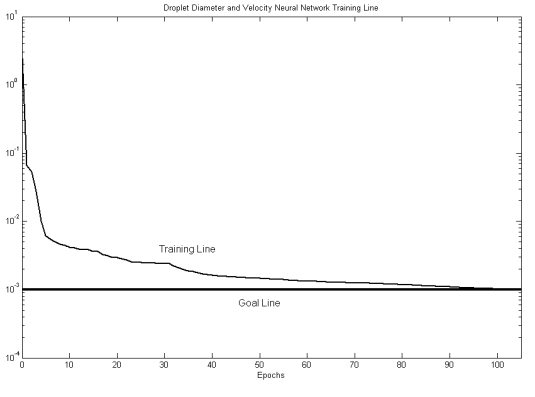


Fig. 4. Training line of ANN

Lots of experimental data is obtained at different operating conditions with PDA. Here 128 data sets are chosen as input-output data source of ANN. Figure 4 shows the neural network training condition. The thin line is the training convergent line and the thick line is the goal line with constant 0.001. It can be seen that the convergent process is very fast. That indicates the structure of ANN is suitable and the number of hidden layer neurons is appropriate. The convergent line reaches the goal line at about 105 epochs. The prediction performance of ANN vs. experimental data of diameter and velocity is shown in Fig. 5. The predicted values are evenly and tightly distributed around two

sides of the regression line. The maximum relative error of diameter and velocity is 5.1% and 7.1% respectively. It demonstrates the high accuracy of ANN and good correlations between prediction values and measurement data. Furthermore, the conclusion can be drawn that most water droplets diameters are in the range between 40 and 65 μm and the maximum velocity is less than 6m/s. We can also notice there are some points of negative velocity[1] in Fig. 5 (b). The reason is that the flow speed of inlet humid air will increase with the decreasing of water/air ratio, thus some small droplets will move upwards along with the air when the velocity of the air is too much higher.

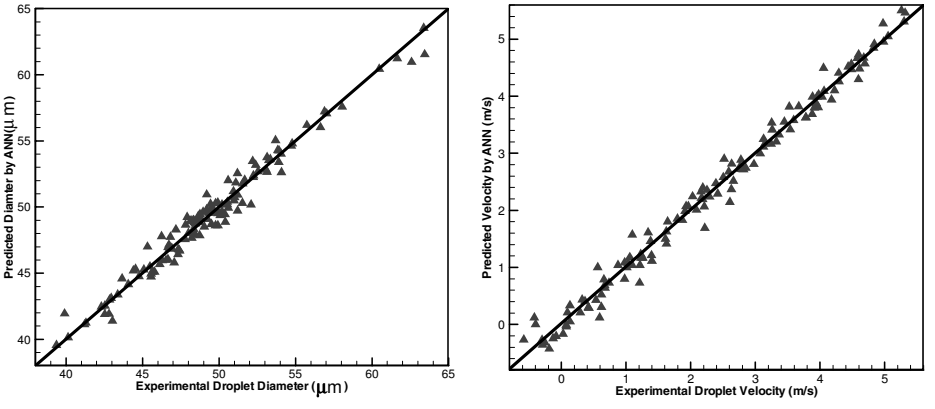


Fig. 5. Correlation of neural network predicted values vs. experimental data

So care should be taken to avoid extrapolation of the results, because, in reality, the neural network model has no reliable basis to give the output response, like any other polynomial regression techniques [14].

After the ANN model is trained, the distribution of diameter and velocity of droplets at any height level section and different operating conditions can be obtained with it. The results can be used as the input of computational fluid dynamic codes or to evaluate the thermodynamic assessment of saturator and explore the behavior of heat and mass transfer process between water droplets and humid air.

5 Conclusions

The saturator, the key component in HAT cycle, combines the complicated water/air two-phase interacting process with severe turbulent flow. It's necessary to evaluate thermodynamic performance of the saturator with experimental method. The diameter and velocity of water droplets are measured with PDA and the experimental data has been processed with ANN method. Before using ANN, the number of input and output variables, the hidden layers neurons and the layer transfer functions are

[1] We define upward direction in the saturator as positive direction.

determined and the input raw data is normalized. The results indicate that the trained ANN model can successfully applied in this work. The convergent speed is fast and the prediction values agree with the real experimental data closely with maximum associated errors less than 5.1% and 7.1%. The results can be used in other further studies, such as CFD codes and thermodynamic assessment of the saturator.

Acknowledgements

This work is supported by Nation Nature Science Foundation of China. The author would like to thank Prof. Jin Mingyue and Jin Hao for their great help.

References

1. Rao, A.D.: Process for Producing Power, US patent no. 4829763 (1989)
2. Xiao, Y.H., Cai, R.X., Lin, R.M.: Modeling Hat Cycle and Thermodynamic Evaluation. Energy Convers. Mgmt. 38(15) (1997) 1605-1612
3. Parente, J.O.S., Traverso, A., Massardo, A.F.: Saturator Analysis for an Evaporative Gas Turbine Cycle. Applied Thermal Engineering 23(10) (2003) 1275-1293
4. Makkinejad, N.: Temperature Profile in Countercurrent/Co-current Spray Tower. International Journal of Heat and Mass Transfer 44(2) (2000) 429-442
5. Denmark: Dantec Measurement Technology GmbH: Dual PDA Manual, User's Guide 3 (1996)
6. Ikeda, Y., Mazurkiewicz, D.: Application of Neural Network Technique to Combustion Spray Dynamic Analysis. Progress in discovery science. Lecture Notes Artificial Intelligence, Vol. 2281. (2001) 408-425
7. Guardani, R., Nascimento, C.A.O., Onimaru, R.S.: Use of Neural Networks in the Analysis of Particle Size Distribution by Laser Diffraction: Tests with Different Particle Systems. Powder Technology 126(1) (2002) 42-50
8. Liu, J.T., Chang, H.B., Hsu, T.Y., et al.: Prediction of the Flow Stress of High-Speed Steel during Hot Deformation Using a BP Artificial Neural Network. Journal of Material Processing Technology 103(2) (2000) 200-205
9. Wang, Y.Z., Liu, Y.W., Weng, S.L.: Experimental Investigation of Two-Phase Flow in Countercurrent Spraying Humidifier with Phase Doppler Anemometry. 4th European Thermal Sciences Conference, Birmingham, UK (2004) 29-31
10. Sozen, A., Arcaklioglu, E., Ozalp, M: A New Approach to Thermodynamic Analysis of Ejector-Absorption Cycle: Artificial Neural Networks. Applied Thermal Engineering 23(8) (2003) 937-952.
11. Nascimento, C.A.O., Guardani, R., Giulietti, M.: Use of Neural Networks in the Analysis of Particle Size Distributions by Laser Diffraction. Powder Technology 90(1) (1997) 89-94
12. Wang, X.D., Yao, M., Chen, X.F.: Application of BP Neural Network for the Abnormity Monitoring in Slab Continuous Casting, Lecture Notes in Computer Science, Vol. 3174 (2004) 601-606
13. Hagan, M.T., Demuth, H.B., Beale, M.H.: Neural Network Design. PWS Publishing Company (1995)
14. Sablani, S.S., Ramaswamy, H., Sreekanth, S., et al.: Neural Network Modeling of Heat Transfer to Liquid Particle Mixtures in Cans Subjected to End-over-End Processing. Food Research International 30(2) (1997) 105-116

Wavelet Neural Networks Approach for Dynamic Measuring Error Decomposition

Yan Shen[1] and Bing Guo[2, 3]

[1] University of Electronic Science and Technology of China, 610054, Chengdu, China
shenyan02@163.com
[2] Kyungwon University, Seongnam, Gyeonggi-Do 405-760, South Korea
[3] Sichuan University, 610065, Chengdu, China
guobing@sohu.com

Abstract. Combining the time-frequency location and multiple-scale analysis of wavelet transform with the nonlinear mapping and self-learning of neural networks, an error decomposition method in dynamic measuring system is proposed. According to whole-system dynamic accuracy theory, the whole-error model of dynamic measuring system is given, and then the whole-error is decomposed into sub-errors by wavelet neural networks, which are traced so that units of the system are found which generate these errors and error transmission characteristic is controlled.

1 Introduction

There are two types of dynamic measuring data in a dynamic measuring system. One is the measured value, and another is the measuring error. After the measuring error must be separated from the measured value effectively, we want to know sub-errors, through which the dynamic measuring system performance can be available by the measuring error separated. So we may improve measuring accuracy and dynamic performance by these sub-errors.

The wavelet neural networks is a new neural network model, which combines the time-frequency location and multiple-scale analysis of wavelet with the nonlinear mapping and self-learning of neural networks. It has great capability of approach, analysis and tolerance [1]. In this paper, first of all, According to whole-system dynamic accuracy theory, the whole-error model of dynamic measuring system is given. Then we use wavelet neural networks to handle and analyze error decomposition in dynamic measuring system, which are traced so that the units of the measuring system are found and error transmission characteristic can be controlled.

2 Error Model of Dynamic Measuring System

A measuring system is composed of many independent units generally, which are different in dynamic characteristic and movement rule by time. A typical dynamic measuring system is shown in Fig. 1 [2].

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1283 – 1289, 2006.

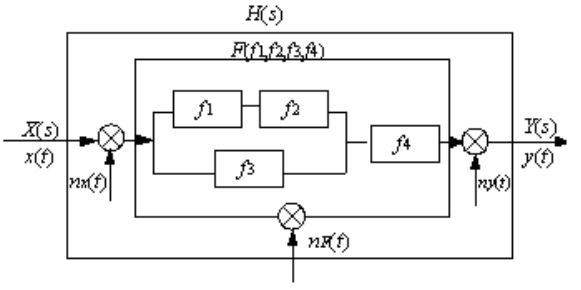


Fig. 1. The structure model of dynamic measuring system

Suppose f_i is transform function of ith unit, which is represents by parameters of its components:

$$\begin{aligned} f_i &= \upsilon\left(c_{i1}, c_{i2}, \cdots c_{in}\right) \\ i &= 1,2,\cdots n;\ j = 1,2,\cdots m \end{aligned} \tag{1}$$

Where: c_{ij} is the jth component parameter of the ith unit.

The transform function is shown in the following measuring system:

$$F\left(f_1, f_2, f_3, f_4\right) = \left(f_1 f_2 + f_3\right) f_4 \tag{2}$$

Where: f_i represents transform function of the ith unit.

So the whole-error model is given by:

$$U_y\left(t\right) = n_x\left(t\right) F\left(f_i\right) + U_F + n_F + n_y\left(t\right) \tag{3}$$

Where: $U_y\left(t\right)$ represents dynamic measuring error in the whole system;

$n_x\left(t\right)$ represents disturbed input;

$n_y\left(t\right)$ represents disturbed output;

$F\left(f_i\right)\left(i = 1,2,\cdots n\right)$ represents system transform function;

n_F represents the whole error caused by unit disturbed;

U_F represents the whole error caused by all units

Suppose the error caused by the ith unit is $e_i\left(t\right)\left(i = 1,2,\cdots n\right)$, the whole error is shown as follows:

$$U_F = \left(e_1\left(t\right) f_2 + e_2\left(t\right)\right) f_4 + e_3\left(t\right) f_4 + e_4\left(t\right). \tag{4}$$

3 Error Decomposition Based on Wavelet Neural Networks

3.1 Error Decomposition Principle

Error decomposition Principle based on wavelet neural networks is: first of all, we sample the whole error signal and produce discrete signal. Then we use the wavelet transform to exact appropriate feature vectors from the signals sampled. The feature

vectors represent sub errors that is belong to different frequency brand respectively. According to the feature vectors, these sub-errors are regard as input feature vectors of three-layer BP neural networks and output of BP neural networks is the whole error of the measuring system. The error decomposition principle is shown in Fig. 2.

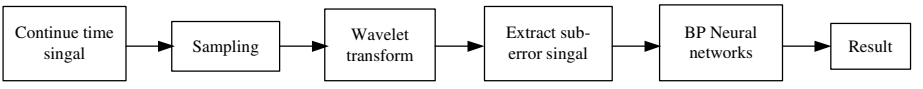


Fig. 2. Error decomposition principle

3.2 Extract Feature by Wavelet Transform

The wavelet transform has introduced a new technique for various signal processing application. The wavelet transform transforms a function by integrating it with modified versions of some kernel function [2][3]. The kernel function and its modified versions are named the "mother wavelet", and daughter functions respectively. The continuous wavelet transform decomposes any function on a family of functions obtained by dilating and translating a basis function, which takes the form:

$$\psi_{a,b}(x) = |a|^{-\frac{1}{2}} \psi\left(\frac{x-b}{a}\right) \tag{5}$$

where a is the dilation(resolution or scale), b is the translation(shift)
$\psi(x)$ is the set of all squares integral or a finite energy function is admissible if:

$$C_\psi = \int \frac{1}{|\xi|} |\hat{\psi}(\xi)|^2 \, d\xi < \infty \tag{6}$$

The continuous wavelet transform is defined as the inner product of signal and wavelets

$$W(a,b) = \left\langle f(t), \psi_{a,b}(t) \right\rangle \tag{7}$$

Suppose $a = 2^j, b = k2^j, (j,k) \in Z^2$ in the equation (5), then the equation (7) become the discrete wavelet transform, which is used to decompose a discrete signal into different resolution levels and maps a sequence of numbers into a different sequence of numbers.

If wavelet functions and scaling functions are used as building blocks to decompose and construct the signal at different resolution levels, the wavelet function will generate the detail version of the decomposed signal and scaling function will generate the approximated version of decomposed signal. The decomposition equation and algorithm can be presented in equation (8) and Fig.3:

$$\begin{cases} c_{j+1}(n) = \sum_{k=Z} h(k-2n) c_j(k) \\ d_{j+1}(n) = \sum_{k=Z} g(k-2n) c_j(k) \end{cases} \tag{8}$$

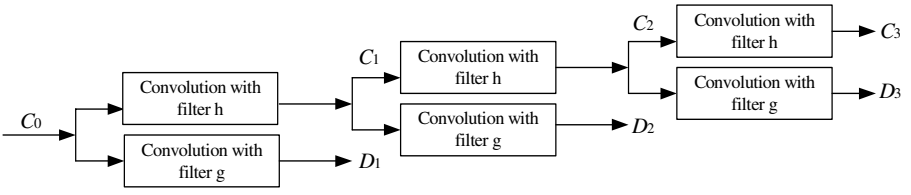


Fig. 3. The multi-resolution wavelet decomposition

So signals at various resolutions give information about different features. At a coarse resolution, we get information about the large features of signals; while at a fine resolution, we get information about more localized features of the signals.
This paper has proposed a method using wavelet transform to extract sub-errors signal, the main steps as follows:

1. Decompose the error signal with disturbance into *N* different resolution levels.
2. Rebuild wavelet packet in each different frequency band, extract 2^N pieces if frequency signal in N layer from low frequency to high frequency. Each frequency band signal shows with $S_1, \cdots S_{2^N}$ separately.
3. Construct sub-error vectors as follows:

$$E = \left[E_1, E_2, \cdots E_{2^N} \right] \tag{9}$$

3.3 The Example of Error Decomposition

A simulation system chosen are the same as fig.1.Suppose transform function of each unit is 1. Each unit produces transform error respectively as following:

$$\begin{aligned} e_1(t) &= 0.5t \\ e_2(t) &= 2\sin(4\pi t) \\ e_3(t) &= n(t) \\ e_4(t) &= \sin\left(50\pi t + \frac{\pi}{4}\right) \end{aligned} \tag{10}$$

where $e_3(t)$ is a white noise which is a zero- mean.

According to equation (3), the whole error of the system is given by

$$\begin{aligned} U_F &= \left(e_1(t) f_2 + e_2(t)\right) f_4 + e_3(t) f_4 + e_4(t) \\ &= \left(0.5t + 2\sin(4\pi t)\right) + n(t) + \sin\left(50\pi t + \frac{\pi}{4}\right) \\ &= 0.5t + 2\sin(4\pi t) + \sin\left(50\pi t + \frac{\pi}{4}\right) + n(t) \end{aligned} \tag{11}$$

The whole error signal is sampled by 100Hz sampling rate, and the whole error signal is shown in Fig.4. Then the approximate signal and detail signal at various levels are obtained by using Daubechies 6 wavelet to decompose the original sampled signals (Fig. 5). Fig.5 shows a white noise which is a zero- mean random variable of disturbance, a linear error whose slope rate is 0.5, and two sine signals, the frequency is 2Hz and 25 Hz respectively, phase different is $\pi/4$. So the results of sub-error of each unit are:

$$\begin{aligned} e_1(t) &= 0.5t + k \\ e_2(t) &= A_1 \sin(2t + \varphi_1) \\ e_3(t) &= n(t) \\ e_4(t) &= A_2 \sin(25t + \varphi_2) \end{aligned} \tag{12}$$

where $A_1, A_2, \varphi_1, \varphi_2, k$ are unknown.

We build a 4-18-1 3-BP neural networks where the sub-errors are regarded as inputting and the whole error is outputting (Fig. 6). Levenberg-Marquart optimal algorithm is used to train neural networks, and training error is 10^{-4}. So the results are shown as follows:

$$\begin{aligned} e_1(t) &= 0.4999t \\ e_2(t) &= 1.996 \sin(2t) \\ e_3(t) &= n(t) \\ e_4(t) &= 0.999 \sin(25t + 0.785) \end{aligned} \tag{13}$$

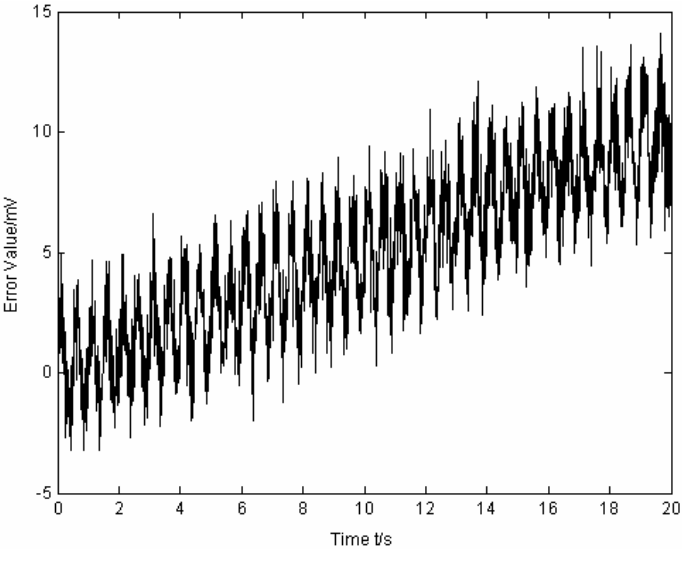


Fig. 4. Error curve of the simulated system

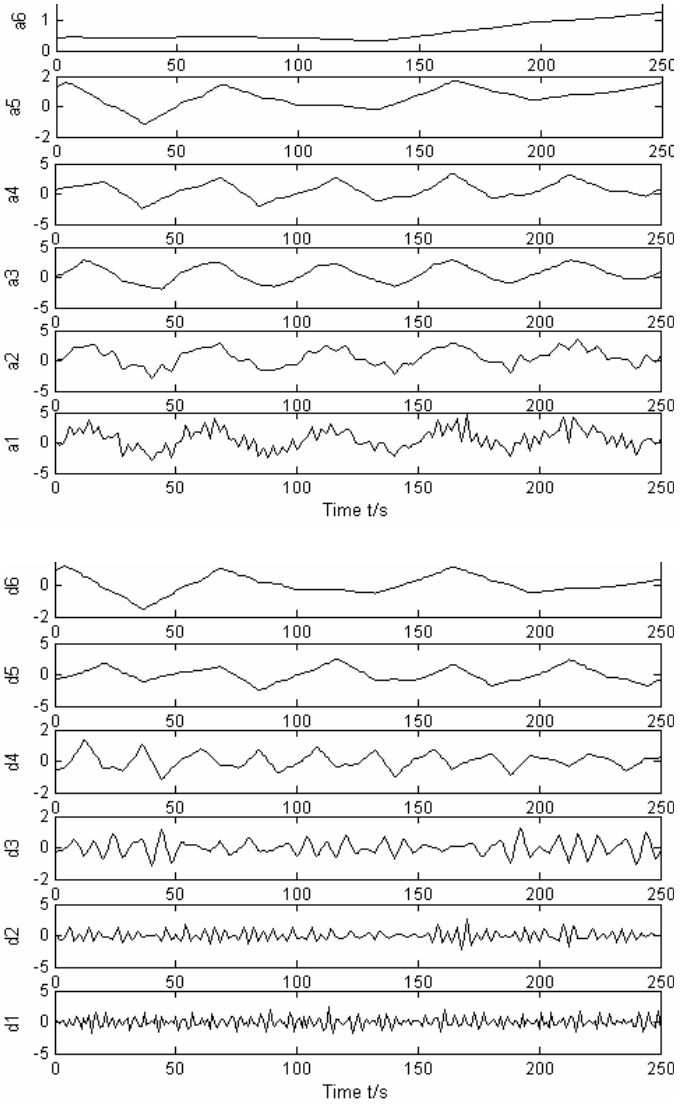


Fig. 5. Daubechies 6 wavelet decomposition

4 Conclusions

After the whole error is decomposed, we may understand transform characteristic of measuring system and master the effect the sub-error of each unit on the whole error and transformation discipline. According to the units contributing to the whole error in the system, we take effective measure to reduce or remove their passive effect in order to improve accuracy of the dynamic measuring system and meet high accurate measuring satisfy.

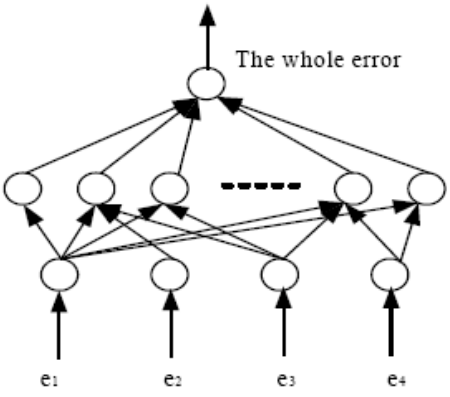


Fig. 6. 3-BP neural networks

References

1. Rushun, S.: Multi Attribute Decision Making Method And Application Based on Wavelet Neural Network. Control and Decision. 1(15) (2000) 765-768
2. Zhen, Y.X., Fei, Y.T.: Research on Error Tracing of Dynamic Measurement by Multi-Resolution Analysis. Journal of Shanghai Jiao Tong University, 37(1) (2003) 9-12
3. Daubechies, I.: The Wavelet Transform, Time-Frequency Localization and Signal Analysis. IEEE Trans. Inform. Theory. 36(5) (1990) 961-1005
4. Gaouda, A.M., Salama, M.A.: Power Quality Detection and Classification Using Wavelet-Multi-Resolution Signal Decomposition. IEEE Trans. power delivery. 14(4) (1999) 1469-1476

Maneuvering Target Tracking Based on Unscented Particle Filter Aided by Neutral Network

Feng Xue, Zhong Liu, and Zhang-Song Shi

Electronics Engineering College, Naval University of Engineering,
Wuhan 430033, China
xfmilk@sohu.com.cn

Abstract. A filtering method aided by neural network to improve the maneuvering target tracking performance is proposed in this paper. Based on unscented Kalman filter, the unscented particle filter (UPF) has more accurate proposal distribution and better approximation to non-linear tracking problem than other Sequential Monte-Carlo methods. The neural network is constructed and trained by the maneuvering features, and the outputs of NN are used as acceleration control parameters to correct model parameters. Simulation results show the performance of UPF aided by NN is much improved than extensive Kalman filter.

1 Introduction

The question of maneuvering target tracking is a nonlinear problem. A maneuvering target detection system usually contains many different kinds of sensors, of which measurement data have disparate feature and accuracy. Thus attempting to include these data efficiently will require novel multisensor data processing method. On the other hand, for the high nonlinearity of maneuvering target tracking, high tracking performance is difficult to obtain using traditional linear tracking methods, such as Interactive Multiple Model (IMM).

To solve above problems, unscented particle filter (UPF) aided by neural network (NN) ([1][2]) is presented to improve the tracking performance in this paper. Multisensor data features are extracted to fuse by NN [3], whose outputs modify the UPF parameters to compensate the tracking model's uncertainty. These construct an online tracking system for maneuvering target. Simulation results verify the effectiveness of NN-aided UPF compared with extensive Kalman filter (EKF).

2 Unscented Particle Filter in Target Tracking

First, in order to tackle the non-linearity of tracking problem, the target motion model is constructed, and the UPF is inferred to improve the accuracy of target state estimation.

2.1 Tracking Model

In the Cartesian coordinates, all nonlinearities of tracking system occur in the measurement relation. Considering the two-dimensional tracking problem, the discrete state and measurement equation are given by

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1290 – 1295, 2006.

$$\mathbf{x}_k = \boldsymbol{\Phi}_{k/k-1}\mathbf{x}_{k-1} + \boldsymbol{G}\mathbf{u}_k + \boldsymbol{\Gamma}\mathbf{w}_{k-1}, \tag{1}$$

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{v}_k), \tag{2}$$

where we define the state vector by $\mathbf{x}_k = [r_{xk}, r_{yk}, \dot{r}_{xk}, \dot{r}_{yk}]$, $\mathbf{u}_k$ is the target maneuvering control parameter consisting of the acceleration components in the x and y direction. Since the target control parameter is unknown, it is considered as process noise in the model. Then the system matrices are given by

$$\boldsymbol{\Phi}(k/k-1) = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \ \boldsymbol{\Gamma} = \begin{bmatrix} T^2/2 & 0 \\ 0 & T^2/2 \\ T & 0 \\ 0 & T \end{bmatrix}, \tag{3}$$

where $\mathbf{w}_{k-1}$ and $\mathbf{v}_k$ are Gaussian white noise with zero mean. Their covariance is calculated by

$$\boldsymbol{\Gamma}E[\boldsymbol{W}(k-1)\boldsymbol{W}(k-1)^T]\boldsymbol{\Gamma}^T = \boldsymbol{\Gamma}\boldsymbol{Q}\boldsymbol{\Gamma}^T = q\begin{bmatrix} \frac{T^4}{4}\boldsymbol{I}_{2\times 2} & \frac{T^3}{2}\boldsymbol{I}_{2\times 2} \\ \frac{T^3}{2}\boldsymbol{I}_{2\times 2} & T^2\boldsymbol{I}_{2\times 2} \end{bmatrix}, \tag{4}$$

where sample time $T = t_{k+1} - t_k$, covariance $\boldsymbol{Q} = q\boldsymbol{I}_{2\times 2}$.

2.2 Unscented Particle Filter

UPF is Unscented Kalman filter (UKF) applied in Sequential Monte Carlo methods (also known as particle filter) ([4][5]). UKF is able to more accurately propagate the mean and covariance of the Gaussian approximation to the state distribution than EKF [6]. In the target tracking problem based on particle filter, UKF can provides heavy tailed distributions as the proposal distribution generator [7]. This section describes UPF, which is used to estimate the target state vector. The UPF process is given by:

Step 1: Initialization. (Sampling time k=0)

Sample N particles from the prior $p(\mathbf{x}_0)$, we can obtain data set $\{\mathbf{x}_0^i\}$:

$$\overline{\mathbf{x}}_0^{ia} = [(E[\mathbf{x}_0^i])^T, 0, 0]_{n_a\times 1}^T, \boldsymbol{P}_0^{ia} = E[(\mathbf{x}_0^{ia} - \overline{\mathbf{x}}_0^{ia})(\mathbf{x}_0^{ia} - \overline{\mathbf{x}}_0^{ia})^T] = \begin{bmatrix} \boldsymbol{P}_0 & 0 & 0 \\ 0 & \boldsymbol{Q} & 0 \\ 0 & 0 & \boldsymbol{R} \end{bmatrix}_{n_a\times n_a}. \tag{5}$$

Step 2: On-line iteration. (k=1, 2, 3…)
Step (2a) Calculate sigma points:

$$\chi_{k-1}^{ia}=\left[\begin{array}{ccc}\mathbf{x}_{k-1}^{ia} & \mathbf{x}_{k-1}^{ia}+\sqrt{(n_a+\kappa)\boldsymbol{P}_{k-1}^{ia}} & \mathbf{x}_{k-1}^{ia}-\sqrt{(n_a+\kappa)\boldsymbol{P}_{k-1}^{ia}}\end{array}\right]. \tag{6}$$

Step (2b) Propagate particle into future (time update):

$$\chi_{k-1}^{ia}=\left[\begin{array}{ccc}\mathbf{x}_{k-1}^{ia} & \mathbf{x}_{k-1}^{ia}+\sqrt{(n_a+\kappa)\boldsymbol{P}_{k-1}^{ia}} & \mathbf{x}_{k-1}^{ia}-\sqrt{(n_a+\kappa)\boldsymbol{P}_{k-1}^{ia}}\end{array}\right], \tag{7}$$

$$\chi_{k|k-1}^{ix}=\boldsymbol{\Phi}_{k/k-1}\chi_{k-1}^{ix}+\boldsymbol{\Gamma}\chi_{k-1}^{w} \quad \hat{\mathbf{x}}_{k|k-1}^{i}=\sum_{j=0}^{2n_a}W_j^m\chi_{i,k|k-1}^{ix}, \tag{8}$$

$$\mathbf{z}_{k|k-1}^{i}=h(\chi_{k|k-1}^{ix},\chi_{k|k-1}^{iv}) \quad \hat{\mathbf{z}}_{k|k-1}^{i}=\sum_{j=0}^{2n_a}W_j^m\mathbf{z}_{j,k|k-1}^{i}, \tag{9}$$

$$\boldsymbol{P}_{k|k-1}^{i}=\sum_{j=0}^{2n_a}W_j^c[\chi_{i,k|k-1}^{ix}-\hat{\mathbf{x}}_{k|k-1}^{i}][\chi_{i,k|k-1}^{ix}-\hat{\mathbf{x}}_{k|k-1}^{i}]^T, \tag{10}$$

where χ^{ix}, χ^{iw} and χ^{iv} present the corresponding weights of state vector, process noise and measurement noise respectively.
Step (2c) Incorporate new observation (measurement update):

$$\boldsymbol{P}_{z_{k|k-1}z_{k|k-1}}=\sum_{j=0}^{2n_a}W_j^c[\boldsymbol{Z}_{j,k|k-1}^{i}-\hat{\mathbf{z}}_{k|k-1}^{i}][\boldsymbol{Z}_{j,k|k-1}^{i}-\hat{\mathbf{z}}_{k|k-1}^{i}]^T, \tag{11}$$

$$\boldsymbol{P}_{x_{k|k-1}z_{k|k-1}}=\sum_{j=0}^{2n_a}W_j^c[\chi_{i,k|k-1}^{ix}-\hat{\mathbf{x}}_{k|k-1}^{i}][\boldsymbol{Z}_{j,k|k-1}^{i}-\hat{\mathbf{z}}_{k|k-1}^{i}]^T, \boldsymbol{K}_k=\boldsymbol{P}_{x_{k|k-1}z_{k|k-1}}\boldsymbol{P}_{z_{k|k-1}z_{k|k-1}}^{-1}, \tag{12}$$

$$\hat{\mathbf{x}}_{k}^{i}=\hat{\mathbf{x}}_{k|k-1}^{i}+\boldsymbol{K}_k(\mathbf{z}_k-\hat{\mathbf{z}}_{k|k-1}^{i}), \boldsymbol{P}_{k|k}^{i}=\boldsymbol{P}_{k|k-1}^{i}-\boldsymbol{K}_k\boldsymbol{P}_{z_{k|k-1}z_{k|k-1}}\boldsymbol{K}_k^T, \tag{13}$$

where state estimation and covariance of proposal density is inferred.
Step (2d) Particle sampling and calculating weight:

$$\mathbf{x}_k^i \sim q(\mathbf{x}_k^i \mid \mathbf{x}_{k-1}^i,\mathbf{z}_k)=N(\hat{\mathbf{x}}_k^i,\mathbf{P}_{k|k}^i), \tag{14}$$

$$w_k^i \propto \frac{p(\mathbf{z}_k \mid \mathbf{x}_k^i)\,p(\mathbf{x}_{0:k-1}^i \mid z_{1:k-1})\,p(\mathbf{x}_k^i \mid \mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i \mid \mathbf{x}_{0:k-1}^i,\mathbf{z}_{1:k})\,q(\mathbf{x}_{0:k-1}^i \mid \mathbf{z}_{1:k-1})}. \tag{15}$$

Step (2e) Resampling:
According to $Pr(\mathbf{x}_{k|k}^{(i)}=\mathbf{x}_{k|k-1}^{(j)})=\tilde{w}_k^{(j)}$, new particle set $\{\mathbf{x}_{1:k}^i, w_{1:k}^i\}$ can be obtained.

Step 3: State vector estimation.
The outputs of the Step (e) can be used to approximate the state vector as follows:

$$\hat{\mathbf{x}}_k=\mathrm{E}[\mathbf{x}_k]=\int\mathbf{x}_k\,p(\mathbf{x}_k \mid \boldsymbol{Z}_{1:k})\approx\sum_{i=1}^{n}w_k^i\mathbf{x}_k^i. \tag{16}$$

3 Maneuvering Target Aided by Neural Networks

Under the model-invariant tracking assumption, high accurate constant velocity target tracking results can be obtained by UPF. However, the process and magnitude of the acceleration are always sudden and unknown. Although a higher value of the process noise could be used, the tracking performance will be degraded. Thus, NN is incorporated into UPF to obtain the maneuvering control parameters.

3.1 Structure of NN-Aided Tracking System

The task of NN is to find the relationship between maneuvering features and acceleration control parameters. As shown in Fig 1 the feed forward NN is constructed including three layers. Three kinds of maneuvering feature vectors are acted as NN inputs, and acceleration vector $\hat{\mathbf{u}}_k$ as output. We used the sigmoid function $tanh(x)$ in the hidden units and the linear function $f(x) = x$ in the input and output units.

First input feature. In the radar measurement data, the first type of acceleration feature at time k can be extracted as follows:

$$f_1(k) = \frac{(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1})^2}{\boldsymbol{P}_{z_{k|k-1} z_{k|k-1}}}, \tag{17}$$

where $\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}$ denotes the innovation of measurement, $\boldsymbol{P}_{z_{k|k-1} z_{k|k-1}}$ is given by (13).

Second input feature. The second input feature comes from the target course change, which can be calculated by UPF estimation. That is:

$$f_2(k) = \frac{\dot{r}_{xk}}{\dot{r}_{yk}} - \frac{\dot{r}_{x(k-1)}}{\dot{r}_{y(k-1)}}, \tag{18}$$

where $\dot{r}_{xk}$ and $\dot{r}_{yk}$ denote the velocity components of estimated state vector.

Three input feature. The Doppler shift reflects the change in the range rate. By the range rate the change of velocity can be gotten. f_3 is as follows:

$$f_3(k) = \frac{2}{\lambda \sigma_{sf}^2} (\dot{D}_k - \dot{D}_{k-1}), \tag{19}$$

where $\dot{D}_k$ is the rang rate, σ_{sf}^2 the variance of Doppler shift, and λ the wavelength of the transmitted wave.

The above three features contain position, course and velocity change information that is adequate to train NN. A cost function can be defined as follows:

$$J_k = E\{\| u_k - \hat{u}_k \|\} \rightarrow \text{minimum}. \tag{20}$$

By several trajectories of typical target maneuvers, NN is trained off-line with three features with known maneuvering control parameters.

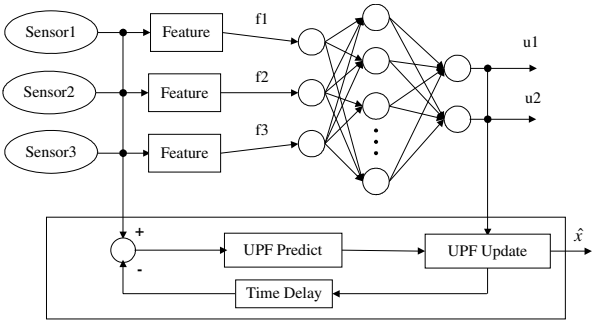


Fig. 1. Structure of NN-aided UPF Tracking System

4 Training and Simulation Results

A simulated maneuvering target is generated to give the noisy measurements. The original motion parameters of the target are $\boldsymbol{x}_0 = [5000, 8660, 10, 6]$, state variances $\boldsymbol{P}_0 = diag[100\times10^3, 100\times10^3, 0.1\times10^3, 0.1\times10^3]$. The target performs random constant acceleration and coordinated turn movement.

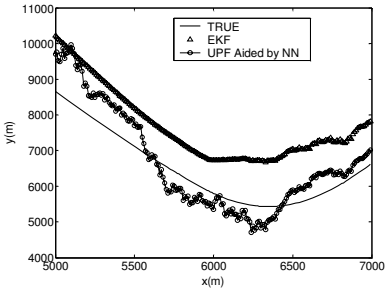


Fig. 2. After 100 Monte Carlo simulations, trajectory tracking comparison of *IMM-EKF* and *NN-aided UPF* with N = 800 particles

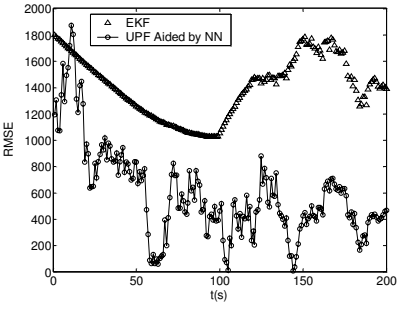


Fig. 3. The tracking range RMSE of *IMM-EKF* and *NN-aided UPF*

The method of the NN-aided UPF presented is right through the simulation results compared with IMM-EKF. NN-aided UPF yields more accurate state estimation than IMM-EKF. After the target maneuvering, IMM-EKF tracking lags behind the target and presents bias feature in trajectory tracking, whereas NN-aided UPF can also track the target without much delay. However NN-aided UPF cost much time in the initial tracking period, thus the future researches should focus on how to increase the convergence speed to realize rapid tracking.

References

1. Takaba, K., Iiguni, Y., Tokumaru, H.: An Improved Tracking Kalman Filter Using a Multilayered Neural Network. Elsevier Science M&L. Compute Modeling, **23** (1996) 119-128
2. Armstrong, B., Holeman, B., S.: Target Tracking with a Networks of Doppler Radar. IEEE Trans on AES., **34** (1998) 33-48
3. Wong, Y.C.: Target Tracking in A Multisensor Environment Using Neural Networks. Doctoral Paper, the University of Arizona, Arizona (2000)
4. Gordon, N. J., Salmond, D. J., Smith, A. F. M.: Novel Approach to Nonlinear/non-Gaussian Bayesian State Estimation, IEE-Proceedings-F, **140** (2) (1993) 107–113.
5. Merwe, R.V., Doucet, A., Freitas, N.: The Unscented Particle Filter. Technical Report CUED/ F2INPENG/ TR 380, Cambridge University Engineering Department, Cambridge (2000)
6. Doncet, A., Freitas, J., and Gordon, N. J. (eds.): Sequential Monte Carlo Methods in Practice, Springer, New York (2001)
7. Joseph, J., LaViola, J.: A Comparison of Unscented and Extended Kalrnan Filtering for Estimating Quaternion Motion. Proceedings of the American Control Conference, (2003) 2435-2440

Application of Principal Component-Artificial Neural Networks in Near Infrared Spectroscopy Quantitative Analysis

Hai-Yan Ji[1] and Zhen-Hong Rao[2]

[1] College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China
yuntian@cau.edu.cn
[2] College of Science, China Agricultural University, Beijing 100094, China

Abstract. The principal components of near infrared spectroscopy were calculated by nonlinear iterative partial least squares (NIPALS). The best number of principal components was determined by cross-validation method. Thus, limited principal components that free from noise and orthogonal each other were obtained. After standardization, these principal components were used as input nodes of back propagation artificial neural networks (B-P ANN). ANN was used to build nonlinear model. In the method, the data of whole spectra can be fully utilized, the best principal components free from noise and nonlinear model obtained, the iterative time of B-P ANN shorted strongly, and better calibration model can be obtained. The method has been applied to quantitatively determine the starch of barely. The calibration and prediction correlation coefficients are 0.982 and 0.945; the relative standard deviations are 1.81% and 2.80%, respectively.

1 Introduction

The vibration absorptions in the near infrared are overtone and combination frequencies of absorption bands that occur in the mid-infrared. These overtones and combinations cause overlapping absorption bands, which make it difficult to visually identity chemical groupings of a molecule from their near infrared spectra. The overtone and combination bands are one to three orders of magnitude weaker than the fundamental bands. With the development of computer's technology and introduction of chemometrics, the analysis problem of this spectral range was solved, thus, near infrared spectroscopy was succeed applied in quantitative analysis. Near infrared spectroscopy has a series of advantages in quantitative analysis, such as analysis speed is faster, sample prepare is simple, may analysis multi-component from single spectrum, non-destroy sample, no chemical pollution etc. Near infrared spectroscopy has wide-ranging application in agriculture product analysis, food component determination, forage, pharmaceuticals, petrochemical product analysis etc. [1-3].

For building near infrared spectroscopy quantitative analysis calibration equation, some method such as stepwise regression analysis, principal component regression, and partial least square (PLS) were general used to build linear model between chemical values and spectral parameters of sample [4,5]. But in the near infrared

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1296 – 1303, 2006.

spectroscopy, some nonlinear exist between the spectral parameters and chemical values, it's nonlinear especially obvious when the concentration range of sample is width. Artificial neural networks (ANN) were used to build the nonlinear model. Long has use ANN in quantitative analysis of near infrared spectroscopy [6]. In the quantitative analysis of ANN, the amount of input nodes in ANN should not too much, otherwise, its iterative time would be longer, so wavelength selected methods were used. We have used stepwise regression analysis method to select the wavelengths of cereal's Fourier transform near infrared diffuse reflectance spectroscopy, used these selected wavelengths as the input nodes of ANN, to quantitative analysis the components of cereal [7]. But the select wavelength method does not fully utilized the spectral information of the whole spectra, may lose some information. In this article, we use nonlinear iterative partial least squares (NIPALS) to calculate the principal component of sample spectra, use cross-validation method to determine the best number of principal component. These principal components were used as input nodes of ANN, to build the quantitative analysis model. In this method, limited principal component free from noise and orthogonal each other were obtained. The method not only fully utilized the data of whole spectra, but also shorted the iterative time strongly, and build better calibration model. Starch of barely was quantitatively analysis by the method, and the results were satisfied.

2 Theory

2.1 Calculating Principal Components of Sample Spectra Using NIPALS Algorithm

Suppose, A is absorbance matrix of n calibration samples on m wavelength, E is statistical residual matrix, using the nonlinear iterative partial least squares (NIPALS) algorithm, divided absorbance matrix A into the product of the score matrix T and the loading matrix W:

$$A_{n\times m} = T_{n\times d} \bullet W_{d\times m} + E_{n\times m} \ . \tag{1}$$

Where, are orthogonal each other respectively among the column variables of T, and among the row variables of W. Where d is the best dimension, which is determined by the cross-validation method. Every column of T is a principal component vector of samples, thus, d is the number of principal components.

2.2 Determined the Number of Principal Components by Cross-Validation Method

For one dimension H, select k samples as prediction from n calibration samples, using remain n-k samples to set up calibration equation, which is using to prediction this k samples. These k samples is reinstated and another k samples are selected as prediction, repeat the process as above. Through calibration and prediction repeatedly, until these entire sample be predicted one time, thus obtain total error sum of squares:

$$\delta(H) = \sum_{i=1}^{n} (C_{iP} - C_{iC})^2 \quad . \tag{2}$$

Where H is one dimension, C_{iP} represents prediction concentration; C_{iC} represents the known concentration.

For every dimension, calculate the total error sum of squares $\delta(H)$, compare these $\delta(H)$, the H corresponding with the minimum of total error sum of squares, is the best dimension (the number of principal component) d.

2.3 Artificial Neural Networks

A feed-forward network was constructed by using three layers of nodes: an input layer, a hidden layer, and an output layer, as shown in figure 1. The input nodes transfer the weighted input signals to the nodes in the hidden layer. A connection between node i in the input layer and node j in the hidden layer is represented by the weighting factor W_{ji}. These weights are adjusted during the learning process.

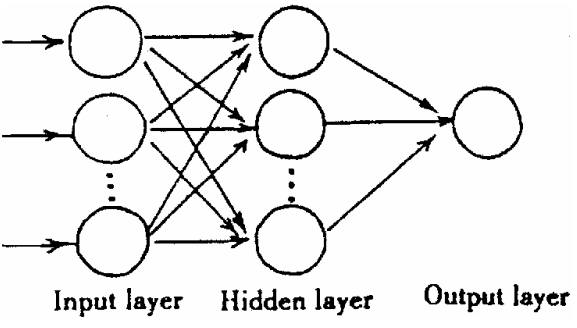


Fig. 1. Three layers B-P ANN for calibration and prediction of content of starch in barley

The output of each hidden node is a sigmoid function of the sum of that node's weighted inputs. The sigmoid function shown in equation (3)

$$f(x) = \frac{1}{1+e^{-x}} \quad . \tag{3}$$

The outputs from each node in the hidden layer are sent to each node in the output layer. For our calibration applications, only one output node was used in the output layer, having an output node equal to the scaled concentration of the component of interest.

The error in the expected output is back-propagated through the network by using the generalized delta rule to determine the adjustments to the weights. When a linear output function is used, the output layer error term is given by

$$\delta_{pk} = t_{pk} - o_{pk} \quad . \tag{4}$$

where δ_{pk} is the error term for observation p at output node k, t_{pk} is the expected output for observation p, and o_{pk} is the actual node output.

The error term at node j of the hidden layer that uses a sigmoid transfer function is the derivative of the sigmoid function multiplied by the sum of the products of the output error terms and the weights in the output layer

$$\delta_{pj} = o_{pj}(1-o_{pj})\sum_{k=1}^{K}\delta_{pk}\omega_{kj} \quad . \tag{5}$$

The error terms from the output and hidden layers are back-propagated through the network by making adjustments to the weights of their respective layers. Weight adjustments, or delta weights, are calculated

$$\Delta\omega_{ji}(n) = \eta\delta_{pj}o_{pi} + \alpha\Delta\omega_{ji}(n-1) \quad . \tag{6}$$

Where $\Delta\omega_{ji}$ is the change in the weight between node j in the hidden layer and node i in the input layer. In EQ 6, η is the learning rate, δ_{pj} is the error term for observation p at node j of the hidden layer, o_{pi} is the observed output for node i of the input layer for observation p, and α is the momentum. The terms n and n-1 refer to the present iteration and the previous iteration, respectively.

An equation similar to EQ 6 is used to adjust the weights connecting the hidden layer of nodes to the nodes in the output layer.

3 Experiment

The 68 barley samples were ground and passed through 0.45mm sieve. Pure barium sulphate ($BaSO_4$) was used as the black (reference) sample. The near infrared spectra were scanned on dispersion near infrared spectrometer. Each sample was scanned 3 times and the mean was calculated as the spectrum of the sample. The spectrum of barley is shown in figure 2. The wavelength range is from 1480nm to 2500nm. Interval 8nm taken one data point; therefore, there are 128 data points contained in each spectrum.

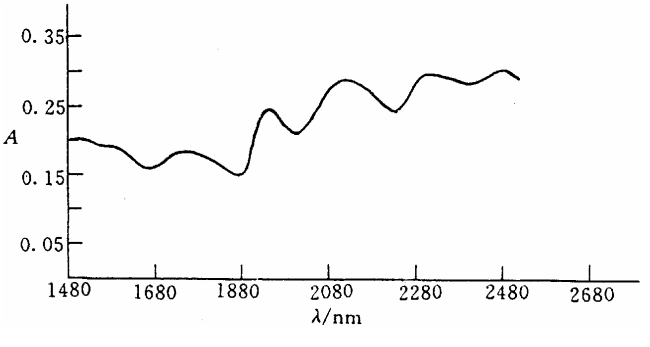


Fig. 2. The near infrared spectrum of barley sample

The standard starch concentrations of 68 barley samples were determined using the traditional optical rotation method. Many times measurement was conducted to ensure the reliability of results. The results of chemical determination were regarded as standard values, and the results of near infrared calculation were regarded as prediction values.

40 samples were used as calibration set, which was used to establish calibration equation between starch concentrations of barely and near infrared spectra. Another 28 samples were used as prediction set, which was used to test the reliability of calibration equation.

4 Results and Discussion

By cross-validation, the best dimension is 8, as shown in figure 3. Thus, through NIPALS algorithm, 8 principal components of starch in barely samples were calculated. After standardization, these 8 principal components were used as the values of input nodes in ANN. The parameters of ANN are list in table 1.

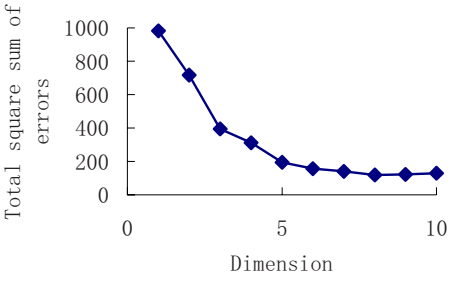


Fig. 3. Determine dimension (the number of principal component)

Table 1. The parameters of ANN

Number of nodes			Learning rate	Momentum
Input layer	Hidden layer	Output layer		
8	6	1	0.15	0.25

After minutes, the best results are obtained when the times of iteration reaches 250. The correlation coefficient and relative standard deviation of starch in calibration set and prediction set are list in table 2. These results are super to the results of PLS that we calculated before.

Table 2. The correlation coefficient (R) and relative standard deviation (RSD) of starch

Sample set	Sample number	correlation coefficient	RSD(%)
Calibration set	40	0.982	1.81
Prediction set	28	0.945	2.80

The plots of scatter points for starch of barely in calibration set and prediction set are shown in figure 4 and figure 5 respectively.

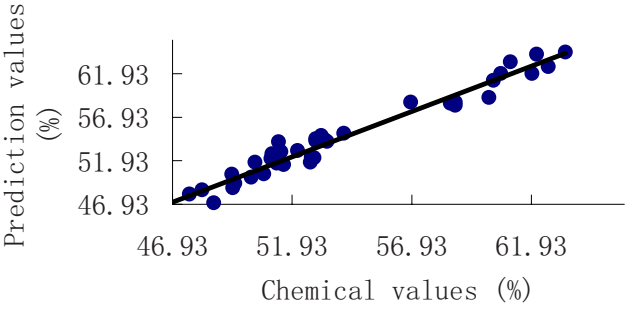


Fig. 4. The plot of scatter points for starch of barely in calibration set

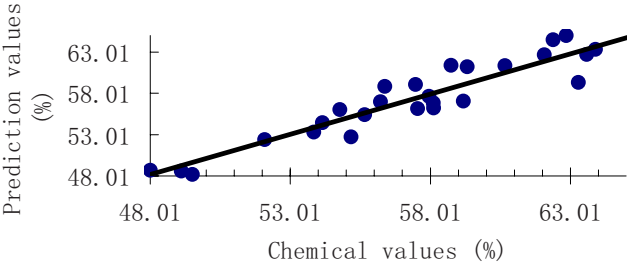


Fig. 5. The plot of scatter points for starch of barely in prediction set

The chemical values, prediction values and relative errors of samples in prediction set are list in table 3.

Some nonlinear is existed between near infrared spectroscopy and concentration of sample components, partial least squares (PLS) was used to build linear model, whereas ANN was used to build nonlinear model, thus, the results of ANN are super to the results of PLS.

There are 128 data points in every sample, if all these data points were used as input nodes of ANN, the structure of ANN would be huge, and the calculated speed would be slower.

Using NIPALS to calculate principal component, and using cross-validation to determine the best dimension, thus, the best number of principal components free from noise were obtained and the principal components were orthogonal each other. The data of whole spectra can be fully utilized, and limited best orthogonal principal components were obtained. Using calculated principal components as the values of input nodes in ANN, the iterative time of B-P ANN shorted strongly.

Table 3. The chemical values, prediction values and relative errors of samples in prediction set

No.	Chemical values	Prediction values	Relative errors (%)
1	53.85	53.30	–1.0
2	49.51	48.20	–2.6
3	55.18	52.73	–4.4
4	48.01	48.70	1.4
5	58.11	56.85	–2.2
6	63.90	63.31	–0.9
7	58.75	61.40	4.5
8	57.55	56.16	–2.4
9	59.19	57.06	–3.6
10	60.67	61.37	1.1
11	54.78	56.06	2.3
12	54.16	54.46	0.6
13	49.12	48.59	–1.1
14	55.66	55.43	–0.4
15	52.09	52.42	0.6
16	56.23	56.98	1.3
17	62.86	65.00	3.4
18	57.48	59.08	2.8
19	62.08	62.67	1.0
20	62.39	64.50	3.4
21	65.55	64.68	–1.3
22	58.12	56.22	–3.3
23	63.59	62.72	–1.4
24	65.49	64.00	–2.3
25	57.96	57.65	–0.5
26	59.33	61.21	3.2
27	56.38	58.84	4.4
28	63.29	59.33	–6.3

5 Conclusions

We demonstration one method that using NIPALS to calculate the principal components of near infrared spectroscopy, using cross-validation to determine the best dimension, and using ANN to build nonlinear model. The advantage of the method is that the data of whole spectra can be fully utilized, the iterative time of ANN can be shorted strongly, and better calibration model can be obtained. The method applied to quantitatively determine the starch of barely, the results were satisfactory.

References

1. Martin, K.A.: Recent Advanced in Near-infrared Spectroscopy On-line. Applied Spectroscopy Review 27 (4) (1992) 325-383
2. Mcclure, W. F.: Near-infrared Spectroscopy: The Giant is Running Strong. Anal Chem. 66 (1) (1994) 43A-53A

3. Jerore, J., Workman, J.R.: Review of Process and Non-invasive Near-infrared and Infrared spectroscopy: 1993-1999. Applied Spectroscopy Reviews 34 (1&2) (1999) 1-89
4. Williams, P.C., Preston, K.R., Norris, K.H., and Starkey, P.M.: Determination of Amino Acids in Wheat and Barley by Near-infrared Reflectance Spectroscopy. Journal of Food Science 49 (1) (1984) 17-20
5. Robert, P., Bertrand, D., Devaux, M.F., Grappin, R.: Multivariate Analysis Applied to Near-infrared Spectra of Milk. Anal Chem. 59 (16/17) (1987) 2187-2191
6. Long, J.R., Grogorion, V.G., Gemperline, P.J.: Spectroscopic Calibration and Quantization Use Artificial Neural Networks. Anal Chem. 62 (17) (1990) 1791-1797
7. Ji, H.Y., Yan, Y.L.: The Quantitative Analysis for Cereal's Components by Artificial Neural Networks. Chemical Journal of Chinese Universities 14 (5) (1993) 618-620

Application of Neural Networks for Integrated Circuit Modeling

Xi Chen[1,2], Gao-Feng Wang[1,2], Wei Zhou[1,2], Qing-Lin Zhang[1,2], and Jiang-Feng Xu[3]

[1] School of Electronic Information, Wuhan University, China
[2] Institute of Microelectronics and Information Technology, Wuhan University, China
[3] Shanghai Institute of Technical Physics, Chinese Academy of Sciences, China
robertchenxi@yahoo.com.cn

Abstract. Application of feedforward neural networks for integrated circuit (IC) modeling is presented. In order to accurately describe IC behaviors, a set of improved equations for dynamic feedforward neural networks has been utilized for IC modeling. The rationality of the improved equations is elucidated by analyzing the relation between the circuits and the equation parameters. Through some special choices of the neuron nonlinearity function, the feed-forward neural networks can themselves be represented by equivalent circuits, which enables the direct use of neural models in existing analogue circuit simulators. Feedforward neural network models for some static and dynamic systems are obtained and compared. Simulated results are included to illustrate the accuracy of the neural networks in circuit modeling.

1 Introduction

In electronics industry, engineers increasingly rely on advanced computer-aided design (CAD) or electronic design automatic (EDA) tools to help them with synthesis and verification of complicated electronic system designs. The circuit simulator software, such as Berkeley SPICE, Mentors' Eldo, and Philips' Pstar, is recognized as the key EDA tool, especially for analogue integrated circuit designs.

In integrated circuit simulation, one needs to perform two major modeling tasks. The first modeling task is to develop new models for emerging devices, whereas the second modeling task is to search for more efficient yet accurate models to replace existed device models or sub-circuits, which are often called macro-models.

In the first modeling task, one often focuses more on model availability than model efficiency, that is, obtaining usable device models to enable the circuit simulation is the primary goal. A classical approach to get a usable model is to make use of available physical knowledge, and forge that knowledge into a numerically well-behaved model. Such a numerical well-behaved model is commonly called as physical model. The relation between the underlying device physics and physical structure remains a very important asset of such handcrafted models. In general, it takes years to develop a good physical model for a new device.

In the second modeling task, one often focuses on model efficiency, that is, to obtain computationally efficient models without compromising its accuracy is the primary goal. To achieve this primary goal, one could obtain discrete behavioral data

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1304 – 1312, 2006.

from device measurements or simulations. These discrete data can be put into lookup tables. The merits of this kind lookup table approach and a number of useful techniques to establish lookup tables are discussed in [1], [2]. This kind of modeling schemes is commonly called as table modeling technique. A very important advantage of the table modeling techniques is that one can, in principle, obtain a quasi-static model of any desirable accuracy by providing a sufficient amount of discrete data. Table modeling can be applied without the risk of finding a poor fit due to some local minimum resulting from optimization. However a major pitfall is that a single quasi-static model cannot express all kind of behaviors relevant to devices and sub-circuits under modeling.

Generally speaking, the computational time in a circuit simulation is dominated by the number of devices and the time to simulate each device. The model efficiency for devices is extremely crucial to speedup circuit simulations. With a great advancement of artificial neural networks, more attentions have been attracted to build circuit models by employing the neural networks [3].

In this work, the possibility to use feedforward neural networks for integrated circuit (IC) modeling will be explored. A set of improved equations for both static and dynamic feedforward neural networks will be utilized. The specific choices for the neuron nonlinearity function, which expresses some behaviors of the (sub-)circuits, will be discussed. It will be shown how the feedforward neural networks can themselves be represented by equivalent electrical circuits, which enables the direct use of neural models in existing analogue circuit simulators. Examples for modeling some static and dynamic systems are included, the examples illustrate that the improved feedforward neural networks are very accurate for circuit modeling.

2 Problem Statements and Definitions

2.1 Circuit Equations

In general, many of electronic circuits can be mathematically described by a system of nonlinear first order differential equations:

$$\boldsymbol{f}(\mathbf{y}(t), \frac{\mathrm{d}\mathbf{y}(t)}{\mathrm{d}t}, \mathbf{p}) = 0 \tag{1}$$

where the real-valued vector $\boldsymbol{y}$ represents electrical input signals, internal variables, and output signals at time t, which could be charges, fluxes, currents or voltages. The real-valued vector $\boldsymbol{p}$ denotes the circuit and device parameters, which may represent component values for resistors, inductors and capacitors, or any other quantities determined by the particular choice of circuit design and manufacturing process.

2.2 Improved Equations for Feedforward Neural Networks

Models using neural networks have demonstrated superior performance in both efficiency and universal approximation, without compromising accuracy. In particular, the classic feedforward neural networks were widely employed to build models in circuit simulations [3]. In order to accurately represent the behaviors of

integrated circuits, improved equations for feedforward neural networks have been introduced as follow:

$$y_{ik} = \boldsymbol{f}(s_{ik}, \alpha_{ik}) \tag{2}$$

$$s_{ik} = \sum_{j=1}^{N_{k-1}} w_{ijk}\, y_{j,k-1} - \theta_{ik} + \sum_{j=1}^{N_{k-1}} v_{ijk}\, \frac{\mathrm{d}y_{j,k-1}}{\mathrm{d}t} \tag{3}$$

Comparing (2) and (3) with the classical static feedforward neural networks equations [4], an additional weight parameter v_{ijk} plays an important role for the frequency dependent part of the connection strength. It is a weight for the rate of change $\mathrm{d}y_{j,k-1}/\mathrm{d}t$ in the output of neuron j in layer k-1. The multiplication of the time derivative $\mathrm{d}y_{j,k-1}/\mathrm{d}t$ and the weight v_{ijk} determines the dynamic response of neuron i in layer k (in practical discrete system, $(y_{j,k-1}(n) - y_{j,k-1}(n-1))/\Delta t$ is used to displace $\mathrm{d}y_{j,k-1}/\mathrm{d}t$). Moreover, $\boldsymbol{f}$ is a vector function that is assumed to be the same for all neurons within the networks. Parameter α_{ik} can be utilized to set an appropriate scale of change in qualitative transitions in function behavior. Thus $\boldsymbol{f}$ for neuron i in layer k takes the form $\boldsymbol{f}(s_{ik}, \alpha_{ik})$, which reduces to $\boldsymbol{f}(s_{ik})$ for functions that don't depend on α_{ik} .

A further reason for the combination of w_{ijk} and v_{ijk} lies in the fact that it simplifies the representation of the (sub-)circuits, which consist of resistors and capacitors. The total current, consisting of the dc current and the time derivative of the diffusion charge, is then obtained by first calculating a bias-dependent nonlinear function with a value proportional to the dc current. In the subsequent neural network layer, this function is weighted by w_{ijk} to add the dc current to the net input of a neuron, and its time derivative is weighted by v_{ijk} to add the capacitive current to the net input.

2.3 Dynamic Feedforward Neural Networks

The static feedforward networks, as a special case of the dynamic feedforward neural networks, have a general capability for representing any continuous multidimensional static behavior up to any desired accuracy with a nonlinear, monotonically increasing and bounded continuous function $\boldsymbol{f}$, requiring not more than one hidden layer. More literature on the capabilities of neural networks and fuzzy systems as universal static approximations can be found in [5], [6].

In order to represent some dynamic behaviors, one should employ the dynamic networks instead of the static networks. One of the most common cases for a build-in circuit model is that dc terminal currents $I^{(dc)}$ and equivalent terminal charges Q of a device are directly and uniquely determined by an externally applied time-dependent voltage $V(t)$, as depicted on the left of Fig. 1. This is typical in quasi-static modeling of MOSFETs to get rid of the non-quasistatic channel charge distribution. The actual quasi-static terminal currents of a device model with parameters P are then given by

$$I(t) = I^{(dc)}(V(t), P) + \frac{\mathrm{d}}{\mathrm{d}t} Q(V(t), P) \tag{4}$$

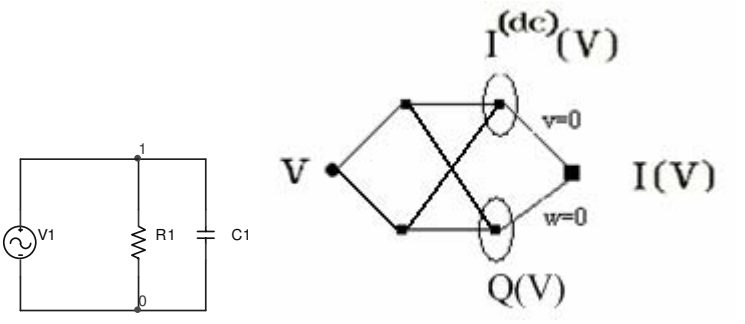


Fig. 1. Differential circuit and equivalent dynamic feedforward neural networks

The left circuit of Fig. 1 can be mapped into the right dynamic feedforward networks. The input of layer 3 is the desired outcome and must therefore be transparently passed on to the network outputs by using linear behavior in f. A number of nonzero w_{ij3} and zero v_{ij3} values are used to copy the dc currents into the net input s_{i3} of output neurons in this layer. Zero w_{ij3} and nonzero v_{ij3} values are used to add the appropriate time derivatives of the charges, as given by the outputs of other neurons in layer 2.

This simple constructive procedure shows that some of the devices and (sub-) circuits can be represented to arbitrary accuracy by the improved dynamic neural networks [7]. It does not exclude the possibility that the same may also be possible with two hidden layers.

2.4 Mapping Function f into Circuit

Two special choices for the neuron nonlinearity f are typical for semiconductor devices and circuits. The first one is the familiar logistic function f_1 [8], [9], as illustrated on the left of Fig. 2, which is strictly monotonically increasing with s_{ik}:

$$f_1(s_{ik}) = \frac{1}{1+e^{-s_{ik}}} \tag{5}$$

on the right of Fig. 2, I is the current through a series connection of two identical ideal diodes, having the cathodes wired together at an internal node with voltage V_0, whereas V is the voltage across the series connection [10], [11]. The relation between I and logistic function f_1 can be derived as follows:

$$I = I_s(e^{(V-V_0)/V_t} - 1) = -I_s(e^{-V_0/V_t} - 1) \tag{6}$$

$$V_0 = V_t \ln(\frac{1+e^{V/V_t}}{2}) \tag{7}$$

$$f_1(V/V_t) = \frac{1}{2}(\frac{I}{I_s} + 1) \tag{8}$$

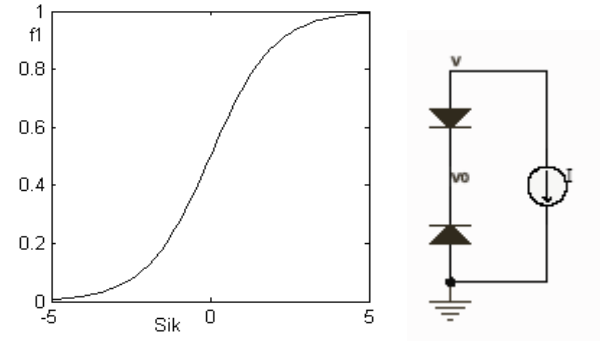


Fig. 2. Logistic function f_1 and equivalent circuit

The other choice for f, which uses the argument α_{ik} to control the sharpness of the transition between linear and exponential behavior, is taken as follows:

$$f_2(s_{ik},\alpha_{ik}) = \frac{1}{\alpha_{ik}^2} \ln \frac{\cosh \frac{\alpha_{ik}^2 (s_{ik}+1)}{2}}{\cosh \frac{\alpha_{ik}^2 (s_{ik}-1)}{2}} \tag{9}$$

as shown in [4], function f_2 can be mapped into the circuit on the right of Fig. 3.

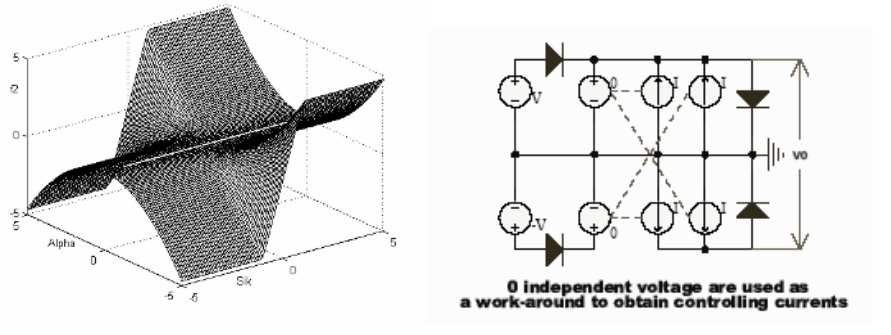


Fig. 3. Nonlinearity $f_2(s_{ik},\alpha_{ik})$ and equivalent circuit

3 Modeling of Static Circuit Systems

Both static and dynamic feedforward neural networks are employed to model the static circuit of simple amplifier in this section. Firstly, the simple amplifier circuit, as illustrated in Fig. 4, is simulated with the transient analysis of Pspice. The input data of V_{in} ($V(2,1)$) and the output data of V_0 ($V(4,0)$) are recorded. Secondly, the data pairs of V_{in} and V_0 are used as the target behavior in a training set. As shown in Fig. 4,

2-layer feedforward neural networks can be constructed to model the simple amplifier circuit, where the functions of the layer 1 are sigmoid, and the ones of the layer 2 are pure-linear.

As an example, V_{in} is the sine source with amplitude of 10mV and a frequency of 1MHz, the output signal V_0 can be obtained as a sine wave with amplitude of 322mV and a frequency of 1MHz. The model software then adapts the parameters of neural networks macro-model with 5000 iterations of Levenberg-Marquardt, until a good match is obtained, during which 401 equidistant time ranges from 0 to 2*us*.

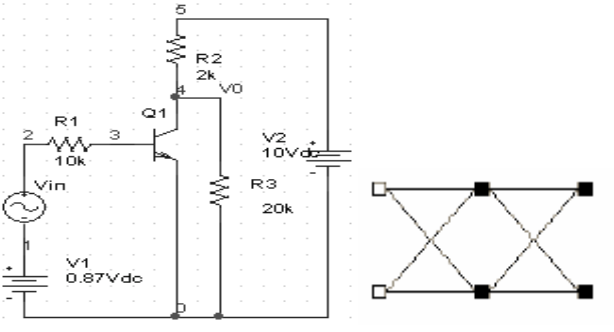


Fig. 4. Amplifier circuit and neural macro-model

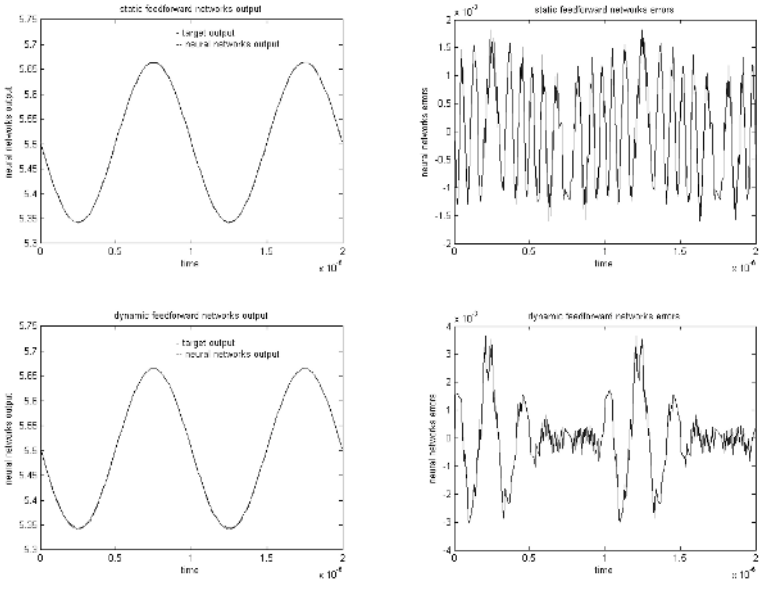


Fig. 5. Results from both static and dynamic feedforward neural networks

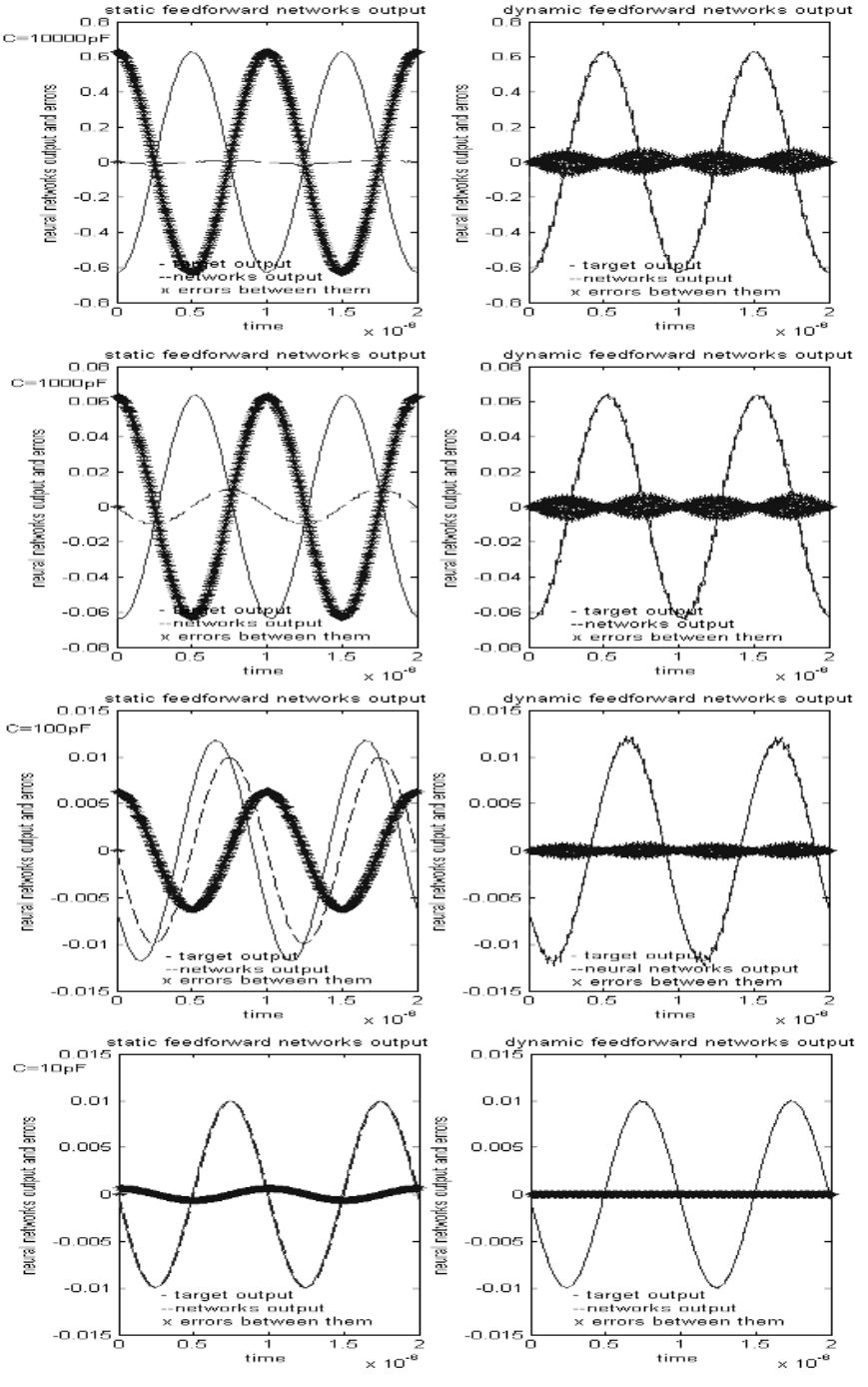


Fig. 6. Results for dynamic circuit system with C_1 from 10000pF to 10pF

The target output data V_0 and the trained outputs from both static and dynamic neural networks are, respectively, depicted in the two left pictures of Fig. 5. The two curves are not distinguishable in both two left pictures, because the differences between the target data and the outputs from both static and dynamic neural networks are too small, as illustrated in the two right pictures of Fig. 5. From the results, one can conclude that the dynamic feedforward neural networks can also accurately approximate to the behavior of the static circuit system, so do the static feedforward neural networks.

4 Modeling of Dynamic Circuit Systems

Both static and dynamic feedforward neural networks are employed to model the dynamic circuit of R-C, which is shown on the left of Fig. 1. Let the nonlinear source $U_1 = A\sin(2\pi ft)$, and one can obtain:

$$I(1,0) = I^{(dc)} + I^{(ac)} = \frac{1}{R_1}\cdot U_1 + C_1\frac{dU_1}{dt} = \frac{A}{R_1}\cdot\sin(2\pi ft) + 2\pi f\cdot A\cdot C_1\cdot\cos(2\pi ft) \qquad (10)$$

with $A = 10V, f = 1MHz$, $R_1 = 1K\Omega$. As discussed in the section 2.3, the 3-layer dynamic feedforward neural networks depicted on the right of Fig. 1 are employed, in which the values of w_{123}, v_{113}, v_{ij1} and v_{ij2} are zero, and the functions of layer 1 are sigmoid, whereas the functions of both layer 2 and layer 3 are pure-linear.

After simulation and modeling, the target output data *I(1,0)* and the trained outputs from both static and dynamic neural networks are, respectively, depicted in Fig. 6 for various values of C_1 from 10000pF to 10pF. In Fig. 6, the left column shows the results from static feedforward neural networks, whereas the right column is from the dynamic feedforward neural networks.

From Fig.6, one can observe that the dynamic feedforward neural networks give much more accurate results than the static counterparts. Only when $I^{(dc)} >> I^{(ac)}$ (e.g., C_1=10pF), the static feedforward neural networks can approximate the circuit with a reasonable accuracy, but give unacceptable errors for larger $I^{(ac)}$ (e.g., $C_1 \geq$100pF). On the other hand, the dynamic feedforward neural networks can very accurately represent this dynamic circuit system for all values of C_1 from 10000pF to 10pF.

5 Conclusion

In this work, a scheme to model integrated circuits employing the feedforward neural networks has been investigated. In order to represent some behavior of dynamic systems, a set of improved equations for dynamic feedforward neural networks has been utilized. Through analyzing some special choices for the neuron nonlinearity function f, the feedforward neural networks can themselves be represented by equivalent circuits, which indicates the probability of directly using neural models in existing analogue circuit simulators. Numerical examples for both static and dynamic electrical systems are included, the results have shown that the dynamic feedforward neural networks models overcome the pitfall of the static feedforward neural networks, and have the excellent performances on approximation to some behavior of both static and dynamic circuit systems.

Acknowledgements

This work is supported by the National Nature Science Foundation of China (60444004, 90307017).

References

1. Meijer, P.B.L.: Table Models for Device Modelling. Proc. Int. Symp. On Circuits and Syst. (1988) 2593-2596
2. Meijer, P.B.L.: Fast and Smooth Highly Nonlinear Table Models for Device Modeling. IEEE Trans. Circuits Syst. 37 (1990) 335-346
3. Hornik, K., Stinchcombe, M., White, H.: Multilayer Feedforward Networks. Neural Networks 2 (1989) 359-366
4. Meijer, P.B.L.: Neural Network Applications in Device and Subcircuit Modelling for Circuit Simulation. Ph.D. Dissertation, Proefschrift Technische Universiteit, Eindhoven (2003)
5. Kim, T., Adali, T.: Universal Approximation of Fully Complex Feedforward Neural Networks. IEEE International Conference on Acoustics, Speech, and Signal Processing (2002) 973-976
6. Stinchcombe, M., White, H.: Universal Approximation Using Feedforward Networks with Non-sigmoid Hidden Layer Activation Functions. International Joint Conference on Neural Network (1989) 613-617
7. Xu, J.J., Yagoub, M.C.E, Ding, R.T., Zhang, Q.J.: Feedforward Dynamic Neural Network Technique for Modeling and Design of Nonlinear Telecommunication Circuits and Systems. Proceedings of the International Joint Conference on Neural Networks (2003) 930-935
8. Ong, E., Flitman, A.: Using Neural Networks to Predict Binary Outcomes. Intelligent Processing Systems (1997) 427-431
9. Charalambous, C., Charitou, A., Kaourou, F.: Comparative Analysis of Artificial Neural Network Models: Application in Bankruptcy Prediction. International Joint Conference on Neural Networks (1999) 3888-3893
10. Suran, J.J.: Low–frequency Circuit Theory of the Double–base Diode. IEEE Transactions on Electron Devices 2 (1955) 44-48
11. Phillippe, P., El-Kamali, W., Pauker, V.: Physical Equivalent Circuit Model for Planar Schottky Varactor Diode. IEEE Transactions on Microwave Theory and Techniques 36 (1988) 250-255

Power Estimation of CMOS Circuits by Neural Network Macromodel

Wei Qiang, Yang Cao, Yuan-yuan Yan, and Xun Gao

School of Electronics Information, Wuhan University,
Luoyu Road 129, Wuhan 430079, China
wiley.qiang@gmail.com

Abstract. Neural network is employed to construct the power macromodel of complementary metal-oxide-semiconductor (CMOS) integrated circuits. In contrast to previous modeling approaches, it does not require empirically constructed specialized analytical equations for the power macromodel, and obtained statistics of a circuit's primary outputs simultaneously. It is suitable for power estimation in core-based systems-on-chips (SoCs) with pre-designed blocks. In experiments with the ISCAS-85 circuits, the average absolute relative error of the macromodel was below 5.0% for not only the average power dissipation, but also the maximum power dissipation.

1 Introduction

Power dissipation has become a critical concern in very large scale integrated (VLSI) circuits design with the advent of portable and high-density microelectronic device, especially for SoCs, which incorporates a diversity of functional blocks in a single chip. Excessive power dissipation may reduce the reliability and performance of VLSI chips. Hence, in order to synthesize circuits with high reliability, it is imperative to obtain a fast and precise estimation of the power dissipation of VLSI circuits at high abstraction levels in the early design stages [1].

Power macromodel is widely used in the high-level power estimation of CMOS integrated circuit design. It is useful when one is reusing a previously designed functional block. The basic idea underlying power macromodeling at the register transfer level (RTL) is to generate a mapping between the power dissipation of a functional block and certain statistics of its input/output signals [2], based on which the power dissipation could be directly obtained.

In this paper, neural network approach is employed to construct the power macromodel. In contrast to other proposed techniques, it could be used to perform RTL power analysis for CMOS circuit without the storage of a large look-up table (LUT) [3], [4], and does not require to empirically construct specialized analytical equations for the macromodel [5], [6]. It is more suitable for circuits with highly non-linear statistical properties than polynomial-based macromodels. In addition to average and maximum power dissipations, statistics of a circuit's primary outputs are also obtained, which makes it more suitable for the power estimation of core-based SoCs.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1313 – 1318, 2006.

2 Neural Network Power Macromodel

Power macromodels should be simple enough and applicable to the whole range of possible input signal statistics of any given circuits. It should include a fixed template, based on which the model construction process could be automatically performed [6]. Most existing abstract power macromodels are based on a template with a proper, limited set of parameters that are mainly responsible for the power dissipation of the circuits.

2.1 Macromodel Characterization

Key issues in designing a power macromodel include choosing a limited set of statistics, based on which the macromodel is constructed by a learning process with evenly distributed samples of power dissipation and statistics parameters. The efficiency of the power macromodel largely depends on the statistics of the input/output vector streams of the functional circuit block.

In this paper, average signal probability p, average switching activity d, average spatial correlation s, average temporal correlation t of the input/output stream are chosen as the input/output parameters to build the macromodel [3], [4], [5]. Consider a circuit block with M input/output ports and a binary input/output stream $\boldsymbol{S} = \{(x_{11}, x_{12}, \ldots, x_{1M}), \ldots, (x_{N1}, x_{N2}, \ldots, x_{NM})\}$, these parameters are defined as follows:

$$p = \frac{1}{MN} \sum_{k=1}^{N} \sum_{j=1}^{M} x_{j,k} \,, \tag{1}$$

$$d = \frac{1}{M(N-1)} \sum_{k=1}^{N-1} \sum_{j=1}^{M} x_{j,k} \oplus x_{j,k+1} \,, \tag{2}$$

$$s = \frac{2}{NM(M-1)} \sum_{k=1}^{N} \sum_{r=j+1}^{M} \sum_{j=1}^{M} x_{r,k} \oplus x_{j,k} \,, \tag{3}$$

$$t = \frac{1}{M(N-l)} \sum_{k=1}^{N-l} \sum_{j=1}^{M} x_{j,k} \wedge x_{j,k+l} \,, \tag{4}$$

where l is the time-shift. If $l = 1$, statistics t can be uniquely determined by the knowledge of p and d [4], therefore, like in [5], an empirical value of 10 is chosen for the time-shift in experiments. Some other constraints, that statistics of p, d and s should satisfy, were also proposed in [4].

2.2 Mapping Methods

In power macromodel techniques, the power estimation process is viewed as a practical vehicle for performing a nonlinear mapping between the power value and certain input/output statistics of a functional block, based on which the power dissipation could be directly obtained for any given input vector streams. Let m_{i} denote the number of the input parameters, and m_{o} the number of the output parameters, the neural

network includes a mapping $f : V_{m_i} \to V_{m_o}$ from m_i-dimensional Euclidean input space to a m_o-dimensional Euclidean output space.

Based on the statistics of average signal probability p_{in}, average switching activity d_{in}, average spatial correlation s_{in}, average temporal correlation t_{in} of the input vector stream, and the corresponding average power dissipation P_{avg} of the functional block, a simple macromodel for P_{avg} could be constructed based on the mapping:

$$f : L\{p_{in}, d_{in}, s_{in}, t_{in}\} \to L\{P_{avg}\}, \tag{5}$$

where $L\{p_{in}, d_{in}, s_{in}, t_{in}\}$ and $L\{P_{avg}\}$ are the respective Euclidean space.

Similar statistics of the output stream of the functional block, including p_{out}, d_{out}, s_{out}, and t_{out}, could be obtained simultaneously if the mapping of the macromodel is taken as:

$$f : L\{p_{in}, d_{in}, s_{in}, t_{in}\} \to L\{p_{out}, d_{out}, s_{out}, t_{out}, P_{avg}\}. \tag{6}$$

It is reasonable to consider that more information of the functional block would lead to higher accuracy of the mapping, however, it might become more difficult for a multi-output mapping to converge under certain learning process [7].

In order to improve the accuracy of the power macromodel, average switching activity of the output stream d_{out} were introduced as an input parameter of the mapping [4]. d_{out} is measured from a zero-delay simulation of the functional block, which add an indispensable step of the power estimation process. It is based on the idea that [4], during high-level (RTL) power estimation, one should perform an initial step of estimating the signal statistics at the visible RTL nodes from a high level functional simulation.

$$f : L\{p_{in}, d_{in}, s_{in}, d_{out}\} \to L\{P_{avg}\}, \tag{7}$$

In fact, there are quite a lot many stratified statistical sampling techniques to improve the efficiency of power estimation, which is based on a lower-cost zero-delay power estimate [8]. If the zero-delay power dissipation P_o is taken as one of the input parameters of the mapping, higher accuracy could be obtained for the power macromodel:

$$f : L\{p_{in}, d_{in}, s_{in}, t_{in}, P_o\} \to L\{P_{avg}\}. \tag{8}$$

Power macromodel techniques are not only useful for the estimation of average power dissipation P_{avg}, but also applicable to the prediction of other measures of power dissipation, including the maximum power dissipation P_{max}. Comparative power macromodel of P_{max} could be constructed based on the following mappings:

$$f : L\{p_{in}, d_{in}, s_{in}, t_{in}\} \to L\{p_{out}, d_{out}, s_{out}, t_{out}, P_{max}\}, \tag{9}$$

$$f : L\{p_{in}, d_{in}, s_{in}, t_{in}, P_o\} \to L\{P_{max}\}. \tag{10}$$

2.3 Macromodel Construction

Neural network, including back-propagation (BP) network and radial-basis function (RBF) network, are employed as the template to construct the power macromodel. For the sake of brevity, only RBF macromodel are presented in this paper. RBF network is a type of feedforward neural network that learns using a supervised training technique. It involves three layers with entirely different roles: the input layer, the hidden layer and the output layer. The design of a RBF neural network is equivalent to a curve-fitting problem in a high-dimensional space, and the learning process is to find a surface that provides a best fit to the training data. Radial basis functions are a special class of functions, whose response decrease exponentially with the distance from a center point. Theoretically, RBF network are able to approach any reasonable continuous function mapping with a satisfactory level of accuracy if enough number of hidden nodes are employed. However, too many hidden nodes may significantly increase the complexity of the RBF-NN macromodel, and accordingly, increase the time consuming as well as the memory cost. In consider that power macromodel is a "high-level" technique, which cannot yield results with very high accuracy due to the intrinsic "roughness" of the macromodel. Hence, in our experiments, the learning process is stopped when the average relative errors of the estimated power value are sufficient small (e.g. less than 10%). The average absolute relative error ε_{avg} are defined as:

$$\varepsilon_{\text{avg}} = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|P_{\text{macro}\,i} - P_{\text{sim}\,i}\right|}{P_{\text{sim}\,i}} \quad , \tag{11}$$

where N is the number of the input vector stream, $P_{\text{sim}\,i}$ and $P_{\text{macro}\,i}$ are the power value of the i stream by gate-level simulation and by power macromodel, respectively.

3 Accuracy Evaluation

Macromodels of the ISCAS-85 benchmark circuits were constructed based on formulas (6) and (9) in previous section. In order to study the accuracy of these macromodels, input vector streams of length 200 were randomly generated, which cover a wide range of signal statistics p, d, s and t. Reference values were obtained by simulating the circuits over 500 input vector streams at gate level, 300 of which were used to train the macromodel, and the rest 200 were used to verify the accuracy. In experiments, it is found that 20 hidden nodes are enough for the RBF macromodel to achieve very good accuracy.

In order to examine the results for all the circuits on the same plot, the power dissipations by gate-level simulated *vs* those by macromodel were plotted in normalized dynamic power unit: d. p. unit = (transitions × fanouts) / (clocks × gates). The combined scatter plot of all ISCAS-85 circuits showing the accuracy of macromodel for both the average power dissipation and maximum power dissipation are presented in

Fig. 1 (a) and (b), respectively. The solid line represents an exact matching between the macromodel values and the gate-level simulation data, whereas the dash lines correspond to the bound of a10% relative error. Both the average power dissipation value and the maximum power dissipation show good agreements with the gate-level simulation results, which prove the ability to learn and therefore generalize of the RBF network, and hence, the efficiency of the macromodel. Results with the same accuracy have also been obtained for the BP macromodel.

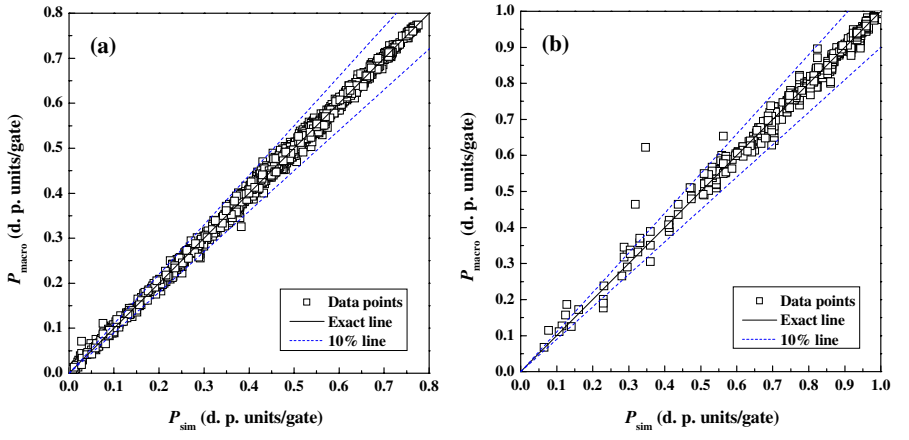


Fig. 1. Agreement between the power values from macromodel and the power values from gate-level simulation of (a) average power dissipation (b) maximum power dissipation

Table 1. Average absolute relative errors of power estimates and output statistics

Circuit	P_{max}	P_{avg}	p_{out}	d_{out}	s_{out}	t_{out}
c432	0.75%	3.15%	2.51%	4.42%	3.99%	5.18%
c499	0.69%	2.10%	0.77%	1.56%	1.05%	1.37%
c880	2.66%	1.90%	0.89%	3.48%	1.84%	1.41%
c1355	1.05%	1.43%	0.79%	1.59%	1.04%	1.36%
c1908	0.72%	2.05%	1.06%	2.49%	2.32%	2.41%
c2670	1.59%	1.70%	0.39%	1.02%	0.78%	0.70%
c3540	1.17%	2.25%	1.43%	3.16%	3.07%	2.79%
c5315	1.07%	1.20%	0.69%	2.22%	1.39%	1.30%
c6288	2.46%	1.47%	1.50%	1.73%	2.66%	2.96%
c7552	1.53%	2.16%	0.58%	1.72%	1.16%	1.12%

Table 1 shows the accuracy of the macromodel for the average and maximum power dissipation, as well as the signal statistics of the output streams. The average absolute relative errors are calculated according to equation (11), with most of the values less than 5%. It is worth noting that although some of the maximum powers by macromodel deviate apparently from those by gate-level simulation in Fig. 1 (b), the

average errors still remain low enough. It is due to the very small probability that the stream with a low P_{avg} to have a very high P_{max}.

It should be mentioned here that the efficiency of the neural network is significantly dependent on the training set. In power macromodeling the generation of input streams with different and diverse statistics significantly affect the accuracy. In our experiment, although randomly generated input vector streams could cover a wide range of signal statistics p, d, s and t, the input space is still far from being exhausted. Recently, advanced techniques for stream generation have been proposed [2], [4].

4 Conclusion

Neural networks are employed to construct power macromodel for high-level power estimation of CMOS circuits. Statistics of the input/output streams of the circuits, including average signal probability, average switching activity, and average spatiotemporal correlations were extracted as the input/output parameters of neural network. Over a wide range of signal statistics, the macromodel shows good accuracy for not only the average power dissipation, but also the maximum power dissipation. It is advantageous to use neural network to build the macromodel because it does not require specialized empirical analytical equations, and statistics of a circuit's primary outputs are obtained simultaneously, which makes it more suitable for core-based power estimation in SoCs.

References

1. Najm, F.N.: A Survey of Power Estimation Techniques in VLSI Circuits. IEEE Trans. VLSI Systems 2(4) (1994) 446-455
2. Liu, X., Papaefthymiou, M.C.: A Markov Chain Sequence Generator for Power Macromodeling. IEEE Trans. CAD of Integrated Circuits and Systems 23(7) (2004) 1048-1062
3. Barocci, M., Benini, L., Bogliolo, A., Ricco, B., Micheli, G.D.: Lookup Table Power Macro-Models for Behavioral Library Components. In: Puiri, V. (eds.): IEEE Alessandro Volta Memorial Workshop on Low-Power Design, IEEE, Piscataway (1999) 173-181
4. Gupta, S., Najm, F.N.: Power Modeling for High-Level Power Estimation. IEEE Trans. VLSI Systems 8(1) (2000) 18-29
5. Bernacchia, G., Papaefthymiou, M.C.: Analytical Model for High-Level Power Estimation. In: White, J., Sentovich, E.M. (eds.): IEEE/ACM International Conference on Computer-Aided Design, IEEE, Piscataway (1999) 280-283
6. Gupta, S., Najm, F.N.: Analytical Models for RTL Power Estimation of Combinational and Sequential Circuits. IEEE Trans. CAD of Integrated Circuits and Systems 19(7) (2000) 808-814
7. Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edn. Prentice-Hall, New York (1999)
8. Ding, C-S., Wu, Q., Hsieh, C-T., Pedram, M.: Stratified Random Sampling for Power Estimation, IEEE Trans. CAD of Integrated Circuits and Systems 17(6) (1998) 465-471

An Efficient Hardware Architecture for a Neural Network Activation Function Generator

Daniel Larkin, Andrew Kinane, Valentin Muresan, and Noel O'Connor

Centre for Digital Video Processing, Dublin City University, Dublin 9, Ireland
{larkind, kinanea, muresanv, oconnorn}@eeng.dcu.ie

Abstract. This paper proposes an efficient hardware architecture for a function generator suitable for an artificial neural network (ANN). A spline-based approximation function is designed that provides a good trade-off between accuracy and silicon area, whilst also being inherently scalable and adaptable for numerous activation functions. This has been achieved by using a minimax polynomial and through optimal placement of the approximating polynomials based on the results of a genetic algorithm. The approximation error of the proposed method compares favourably to all related research in this field. Efficient hardware multiplication circuitry is used in the implementation, which reduces the area overhead and increases the throughput.

1 Introduction

Artificial neural networks (ANN) have found widespread deployment in a broad spectrum of classification, perception, association and control applications [1]. However, practical ANN implementations for high dimensional data tasks, such as multimedia analysis, computer gaming, etc, create considerable demands for microprocessor cycles. This is due to the fact that such ANN require extremely high throughput, in addition to a large number of inputs, neurons and layers. The associated computational complexity is highly undesirable from a real time operation and low power consumption perspective. This poses considerable problems for constrained computing platforms (e.g. mobile devices) which suffer from limitations such as low computational power, low memory capacity, short battery life and strict miniaturisation requirements. An attractive solution to this is to design systems whereby ANN complexity can be addressed by offloading processing from the host processor to dedicated hardware for general purpose ANN acceleration.

There has been considerable research in both analog and digital hardware ANN implementations – [2][3][4]. Low complexity approaches have the benefit of allowing reduced silicon area, but this typically comes at the expense of output performance. It is generally accepted that, as the resolution and precision of the inputs, weights and the activation function are reduced, so too is the ability of the ANN to act as a universal approximator [5][6]. However, in an era where large microprocessors now use half a billion transistors, the overall benefit of

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1319–1327, 2006.

area savings in the order of tens to hundreds of transistors is questionable, if the output performance and scalability is compromised. This observation has motivated us to design a high precision, power efficient hardware activation function generator, which is capable of accommodating multiple activation functions and their derivatives.

The rest of this paper is organised as follows: Section 2 details related prior research in the area. Section 3 proposes the use of a minimax spline approximation scheme. Section 4 outlines an associated hardware architecture for this scheme. Section 5 details hardware synthesis results and power consumption estimates, whilst section 6 draws conclusions about the work presented.

2 Related Research

Given its desirable non linear characteristics and ease of differentiability, a sigmoid based activation function, such as that defined in eqn. 1, is commonly used in neural networks [7]. However, a direct hardware implementation is not practical as it requires excessive logic, resulting in significant power loss.

$$y(x) = \frac{1}{1 + e^{-x}} \tag{1}$$

Consequently, a number of approximations amenable to hardware implementation have been developed. Since a direct look up table (LUT) implementation uses excessive memory, approaches typically fall into the following broad categories: piecewise linear approximations [8][9][10][11] [12][13], piecewise second order approximations [11] and combinatorial input/output mappings [14]. Furthermore, there is considerable variance within each category. For example, an A-Law companding technique is used in [8], a sum of steps approximation is used in [9], a multiplier-less piecewise approximation is presented in [10] and a recursive piecewise multiplier-less approximation is presented in [13]. An elementary function generator capable of multiple activation functions using a first and second order polynomial approximation is detailed in [11]. Recently, a combinatorial approach has been suggested that considerably reduces the approximation error [14].

Our approach, which is described in detail in Section 3 & Section 4, uses a first order minimax polynomial approximation. The use of a minimax polynomial has been suggested before in the context of a floating point activation function approximation [12], however we further minimise the maximum error and implement an area and power efficient architecture. As will be seen from the benchmarks in Section 5, our approach produces the best approximation error, whilst being suitable for the implementation of multiple activation functions.

2.1 Data Representation and Precision Requirements

In function approximation there are two sources of error, the approximation method error and the data representation error resulting from the use of a finite

number of bits. To minimise area and power consumption, the minimum number of bits should be chosen, which will result in an acceptable error. Reduced precision Hardware ANN issues were explored in [5] [6]. It was found that 10 precision bits were sufficient for multi-layer perceptrons (MLP) trained via back propagation [6]. Using fewer precision bits than this will effect the convergence speed for on-chip learning, and in some cases may completely prevent convergence. The number of integer bits should be chosen based on the range of the activation function. The functions we have implemented (see Section 5) have a maximum usable input integer range of ± 8 and a maximum usable output range of ± 1. Outside of these bounds greater than 10 bits of precision would be required to represent the values. For these reasons we propose using a 4.10 fixed point representation (4 integers bits and 10 fractional bits) for inputs and a 2.10 fixed point output representation (2 integer bits and 10 fractional bits) for outputs. The data representation and precision used in related research is shown in Table 1.

Table 1. Data widths and precision used in related research

	Input				Output			
	Total Bits	**Range**	**Integer Bits**	**Fractional Bits**	**Total Bits**	**Range**	**Integer Bits**	**Fractional Bits**
Myers et al [8]	16	[-8,8)	4	12	8	[0,1)	1	7
Alippi et al [9]	not discussed							
Amin et al [10]	8	[-8,8)	4	4	8	[0,1)	1	7
Vassiliadis et al [11]	14	[-8,8)	3	10	14	[-8,8)	4	10
Faiedh et al [12]	Single precision floating point							
Basterretxea et al [13]	not discussed							
Tommiska-337 [14]	6	[-8,8)	3	3	7	[0,1)	0	7
Tommiska-336 [14]	6	[-8,8)	3	3	6	[0,1)	0	6
Tommiska-236 [14]	5	[-4,4)	2	3	6	[0,1)	0	6
Tommiska-235 [14]	5	[-4,4)	2	3	5	[0,1)	0	5

3 Proposed Approximation Scheme

Polynomial approximating functions, the best known of which is the Taylor's Series, can be used to represent any arbitrary continuous function. Whilst the error in a Taylor's Series is very small at the expansion point, it rapidly increases at the boundaries of the interval. Therefore when using a finite order polynomial, often a more evenly distributed error is preferable. Consequently other approximating polynomials, such as least squares, have been employed [11]. We propose using a minimax approximation instead [15]. A minimax polynomial exists for every approximation and has the characteristic of minimising the maximum error, by evenly distributing the error across the entire approximation range. To reduce the order of the approximating polynomial, the input domain of the function can be sub-divided into smaller intervals. This allows a polynomial of much lower order to approximate each of the of sub intervals. The resulting composite function is known as a piecewise polynomial or spline. Using a spline-based activation function approximation offers the benefit that multiple activation functions can

be accommodated with ease, by merely changing the coefficients of the approximating polynomial. This also means that minimal extra hardware is required to support the additional functions.

We use a Remez exchange algorithm in Matlab [16] to find the appropriate coefficients to generate minimax polynomials on discrete intervals for each activation function. This approach is novel in the context of fixed point ANN activation function approximation. As is detailed in Section 5, this greatly improves the approximating error relative to other polynomial approximating schemes such as least squares. We advocate employing a first order minimax approximation. This has the benefit of avoiding higher order x^n operations. The minimal error representable from 10 bits of precision (see Section 2) is reached when using only a small number of first order minimax approximating polynomials. To further reduce the number of polynomials required to achieve a specified accuracy, we employ the common approach of range reduction. Range reduction exploits inherent function redundancies such as function symmetry, periodic behaviour etc to allow fewer polynomial segments represent the function.

3.1 Optimisation of the Location of the Approximating Polynomials

The placement of the approximating polynomials on the input range clearly has a major bearing on the overall approximation error. The simplest approach is to evenly distribute the polynomials over the approximating range. However, astute placement can reduce the approximating error, although the potential search space is large. For example when using 5 polynomials in a 0 to 8 range with a precision of 10^{-3}, there are in the order of 10^{17} possible combinations for the location of the polynomials. Due to this large search space issue, we propose using a genetic algorithm (GA) to find the optimum location of the approximating polynomials. Unlike an exhaustive search, this solution is scalable even if the input range becomes extremely large, for example if using a double precision floating point representation.

The GA was implemented using the Genetic Algorithm Optimization Toolbox (GAOT) for Matlab [16]. The fitness function firstly uses the Remez exchange algorithm to generate the minimax polynomial coefficients for each candidate in the seed population. The Remez exchange algorithm works by solving a set of linear equations. Using the coefficients which are generated at each step, the minimax spline approximation for the chosen activation function (e.g. sigmoid) is constructed. The mean and maximum errors are then calculated from this approximation, using at least 10^6 samples. The GA explores the search space whilst attempting to minimise these mean and maximum error values.

As is the norm, the GA needed extensive tuning through trial and error exploration of the different input parameters. Initial population sizes of 10 to 1,000 were considered, along with extensive investigation into different crossover functions and different mutation functions. We achieved best results using arithmetic crossover and a multi-point non-uniform mutation with 5 mutation points. The GA typically improved the approximating error by 30% to 60% relative to an even distribution of the approximating polynomials.

4 Hardware Architecture

A simplified block diagram of the hardware datapath can be seen in figure 1. The range reduction and range reconstruction blocks currently have a trivial implementation, because only symmetry around the Y-axis is exploited. However, this is sufficient for a wide range of potential activation functions.

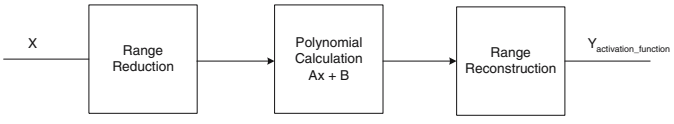


Fig. 1. Simplified datapath

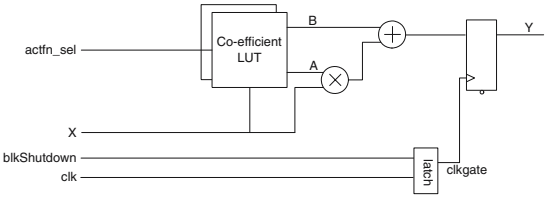


Fig. 2. Polynomial evaluation

Figure 2 shows a direct implementation of the linear polynomial $(AX + B)$ calculation. The value of the input, X, is used as an index into a LUT, which stores the values of the A and B coefficients. A single clock cycle array multiplier is used in conjunction with an adder to generate the approximation. However, this implementation has a number of drawbacks. Firstly, relative to the area of the full design, the hardware multiplier requires a considerable area overhead. Secondly, the multiplication is costly in terms of power consumption. This has motivated us to investigate alternative micro architectures for the polynomial evaluation. Consequently we propose using a radix-4 modified Booth algorithm to generate the partial products, this halves the number of partial products [17]. Efficient summation of the partial products is then achieved by using (7;2) column compressors [17]. The carry out of the (7;2) compressor does not depend on the carry in, therefore the critical path is improved and a higher clock frequency is supported. The resultant carry and save are added to the B coefficient using a (3:2) counter. A carry propagate adder is used to generate the final result. Employing these techniques provides the proposed architecture (see fig. 4) with a good trade off between speed, area and power consumption. The architecture is easily modified for additional pipeline stages should a higher throughput be required. Power reduction is further tackled at the gate level, by employing clock gating and using tristate buffers on the inputs. These two techniques help minimise the power when the block is not active.

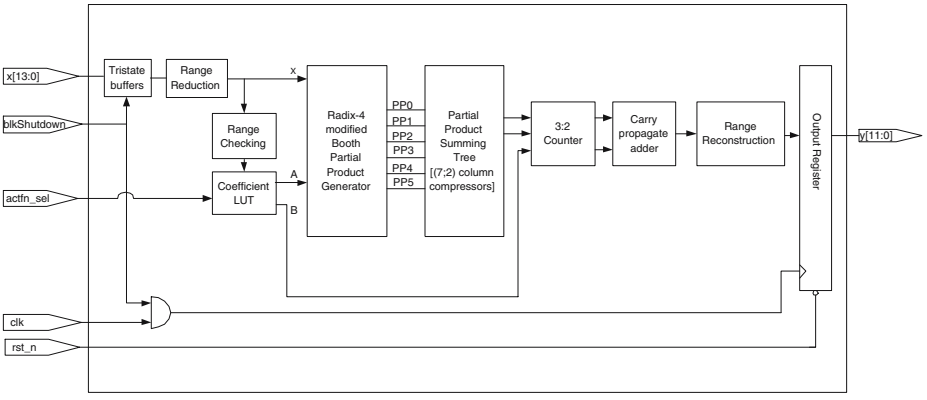


Fig. 3. Proposed hardware architecture

5 Results

We use the sigmoid activation function approximation to compare our approach to previous research. We tested our method using 2 to 8 approximating polynomial segments. The maximum and average error results for these tests can be seen in table 2. This table allows a fair comparison of method error between various approaches who share the same range and number of segments (relevant for piecewise approximations). In the cases where a direct comparison is possible, our method clearly outperforms related research. Tommiska [14] uses a direct input/ output combinatorial mapping, therefore the number of segments is not applicable. In this case it is fair to claim our method outperforms this approach on an ± 8 range when using 5 or more segments. The improvement

Table 2. Maximum and average Sigmoid Approximation errors

Design	Range	Segments	Maximum Error	Average Error
Myers et al [8]	[-8,8)	N/A	0.0490	0.0247
Alippi et al [9]	[-8,8)	N/A	0.0189	0.0087
Amin et al [10]	[-8,8)	3	0.0189	0.0059
Vassiliadis et al (First Order) [11]	[-4,4)	4	0.0180	0.0035
Vassiliadis et al (Second Order) [11]	[-4,4)	4	0.0180	0.0026
Faiedh et al [12]	[-5,5]	5	0.0050	n/a
Basterretxea et al (q=3) [13]	[-8,8)	N/A	0.0222	0.0077
Tommiska (337) [14]	[-8,8)	N/A	0.0039	0.0017
Tommiska (336) [14]	[-8,8)	N/A	0.0077	0.0033
Tommiska (236) [14]	[-4,4)	N/A	0.0077	0.0040
Tommiska (235) [14]	[-4,4)	N/A	0.0151	0.0069
Proposed approach	[-8,8)	2	0.0158	0.0068
Proposed approach	[-8,8)	3	0.0078	0.0038
Proposed approach	[-8,8)	4	0.0047	0.0024
Proposed approach	[-8,8)	5	0.0032	0.0017
Proposed approach	[-8,8)	6	0.0023	0.0012
Proposed approach	[-8,8)	7	0.0017	0.0009
Proposed approach	[-8,8)	8	0.0013	0.0009

Table 3. Max and Mean Errors approximations for other functions

Design	Sigmoid Derivative			Tanh		
	Range	Maximum Error	Average Error	Range	Maximum Error	Average Error
Vassiliadis et al (1^{st} Order)	[0,8)	8.8×10^{-3}	2.6×10^{-3}	[0,8)	5.7×10^{-2}	5.0×10^{-3}
Vassiliadis et al (2^{nd} Order)	[0,8)	4.6×10^{-3}	5.0×10^{-4}	[0,8)	1.6×10^{-2}	1.6×10^{-3}
Our approach (4 segments)	[0,8)	3.1×10^{-3}	1.7×10^{-3}	[0,8)	9.5×10^{-3}	2.4×10^{-3}

achieved by leveraging the effective search capabilities of genetic programming is apparent from comparing our five segment approach with the five segment approach of [12], who employs a minimax polynomial with a floating point data representation. The GA is predominately responsible for the 36% improvement in error. Table 3 shows maximum and average approximation errors for additional functions, including Tanh and the sigmoid derivative. Tanh is another commonly useful activation function, whilst the sigmoid derivate is important if on-chip training (using back propagation) is used. Our approach gives a better maximum error than both the first and second order approximation of [11].

5.1 Hardware Implementation Results

Two architectures were implemented, a conventional array multiplier approach and the efficient multiplier architecture, which was described in Section 4. Both architectures complete processing within one clock cycle. Similar to other single cycle activation function hardware architectures, this is considerably faster than a software implementation. For example we profiled a direct implementation (ANSI C) of Eqn. 1 on an ARM 920T processor (commonly found on mobile devices), a single evaluation requires 395 clock cycles to complete. Consequently there is considerable throughput and power reduction gains from using a hardware implementation.

The proposed architecture was captured in Verilog HDL and synthesised using Synopsys Design Compiler and Synplicity Synplify Pro for a 90nm TSMC ASIC library and a Xilinx Virtex 2 FPGA respectively. Dynamic power consumption estimates for the ASIC library (1.2V source) were generated using Synopsys Prime Power, whilst Xilinx X-Power was used for the FPGA. A summary of the synthesis results for an eight segment implementation can be seen in table 4. Using the ASIC library the area and power figures are comparable for both designs at 100 MHz. However as the frequency increases, the differences become

Table 4. Hardware synthesis and power consumption results

Design	Frequency [MHz]	Area [Gates]	Average Power Milliwatts
ASIC - Array multiplier	100	1,812	0.1101
ASIC - Array multiplier	270	2,907	0.2810
ASIC - Proposed approach	100	1,783	0.1379
ASIC - Proposed approach	270	2,272	0.4167
FPGA - Array multiplier	40	5,383	8.000
FPGA - Proposed approach	40	2,904	9.000

more pronounced. The maximum clock frequency of the array multiplier approach is 270MHz, whilst the corresponding figure for the proposed approach is 330MHz. The proposed approach uses almost 22% less area at 270MHz than the array multiplier. The difference in area between the architectures on the FPGA is even more pronounced, as can be seen from table 4. As would be expected, the maximum clock frequency for both designs is considerably lower when using the FPGA, fortunately this can be easily improved by adding additional pipeline stages.

6 Future Work and Conclusions

This paper has presented a high precision, scalable hardware architecture for an activation function generator. It represents preliminary work toward a complete power efficient hardware ANN accelerator suitable for high dimensional data sets. An extension to the number of activation functions supported is planned, along with modifications to the GA to take account of representation issues caused by fixed width word length. In addition we also intend to conduct a thorough investigation on the suitability of using floating point representation.

Acknowledgement

The support of the Commercialisation Fund of Enterprise Ireland is gratefully acknowledged.

References

1. Widrow, B., Rumelhart, D.E., Lehr, M.A.: Neural Networks: Applications in Industry, Business and Science. Communications of the ACM **37** (1994) 93–105
2. Cauwenberghs, G., Bayoumi, M.: Learning on Silicon - Adaptive VLSI Neural Systems. Kluwer Academic (1999)
3. Zhang, D., Pal, S.K.: Neural Networks and Systolic Array Design. World Scientific, New Jersey (2002)
4. Ruckert, U.: ULSI Architectures for Artificial Neural Aetworks. IEEE Micro **22** (2002) 10–19
5. Draghici, S.: On the Capabilities of Neural Networks using Limited Precision Weights. Neural Networks **15** (2002) 395 – 414
6. Holt, J., Hwang, J.: Finite Precision Error Analysis of Neural Network Hardware Implementations. IEEE Transactions on Computers **42** (1993) 280 – 291
7. Haykins, S.: Neural Networks - A Comprehensive Foundation. Prentice-Hall (1999)
8. Myers, D.J. and Hutchinson, R.A: Efficient Implementation of Piecewise Linear Activation Function for Digital VLSI Neural Networks. Electronics Letters **25** (1989) 1662–1663
9. Alippi, C., Storti-Gajani, G.: Simple Approximation of Sigmoid Functions: Realistic Design of Digital VLSI Neural Networks. In: Proceedings of the IEEE Int'l Symp. Circuits and Systems. (1991) 1505–1508

10. Amin, H., Curtis, K., Hayes Gill, B.: Piecewise Linear Approximation applied to Non-Linear Function of a Neural Network. IEE Proceedings Circuits, Devices and Systems **144** (1997) 313 – 317
11. Vassiliadis, S., Zhang, M., Delgado-Frias, J.: Elementary Function Generators for Neural-Network Emulators. IEEE Transactions on Neural Networks **11** (2000) 1438 – 1449
12. Faiedh, H., Gafsi, Z., Besbes, K.: Digital Hardware Implementation of Sigmoid Function and its Derivative for Artificial Neural Networks. In: Proceedings. of the 13th International Conference on Microelectronics, Rabat, Morocco (2001) 189 – 192
13. Basterretxea, K., Tarela, J., del Campo, I.: Approximation of Sigmoid Function and the Derivative for Hardware Implementation of Artificial Neurons. IEE Proceedings of Circuits, Devices and Systems **151** (2004) 18–24
14. Tommiska, M.: Efficient Digital Implementation of the Sigmoid Function for Reprogrammable Logic. IEE Proceedings Computers and Digital Techniques **150** (2003) 403 – 411
15. Pineiro, J., Oberman, S., Muller, J., Bruguera, J.: High-Speed Function Approximation Using a Minimax Quadratic Interpolator. IEEE Transactions on Computers **54** (2005) 304 – 318
16. The MathWorks: MATLAB and Simulink for Technical Computing (2006) http://www.mathworks.com/.
17. Koren, I.: Computer Arithmetic Algorithms. 2nd edn. A K Peters Ltd (2001)

A Design and Implementation of Reconfigurable Architecture for Neural Networks Based on Systolic Arrays

Qin Wang, Ang Li, Zhancai Li, and Yong Wan

Information Engineering School, University of Science and Technology Beijing, China
wangqin@ies.ustb.edu.cn

Abstract. This paper proposes a reconfigurable architecture for VLSI implementation of BP neural networks with on-chip learning. Basing on systolic arrays, this architecture can flexibly adapt to neural networks with different scales, transfer functions or learning algorithms by reconfiguration of basic processing components,. Three kinds of reconfigurable processing units (RPU) are proposed firstly basing on the analysis of neural network's reconfiguration. Secondly, the paper proposes a reconfigurable systolic architecture and the method of mapping BP networks into this architecture. The implementation of an instance on FPGA is introduced in the last. The results show that this flexible architecture can also achieve a high learning speed of 432M CUPS (Connection Updated Per Second) at 100MHz using 22 multipliers.

1 Introduction

Artificial neural networks (ANN) are widely used in intelligence control and pattern recognition. However, the traditional software implementation basing on general-purpose processor have two problems: one is lacking of parallelism, the computing speed could not satisfy the real time requirement; the other is that many embedded applications have special restriction on stability, power and size of ANN system. For these reasons, researchers have proposed kinds of hardware implementation methods, including FPGA (Field Programmable Gate Array) implementations [1], neuro-chip [2][5][6][8] and DSP accelerator board [7], etc.

However, neural hardware has the shortcoming of lacking flexibility. For different applications (such as different control objects), neural networks vary in scale, parameter, topology and training methods. But hardware can hardly adapt to these changes. Relative works, such as general purpose neuro-chip [2] and general purpose neuro-computer [3], have given some ideas to improve the flexibility of neural hardware. But the high cost of these methods became another block for their using in practical application. Moreover, time-to-market is another sensitive requirement for engineering applications.

In order to seek a proper solution, this paper proposes a reconfigurable architecture based on systolic array. The main idea is: neural algorithm are decomposed into several basic computations that are executed by reconfigurable processing units (RPU),

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1328 – 1333, 2006.

which are designed carefully as IP cores and saved in core library; all the RPUs are interconnected in regular arrays, so when neural networks changed, new hardware can be reconfigured by increase (or decrease) the amount of RPUs or replace RPUs with different functions; reusable IP (Intellectual Properties) cores and FGPA devices provide the fast prototyping ability.

This architecture is researched with an example of the most widely used BP networks. The reconfiguration of neural networks is analyzed and three kinds of RPUs are proposed in section 2. Section 3 introduces the reconfigurable systolic architecture for neural processor and the hardware-mapping algorithm. Section 4 gives out an implementation example and its' results. And in the last section, the directions of future research work are concluded.

2 Reconfiguration Analyzation of BP Networks

From the structure and mathematical model of neural networks, we can conclude that neural networks can be represented as a four elements tuple [V, E, f, s], where 'V' is the set of neurons, 'E' is the set of connections between different neurons, 'f' stands for transfer function, and 's' denotes learning algorithm. So, the reconfiguration of neural networks can be divided as three aspects: structure reconfiguration, transfer function reconfiguration, and learning algorithm reconfiguration.

To explicitly explain those reconfiguration features, BP network is used as an example for analyzing. Considering from the view of computing, the execution of BP algorithm can be divided as three stages known as feed-forward, error back-propagation and weight updating. The mathematical descriptions are as follows:

$$o_i^k = f\left(\sum_{j=1}^{N_{k-1}} w_{ij}^k o_j^{k-1}\right) \tag{1}$$

$$\delta_i^k = f'(\cdot)\sum_{j=1}^{N_{k+1}} \delta_j^{k+1} w_{ji}^{k+1} \tag{2}$$

$$w_{ij}^k(t+1) = w_{ij}^k(t) + \alpha\delta_i^{k+1} o_j^k \tag{3}$$

Where o_i^k represents the i-th neuron's output in the k-th layer; w_{ij}^k represents weight goes from the j-th neuron in layer k-1 to the i-th neuron in layer k; N_k is the number of neurons in layer N; α is learning rate. Threshold value can be treated as a special weight; it connects a clamped input 1, so (1) is simplified to multiply-accumulation. A virtual layer is defined behind the output layer, which contains two nodes that have $\delta_1=1$, $\delta_2=-1$ and $w_{1q}=y_q$, $w_{2q}=o_q$ respectively. By this means, the δ of both hidden and output layers can be computed in (2).

Components for different operations can be merged into three kinds of RPUs: MAC for multiply-accumulation; F for non-liner function mapping; WU for weight update.

3 Reconfigurable Systolic Architecture and Mapping Algorithm

3.1 Architecture Design

Neural network is a massively parallel computing system. However, parallel computing in hardware faces bottleneck of data communication, which is how to transfer operation data (generally from a host) to multiple parallel-computing units, and how to send the results to memories simultaneously. Besides, we can know from the algorithm of BP networks that multiply of weight matrix and input vector (multiply-accumulation in scalar) is the main computation, so how to implement this matrix-vector multiply efficiently is the key of neural hardware.

Systolic array [4] is a processing unit's network that could transfer and compute operation data rhythmically. This architecture can settle the data bottleneck problem effectively and can implement matrix-vector multiply using pipeline. Besides, the modularity and expandability of systolic array are two other features suitable for our reconfigurable approach.

Connecting the RPUs proposed in former section in arrays, we could get the reconfigurable systolic architecture shown in Fig. 1(d). In this architecture, MAC array do the multiply-accumulation job in feed-forward and back-propagation stage, WU array compute new weight values, between these two arrays are WM, TM is memories for temporary results, all operations are controlled by finite state machine FSM. Because of the regularity of arrays, when neural networks changed in scale, new hardware can be reconfigured by increasing (or decreasing) the amount of RPUs or modifying the multiplex methods in FSM control. Besides, we can replace F and WU modules with different functions so as to adapt to transfer function and learning rules changes.

3.2 Mapping Algorithm

Mapping algorithm refers to how the data and computing processes of BP networks can be mapped into the components and data path of the architecture.

In this architecture, we use neuron parallelism as the basic computing mode for RPU, which requires one multiplier per neuron, all the neurons in the same layer work in parallel, it has lower degree parallelism but also need less hardware resources.

Assume N=Max(n(1),n(2)……,n(M)), which n(i) is the neuron numbers of the i-th layer. Then the number of MAC nMAC≤N, all computations of the k-th neuron in some layer are arranged to the (k mod nMAC)-th MAC. Because the multiply-accumulation results come out from systolic array in sequence, only one F module is needed (nF=1). This is an important feature of systolic array. Great hardware resources are saved in the implementation of nonlinear function no matter in lookup table method or in approximation method.If nWU (the number of WU) > nMAC, RAW (read after write) error will occur in weight update phase. If nWU < nMAC, two WM read operations and one WM write operation may happen at one same time. This conflict can be avoided if nWU=nMAC.

The biggest amount of operands in neural networks is weight. In order to save storage space and operation time, this design uses a unified memory structure for computing in three phases of BP algorithm—feed-forward, back-propagation and weight updating. However, we can see from (1), (2) and (3) that weight matrix used in back-propagation is the transpose of its counterpart in feed-forward and weight updating stages. So the key issue is how to implement three pipeline processes basing on one storage structure.

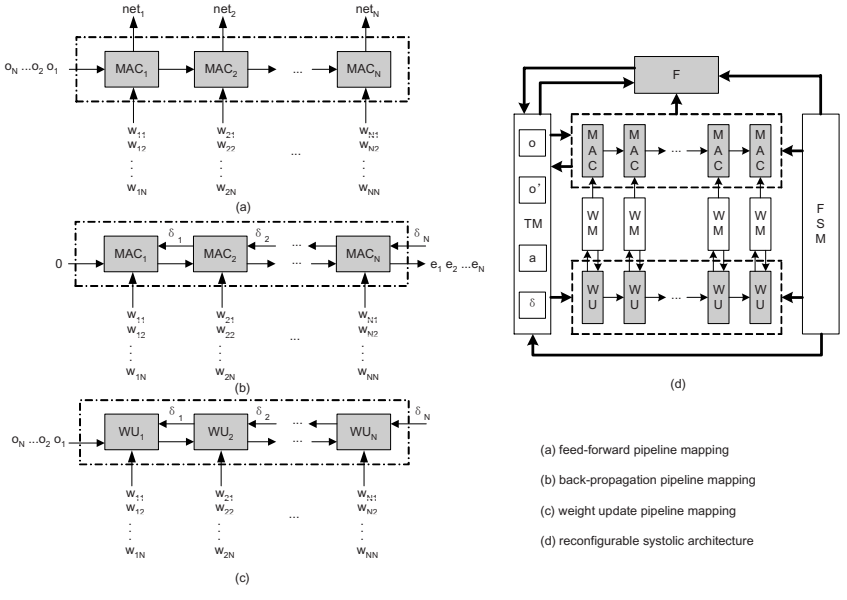


Fig. 1. Reconfigurable systolic architecture and mapping algorithm

Rule 1: (feed-forward pipeline, MAC array working at the k^{th} layer, Fig.1(a))

1. The outputs $o_1o_2\ldots o_N$ of the (k-1)-th layer enter the MAC array in sequence, and be transferred in pipeline.
2. In the i-th tap, o_1 arrive at MAC_i as the first operand, and $w_{i1}w_{i2}...w_{iN}$ start to enter MAC_i in sequence as the second operand.
3. o_j multiply w_{ij} in MAC_i and the product is added to accumulation result in last tap; starting from the Nth tap, the neuron accumulation result net_i goes out from MAC_i in sequence.

Rule 2: (back-propagation pipeline, MAC array working at the k^{th} layer, Fig.1(b))

1. Initial value 0 enters MAC_1's accumulation pipeline input in each tap.
2. δ_i of the (k+1)-th layer is pumped back to MAC_i and holds on as the first operand. In the i-th tap, $w_{i1}w_{i2}...w_{iN}$ start to enter MAC_i in sequence as the second operand.

3. δ_i multiply w_{ij} in MAC_i and the product is added to accumulation result come from MAC_{i-1}, the sum is transferred to MAC_{i+1} through accumulation pipeline input; starting from the N^{th} tap, the accumulation result e_i goes out from MAC_N in sequence.

Rule 3: (weight update pipeline, WU array working at the k^{th} layer, Fig.1(c))

1. The outputs $o_1o_2\ldots o_N$ of the (k-1)-th layer enter the WU array in sequence, and be transferred in pipeline.
2. δ_i of the (k+1)-th layer is pumped into WU_i and holds on as the first operand. In the i-th tap, o_1 arrive at WU_i as the second operand, and $w_{i1}w_{i2}...w_{iN}$ start to enter WU_i in sequence as the third operand.
3. δ_i multiply o_j and α in WU_i and the product is added to w_{ij}, the sum is new weight value w_{ij}, and it is write back to the weight memory.

4 Implementation Results

We implement the proposed architecture with a example of 25×10×10 three layers feed-forward network, which has sigmoid transfer functions and uses basic BP algorithm as learning rule.

16-bit fix-point data is used for data representation. Because the computing rules for feed-forward and back-propagation pipeline are different, the data path of MAC module is designed having the ability of dynamic reconfiguration, which is controlled by a "forward/backward" signal coming from FSM. According to the distribution of sigmoid function, two compressed look-up-tables are used to implement transfer function and its' derivative in F module.

We explored the scale reconfiguration by using different number RPUs to implement the same neural network. Using Xilinx virtex2 pro FPGA as target device, the synthesis results are shown in table 1 :

If nMAC=10, it takes 81 cycles to complete the iteration of one training vector. At the frequency of 100MHz, the learning speed (weight updated per second) is 432M CUPS, which is about 60 times speedup than software implementation on PC (Pentium4 processor, Windows 2000 operation system). For different learning algorithm, we replace the WU module with new MWU module, which uses momentum back-propagation algorithm to update weights. It has a lower Max Freq. in the synthesis results, but it also takes 81 cycles to complete the iteration of one training vector.

Explicit comparisons between various typical implementations are listed in table 2. The approach of ours achieved the greatest performance also with flexible advantages.

Table 1. Synthesis results with different number RPUs

xcv2p4-6	nMAC=10	nMAC=5
Slices	1600	1056
Slice Flip Flops	1320	840
4 input LUTs	2820	1904
BRAM(18Kb)	17	20
MULT(18×18bit)	22	12
Max. Freq. (MHz)	104.150	104.150

Table 2. Comparisons of various implementations

Name	Chip Type	Size (Processing Elements, PE)	Precision (bits)	Performance (CUPS)	Reconfiguration Ability*
CNAPS[5]	Neuro-chip: N6400	64 (per chip)	16(fix point)	256M	S
SYNAPSE[6]	Neuro-chip: MA16	4 (per chip)	16(fix point)	4.25M	S
RAP[7]	DSP : TM320C30	4 (per board)	32(floating point)	13.2M	S, T
RSA[8]	Neuro-chip: RSAP-16	16 (per chip)	16(fix point)	19.9M	S, L
PC	CPU: Pentium4 2.0G	1 (per chip)	32(floating point)	7.78M	S, T, L
Ours	FPGA: XC2VP20	20 (per chip)	16(fix point)	432M	S, T, L

*S stands for scale reconfigurable, T stands for transfer function reconfigurable, L stands for learning algorithm reconfigurable.

5 Conclusion

The paper discussed the reconfigurable systolic architecture based on BP networks. It should be noticed that this reconfigurable architecture not only fit for BP networks, but also can be generalized to fit for more kinds of ANN. Reconfigurable devices FPGA and advanced EDA development methods (such as IP reuse) make the implementation of this architecture more easy and fast. Our focus and directions of future research are to build up a reconfigurable platform basing on this architecture. Through the two key technologies of RPU library construction and mapping algorithms, the platform can automatically generate synthesable hardware description codes for neural networks. By using this platform, neural hardware design can be improved from RTL level to algorithm level, which will push forward the application of neural hardware in relative fields.

References

1. Ferreira, P., Ribeiro, P., Antunes, A., et al.: Artificial Neural Networks Processor-A Hardware Implementation Using A FPGA. In: The 4th International Conference on Field-Programmable Logic and Applications (2004) 1084-1086
2. Lu, C., Shi, B.X., Lu, C.: A General-purpose Neural Network with On-chip BP Learning. In: IEEE International Symposium on Circuits and Systems (2002) 520-523
3. Wang S.J., Li Z.Z.: Discussion on the Basic Mathematical Models of Neurons in General Purpose Neurocomputer. Acta Electronica Sinica 29(5) (2001) 577-580
4. Kung, S.Y., Hwang, J.N.: Parallel Architectures for Artificial Neural nets. In: IEEE International Conference on Neural Networks (1988) 165-172
5. Hammerstrom, D.: A VLSI Architecture for High Performance, Low-cost, On-chip Learning. In: International Joint Conference on Neural Networks (1990) 573-574
6. Ramacher, U., Raab, W., Anlauf, J., et al.: Multiprocessor and Memory Architecture of the Neurocomputers SYNAPE-1. In: The 3rd International Conference on Microelectronics for Neural Networks (1993) 227-231
7. Morgan, N., Beck, J., Kohn, P., et al.: The Ring Array Processor: A Multiprocessing Peripheral for Connectionist Applications. Journal of Parallel and Distributed Computing 14(3) (1992) 248-259
8. Murtagh, P.J., Tsoi, A.C.: A Reconfigurable Bit-serial VLSI Systolic Array Neuro-chip. Journal of Parallel and Distributed Computing 44(1) (1997) 53-70

Hardware In-the-Loop Training of Analogue Neural Network Chip

Liang Zhang and Joaquin Sitte

School of Software Engineering and Data Communications,
Queensland University of Technology, Brisbane, Australia
{lj.zhang, j.sitte}@qut.edu.au

Abstract. In this paper we present the results of neural network hardware in-the-loop training for an analogue Local Cluster Neural Network (*LCNN*) chip. We use a Probabilistic Random Weight Change (*PRWC*) algorithm that is a combination of the random weight change and simulated annealing algorithms. We applied the *PRWC* algorithm to in-the-loop training for multi-dimensional function approximations and for predictions. We discuss the training strategy and the experiment results.

1 Introduction

The *Local Cluster neural net (LCNN)* proposed by Geva and Sitte [1] is a special kind of multilayer perceptron where clusters of sigmoidal neurons generate localized functions. The *LCNN* has been realized in the *LCX* analog VLSI implementation [2] [3]. Characterization of the *LCX* chip had showed that it closely reproduces the mathematical function of the *LCNN* [4]. In this paper we investigate the training of *LCX* chip for multivariate function approximation.

During the past decade, different methods for training analog neural nets have been proposed and applied. Most of this research focused on *model-free* techniques for training analog neural nets. The techniques were tested in simulations or in neural net hardware for simple problems. Random Weight Change was used for training a neural net chip [5]; Weight Perturbation was used for XOR function in an analog circuit [6]; Back Propagation, Weight Perturbation and Simulated Annealing were used in neural net training in simulations [7] [8]. However there are no reports on training an analog VLSI neural net for function approximation. We refer to these techniques and formulate the Probabilistic Random Weight Change algorithm for training the *LCX* analog chip function approximation. This paper briefly describes the *LCNN* and the *LCX* chip, explains the training procedure and specific training algorithm, presents the results for various training tasks and discusses the training convergence and accuracy.

2 Local Cluster Neural Network (*LCNN*) and Its Analog Hardware Implementation

Fig. 1 shows the signal flow diagram for a segment of two clusters of a Local Cluster Neural Network (*LCNN*). The *LCNN* uses sigmoidal neurones in two

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1334–1339, 2006.

hidden layers to form functions localised in input space, which is similar to Radial Basis Functions (*RBF*) but capable of representing a wider range of localised function shapes [3]. The *LCNN* output is a linear combination of localised scalar functions in n-dimensional input space:

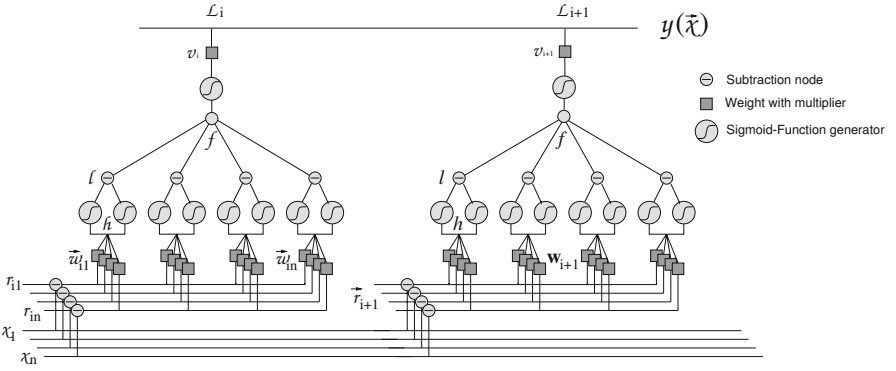


Fig. 1. Network structure of LC net

$$y(\boldsymbol{x}) = \sum_{\mu=1}^{m} v_\mu L_\mu(\mathbf{W}_\mu, \boldsymbol{r}_\mu, k, \boldsymbol{x}) \tag{1}$$

where v_μ is the output weight, $\mathbf{W}_\mu$ is the weight matrix that determines the local function shape, and the weight vector $\boldsymbol{r}_\mu$ determines the position of the localised function and k is the sigmoid slope.

The *LCX* chip has eight equal clusters, six inputs, one output and 8-bit digital weight storages. Each cluster receives the same inputs and has the structure as show in Fig. 2.

3 Training the *LCX* chip

The *LCX* chip does not have on-chip weight adaptation circuitry. Weight values have to be down-loaded on the chip from an external source. This suggests off-chip computation of the weights. However, this method does not work because of the fluctuations and deviations of the fabricated analogue circuits from their design specifications[4]. The weights obtained by training a software simulated *LCNN* will produce a different function on the chip due to unknown fluctuations. The solution to this problem is hardware-in-the-loop training where the weight update is calculated using the output from the hardware neural net instead of the simulated neural net from the computer. Fig. 3 shows a block diagram for the hardware-in-the-loop training of the *LCX* chip. The computer sends input and weights to the chip, reads the output from the chip, adjusts weights, and sends them to the chip for next training iteration.

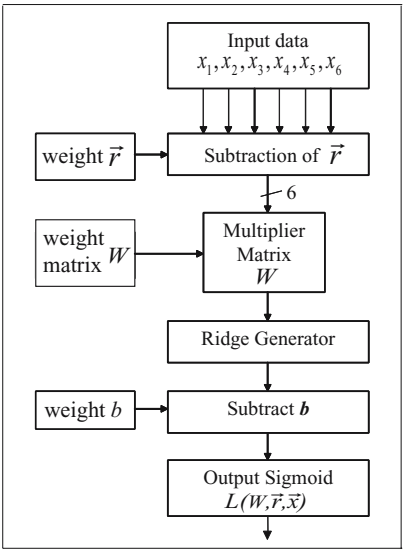


Fig. 2. The structure of LC clusters

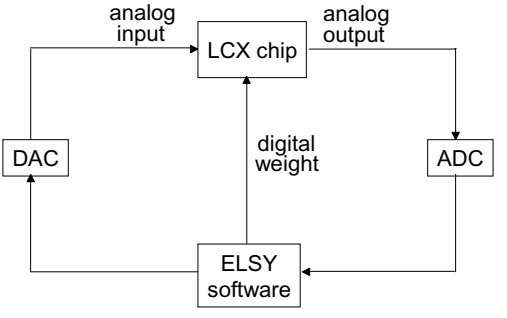


Fig. 3. LCX chip in-the-loop training

4 Training Algorithms

Training of a Neural Network consists in finding the weights that minimise the error of the output compared to a desired response. We used the mean square deviation as the error function:

$$E = \frac{1}{p}\sum_{i=1}^{p}(y_i - y_i^*)^2 = \frac{1}{p}\sum_{i=1}^{p} e_i^2 \tag{2}$$

where p is the number of training samples, y is the neural network output and y^* is desired output.

4.1 Gradient Descent Training Algorithm

In the *LCNN* simulation we obtained good results with a modified gradient descent (*GD*) method, that uses individual adaptive learning rates. The weight update equations are (3-5). For a detailed description of this algorithm see [1].

$$\Delta w_{\mu ij} = \eta_w \sum_p e_p \frac{\partial y(\boldsymbol{x_p})}{\partial w_{\mu ij}} = \eta_w \sum_p e_p v_\mu \sigma_o^{'}(f_\mu(\boldsymbol{w}, \boldsymbol{r}, k_1, \boldsymbol{x_p}) - b)\frac{\partial f_\mu(\boldsymbol{x_p})}{\partial w_{\mu ij}} \tag{3}$$

$$\Delta r_{\mu j} = \eta_r \sum_p e_p \frac{\partial y(\boldsymbol{x_p})}{\partial r_{\mu j}} = \eta_r \sum_p e_p v_\mu \sigma_o^{'}(f_\mu(\boldsymbol{w}, \boldsymbol{r}, k_1, \boldsymbol{x_p}) - b)\sum_{i=1}^{n}\frac{\partial l_{\mu i}(\boldsymbol{x_p})}{\partial w_{\mu ij}} \tag{4}$$

$$\Delta v_\mu = \eta_v \sum_p e_p \frac{\partial y(\boldsymbol{x_p})}{\partial v_\mu} = \eta_v \sum_p e_p L_\mu(\boldsymbol{w}, \boldsymbol{r}, k_1, k_2, \boldsymbol{x_p}) \tag{5}$$

This method can not be used for LCX chip-in-the-loop training because intermediate values are required as in equations (3) – (5) and these are not accessible on the LCX chip. An alternative training algorithms has to be found.

The Random Weight Change (RWC) [5] and the Simulated Annealing (SA)[9] are potential candidates. These two algorithms have the advantages of being *model-free* and no intermediate outputs are needed in the calculation.

4.2 Random Weight Change (RWC) and Simulated Annealing (SA)

The Random Weight Change (RWC) algorithm [5] is defined by equations (6), (7). If the new weights decrease the error E, the same weight *changes* are iterated. When the error E increases, the weight changes are reset randomly.

$$w_i(n+1) = w_i(n) + \Delta w_i(n+1) \tag{6}$$

$$\Delta w_i(n+1) = \begin{cases} \Delta w_i(n) & \Delta E < 0 \\ \delta \times Rand(n) & \Delta E \geq 0 \end{cases} \tag{7}$$

The Simulated Annealing (SA) algorithm [9] employs a random search that not only accepts weight changes that decrease error, but also some changes that increase it. In each training step, weights are updated by small random values. If the error decreases, the updated weights are accepted. The case of error increasing is treated probabilistically: the probability that the updated weight is accepted is $p(\Delta E) = e^{-\frac{\Delta E}{T}}$.

4.3 The Algorithm for In-the-Loop Training

To get better convergence and more effective training, we construct a new algorithm, the Probabilistic Random Weight Change ($PRWC$) for the LCX chip in-the-loop training based on the RWC and SA. The $PRWC$ algorithm is defined as follows:

$$w_i(n)' = w_i(n) + \Delta w_i(n) \tag{8}$$

$$\Delta w_i = \begin{cases} Lr & k = 0 \\ (-1) \times Lr & k = 1 \\ 0 & k \neq 0 \quad \& \quad k \neq 1 \end{cases} \tag{9}$$

$$k = rand \bmod m \qquad (m = 3, 4, 5, ...) \tag{10}$$

$$\begin{cases} w_i(n+1) = w_i(n)' & \Delta w_i(n+1) = \Delta w_i(n)/Lr & E(n)' < E(n) \\ w_i(n+1) = w_i(n) & \Delta w_i(n+1) = new\_\Delta w_i & E(n)' \geq E(n) \end{cases} \tag{11}$$

where Lr is the learning rate, $rand$ is a positive random value. k is the remainder of the random number divided by m. The weight w_i and weight change Δw_i in the next training epoch are calculated as equations (11). Δw_i is decided by k in equation (9), where the weight w_i is changed with probability $2/m$. The probability $2/m$ becomes smaller with decreasing error. Each adjustable weight in LCX chip will be updated with probability $2/m$ in training.

5 Experiment Results for *LCX* Chip In-the-Loop Training

To investigate and analyse the *LCX* chip's ability for function approximation, with hardware-in-the loop training, we used several different target functions, with input dimension up to six. Because of lack of space we only show 2 representative results. One of the target functions is Mexican Hat function. The plot on the right of Fig. 4 shows a one-dimensional target function and the result after 600 training epochs. The mean square error (*MSE*) is plotted on the left.

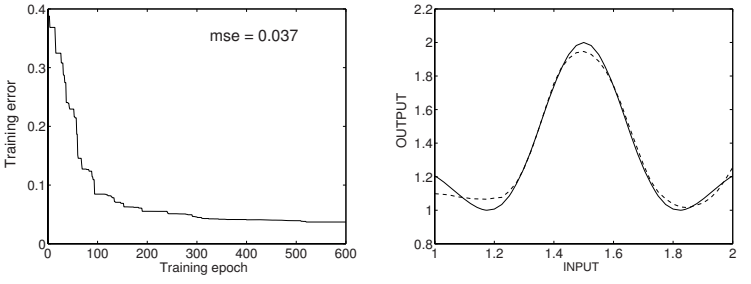


Fig. 4. The training errors and output (solid line: target, dashed line: training result)

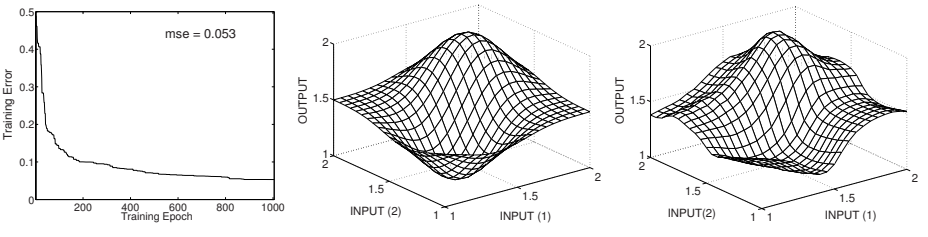


Fig. 5. The training errors, training target and output for a 2 dimensional input

Table 1. The mean square errors for all input channels in five training trials

test	channel 0	channel 1	channel 2	channel 3	channel 4	channel 5
1	0.037	0.077	0.060	0.057	0.054	0.068
2	0.034	0.045	0.032	0.063	0.048	0.064
3	0.073	0.050	0.043	0.056	0.072	0.085
4	0.051	0.065	0.037	0.042	0.044	0.049
5	0.051	0.057	0.049	0.048	0.027	0.059
average	0.049	0.059	0.044	0.053	0.049	0.065

Table 1 shows the results in statistic experiment: the final *MSE* for each of the input channels repeated five times. These results demonstrated good convergence of *PRWC* for analog neural net chip in-the-loop training.

The second target function consists of two displaced 2D Gaussian functions of opposite sign. Fig. 5 shows the training error, the training target and training output for a 2 dimensional input. After 1000 training epochs, the *MSE* is 0.053.

6 Conclusions

The results obtained with hardware in-the-loop training of the *Local Cluster Neural Network (LCNN)* analog chip confirm that the influence of fluctuations and deviations in an analog Neural Network chip can be compensated by in-the-loop training. The *PRWC* training algorithm has good convergence for multi-dimensional functions with up to six inputs (the number of inputs available on the chip). The speed of training is still determined by the weight updating computation carried out on the computer and the downloading of the weights into the chip. The *PRWC* algorithm is simple and suitable for implementation in hardware in a future version of the *LCNN* chip.

References

1. Geva, S., Malmstrom, K., Sitte, J.: Local Cluster Neural Net: Architecture, Training and Applications. Neurocomputing 20 (1998) 35–56
2. Körner, T.: Analog VLSI Implementation of a Local Cluster Neural Net. PhD thesis, University of Paderborn (2000)
3. Sitte, J., Körner, T., Rückert, U.: Local Cluster Neural Net Analog VLSI Design. Neurocomputing 19 (1997) 185–197
4. Zhang, L., Sitte, J., Rückert, U.: Local Cluster Neural Network Chip for Control. Proc. 7th International Conference on Adaptive and Natural Computing Algorithms. Springer-Verlag, SpringerWienNewYork (2005) 116–119
5. Liu, J., Brooke, M., Hirotsu, K.: A CMOS Feedforward Neural-Network Chip With On-Chip Parallel Learning for Oscillation Cancellation. IEEE Trans. Neural Networks 13(5) (2002) 1178–1186
6. Jabri, M., Flower, B.F.: Weight Perturbation: An Optimal Architecture and Learning Technique for Analog VLSI Feedforward and Recurrent Multilayer Networks. IEEE Trans. Neural Networks 3(1) (1992) 154–157
7. Cairns, G., Tarassenko, L.: Learning with Analogue VLSI MLPs. Proc. 4th International Conference on Microelectronics for Neural Networks and Fuzzy Systems. IEEE Computer Society Press (1994) 67–76
8. Boese, K. D., Kahng, A.B.: Simulated Annealing of Neural Networks: the 'Cooling' Strategy Reconsidered. Proc. IEEE ISCAS (1993) 2572–2575
9. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by Simulated Annealing. Science 220(4598) (1983) 671–680

Implementation of a Neural Network Processor Based on RISC Architecture for Various Signal Processing Applications

Dong-Sun Kim[1], Hyun-Sik Kim[1], and Duck-Jin Chung[2]

[1] DxB·Communication Convergence Research Center,
Korea Electronics Technology Institute,
68 Yatap-dong, Bundang-gu, Seongnam-si, Gyeonggi-do 463-816, Korea
{dskim, hskim}@keti.re.kr
[2] Information Technology and Telecommunications, Inha University,
253 Younghyun-Dong, Nam-Gu, Incheon 402-751, Korea
djchung@inha.ac.kr

Abstract. In this paper, hybrid neural network processor (HANNP) is designed in VLSI. The HANNP has RISC based architecture leading to an effective general digital signal processing and artificial neural networks computation. The architecture of a HANNP including the general digital processing units such as 64-bit floating-point arithmetic unit (FPU), a control unit (CU) and neural network processing units such as artificial neural computing unit (NNPU), specialized neural data bus and interface unit, etc. The HANNP is modeled in Veilog HDL and implemented with FPGA. Character recognition problems and Kohonen self-organization problems are applied to the proposed HANNP to justify its applicability to real engineering problems.

1 Introduction

For over a decade, artificial neural networks (ANNs) have been recognized in many fields of artificial intelligence as powerful tools for complex problem solving tasks due to their learning capability [1,2]. However, their use in real-time applications such as speech recognition, image processing and robot control often demands high performance and hardware system cost. In this reason, many ANN applications eventually require specialized neural network processor (SNNP) to achieve adequate performance at reasonable cost and a number of special purpose neural network processors specialized for the implementation of ANN architectures are proposed [6]. However, the design effort for SNNP is still in a relatively early stage and existent designed specialized neural network processors have a drawback that it is not appreciate for using in general digital signal processing due to their specialized architecture for neural network computation. In this paper, we propose a new neural network processor that can efficiently support general digital signal processing as well as ANN computation in order to overcome the limits of the previous specialized ANN processors. We have focused not

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1340–1349, 2006.

on designing and implementing a high-speed neural network processing engine itself, but also on building a hybrid neural network processor that can handle the general digital signal processing operating unit. For supporting ANN and general RISC instructions,we proposed hybrid neural network processor that is consisted of several major parts: 64-bit floating-point arithmetic unit (FPU), a control unit (CU), special data bus, interface unit, and neural network processing unit (NNPU). The proposed NNPU employs SIMD architecture for neural network processing because data intensive operators in neural networks need high computing power and have a great potential for SIMD parallel processing. Such architectural features of Hybrid Artificial Neural Network Processor (HANNP) are optimally customized for achieving high-speed digital signal processing while providing flexibility for efficient computation of the ANN. The structure of this paper is as follows: In the next section, the artificial neural networks are briefly reviewed with the proposed neural instruction set. The architecture of the proposed hybrid neural network processor is presented along with the brief descriptions of each subcomponent in the section 3. Experimental results including instruction set usage measurements and implementation result are presented in section 4, and we conclude in section 5.

2 Artificial Neural Network

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. It is composed of a large number of highly interconnected processing elements working to solve specific problems and can be implemented by means of a massively parallel processor with tools for gaining knowledge from experience. Each ANN is composed of different number of neurons and the main characteristics of ANNs are [4]:

- Adaptive Learning: The behavior of the network is changed according to the data given as input. The network is decided how it will be reacted and continued when new data is fed to the system.
- Self-Organization: The structure of the network is changed according to the data given. The structure of the network can be changed either by changing the strength between the neurons or the learning algorithm.
- Error tolerance: The network is capable of finding a generalization for new or distorted data. A model is determined for each given data.
- Real-time operation: As a consequence of parallel processing, real time operation becomes possible.
- Parallel information processing: Like the neurons in the human brain, the processing between the neurons in ANN is parallel.

2.1 Design and Structure of ANN Unit

In order to design processing units of ANN, we consider multilayer neural networks composed by source nodes which correspond to the input layer and by

one output layer of computation nodes [7]. The input layer and output layer are responsible for extracting knowledge from the environment and for the communication with the environment respectively. The hidden layers are responsible for the execution between these two layers. The signal propagates along the network from the input to the output [4]. This kind of network is called Multilayer Perceptron (MLP) and is characterized by the following three properties:

- The model of each neuron presents a smooth nonlinearity in the output, like in the logistic function.
- The network consists of one or more hidden layers between the I/O layers, learning complex problems through the progressive identification of meaningful aspects from the input data set.
- The high number of synaptic connections determines the high degree of connectivity and a change in a connection influences all the synapses population or their weights.

The MLP are frequently used to solve complex problems by a supervised learning process based on a popular technique known as back-propagation algorithm (BP). It consists of two distinct steps: a forward step and a backward step. In the former an input vector is applied to the network and its effect propagates through the layers. At the output layer, a set of values is produced as actual response of the network. That response is then compared to a set of target values and during the backward step, the synaptic weights are corrected considering the errors between real and expected output. One presentation of the entire training set is called epoch. The learning process goes on a epoch by epoch basis until the synaptic weights and threshold levels of the networks stabilization. In the back-propagation learning, a training set is usually presented to the network to evaluate the synaptic weights considering as many training examples as possible, to achieve a good capacity of the network in generalization performance [7].

2.2 Instruction Set for HANNP

The function of the proposed processor is programmed by means of 24 customized instructions for neural network specified application and general RISC instructions such as *add*, *store* and *load* as shown in Table 1. They include instructions for memory access, data transfers, arithmetic operations, and flow controls. As for the normal instruction set architecture, the processor provides typical instruction sets and most of them are executed in one internal cycle. Out of all instructions, six are for *immediate data processing*, fifteen for *arithmetic operations*, five for *data move*, nine for *branch operation*, fourteen for *data loading and store*, and twelve for *logic operation*. A 9 out of 32-bit word instruction is served to opecodes. The compute operations of both the ALU and floating-point unit are based on two operands and have the same format. Table 2 shows the neural instruction list including several special purpose instructions such as *BR, WLD*, etc. *BR* is used for for neural network specified application. The selection is made by the flag bit appended to *BR* instruction. *WLD* loads data from a local memory of its neighbor processing units or its own processing unit to register file at this time.

Table 1. Instruction Set

Categories		Instructions
Immediate Data processing	Shift operation	RSLI, RSRI, LSLI, LSRI
	Index operation	PIX, IXP
Arithmetic	Floating-point Multiplication	FMUL,FMULA
	Integer Multiplication	MUL, MULA
	Addition/Subtraction	ADD, ADC, SUB, SUBC, NEG
	Special Arithmetic operation	NORD, NORS, INTFP, FPINT, ABS, SWP
Move	Normal Move operation	MV, MVN
	Register Move operation	MVR, MRS, MSR
Branch	Compare operation	CMP, CFP, CMN, CLT, CMT, TEQ, TST
	Branch operation	B, BX
Load Store	Load operation	LDB, LDH, LDW, LDRB, LDRH, LDRW, MLD
	Store operation	STB, STH, STW, STRB, STRH, STRW, MST
Logical operation	Basic operation	AND, XOR, OR, NOT
	Bit Manipulation	EXT, SET, CLR
	Special operation	BITC, SHR, SHL, NOP, WAIT

Table 2. Neural Instruction Set

Instruction	Description	
NBR CRn,C,D	DBR or RR ← GM@CRn or CRN	Broadcast a data in a register *GRn* of a general prupose register file a data bus(D) or a ring bus(R)
NMAC Rm,Rj,S	AR ← (Rm × Rj)+AR	Multiply *Rm* and *Rj*, then add with a accumulator register *AR*
NPSUBim Rj, ♯(imm:16) C S	Rj ← Rj-♯(imm:16)	Subtract *Rn* with an immediate value (C=0: subtract in PU,C=1: subtract in CU)
NBS OPT,Rn,Rm, SH,OPT,S	Rn ← Barral shift(Rm)	Rn ← Barral shift(Rm) with shift amount of *CRj*
NWLD Rn ♯(LMAddr:12) M S	Rn ← LM@(LMAddr +RF0)	Load a data in a local memory to a register *Rn*
WST Rn ♯(LMAddr:12) S	LM@(LMAddr+RF0)← Rn	Store a data in a register *Rn* to a local memory
NSHIFT Rn,Rm	Rn← 1 bit shift (Rm)	Shift a data in a register *Rm* with one bit to a register *Rn*
NFU CRn,CRm,Cj,M,D	GM@CRn and (DBR or RR) or CRn ← NFU OR@CRm with shift offset in Cj	NFU look-up table access with a data in an *OR* of a *PU*and broadcast NFU data to a data bus(D) or a ring bus(R)

2.3 HANNP Instruction Scheduling

Generally, RISC processors offer increased performance through the use of separate functional units and sophisticated instruction sequencers. However, the order in which the instruction sequence is presented to the sequencer can still have a large impact on the performance. Also, in many RISC processors, a fixed number of instructions after each branch instruction are always executed,

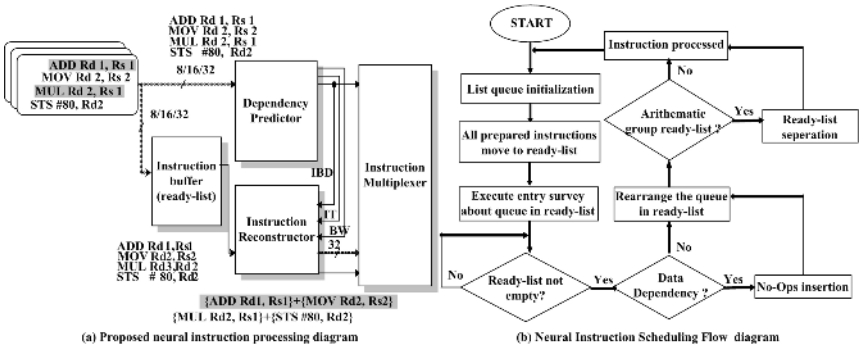


Fig. 1. Proposed Neural Instruction Scheduling

irrespective of whether the branch is taken or not and such delay slot or an instruction from a branch target after a not-taken branch increases the execution time. Thus we use instruction scheduling in conjunction with parallel processing in order to achieve high performance as following in Figure 1(a). Each of our execution units takes a certain number of cycles to execution. If we execute the code in order, then we end up with many bubbles (*NOPs*) in the resulting code, wasting CPU cycles. So, by scheduling instructions, we put more instructions through the parallel processing by filling these bubbles. We have a fixed number of execution units, such as the ALU, floating point, load/store units, neural network processing units. Proposed instruction scheduling flow is following as Figure 1(b). To schedule instructions, we first move instructions in the instruction buffer (*ready-list*), which is executed entry survey about queue. If *ready-list* is not empty, it checks the data dependency among instructions using dependency predictor. If there is dependence, then insert a *NOP* as the next instruction in instruction scheduler, which uses up one cycle. And next, we rearrange the queue and check arithmetic group within the *ready-list*. It there is no arithmetic group, then instruction is processed. To enhance the processing performance, we separate the arithmetic and neural network group from *ready-list* to execute parallel with each instructions. Repeat this process until all instructions have been scheduled. So, we can increase the throughput of CPU about 18% than normal instruction and reduce data-path bottleneck. And we more reduce power consumption with prediction scheme.

3 Architecture of the Proposed HANNP

The HANNP is consisted of several parts of RISC processor more special than required for general digital signal processing and artificial neural network computing module. The HANNP is superscalar based RISC architecture and calculates ANN instruction's dependence and allocates instruction's processing order bringing Machine code by block unit from memory for high speed computation. Figure 2 is the block diagram of proposed HANNP. As shown in Figure 2,

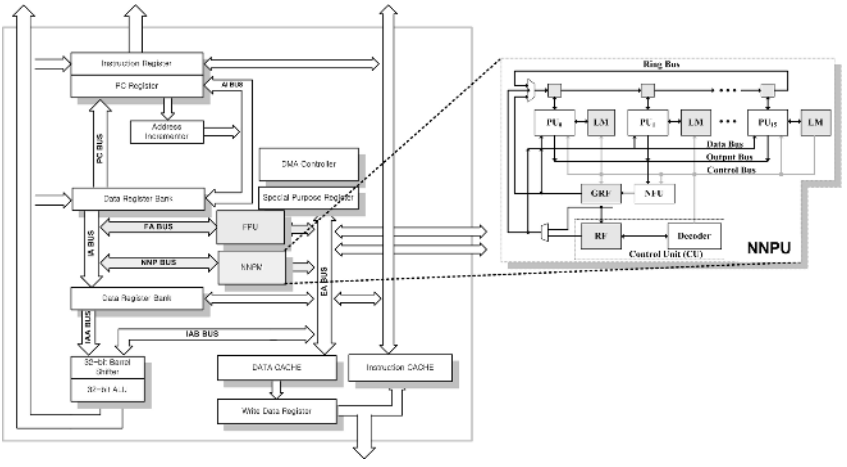


Fig. 2. Structure of the Proposed HANNP

HANNP includes the 64-bit floating-point arithmetic unit (FPU), a DMA control unit (DCU), special data bus and interface unit and neural network processing unit (NNPU). All numerical, logical arithmetic unit and neural network processing module in HANNP have a separate internal registers with bus MUXs for fast storing and loading. These units are checked data dependency by the proposed neural instruction scheduler, so this structure reduces delay time among each block as decreasing data bottleneck.

3.1 Arithmetic Unit

In HANNP, Floating Point Arithmetic Unit (FPU) and Arithmetic Logical Unit (ALU) construct a basic arithmetic group. Devised FPU basically supports single precision and double precision according to IEEE 754 - 1985 standard. In order to express an real number, if $s \subseteq \{0, 1\}$, e is positive integer and significand is a positive real number,$N(s, e, f) = (-1)^s \cdot 2^e \cdot f$. Standard is defining 4 kinds of floating point format and these formats are divided into basic format of single precision and double precision and extended format. In case of single precision, biased exponent add 127 to real exponent and in case of double precision, it adds 1023 more. So, range of exponent is from “00000001” to “11111110” and floating point can express in order according to absolute value. Also, Mantissa is decided to express by $1.fffff \ldots f$. As “1.” of front is fixed bit, it is defined implied bit and express 23bit’s mantissa part. Number expression method of this *IEEE*-754 standard can greatly divide to three parts of Normalized Number, Special Number and Denomarlized Number that implied bit is 0. Special number is consisted of infinity, *NaN* (Not a Number) and signed zero. In the case of *NaN*, is divided by *SNaN* (*Signaling NaN*) and *QNaN* (*Quiet NaN*). Basically, FPU use parallel data transmission way as considering generality and can use coprocessor interface according to user’s necessity.

3.2 Neural Network Processing Unit

The Neural Network Processing Unit (NNPU) of the HANNP employs SIMD architecture consisting of 16 processing units (PUs), a non-linear functional unit (NFU), and a control unit (CU), which are connected through two global data buses, one control bus, and a ring bus as shown in Figure 2. PU is a basic computation unit of the NNPU and is integrated 16 PUs in one neural net processor [7]. Each PU consists of 32-bit fixed point numerical arithmetic units, a 32-bit 16-word register file, 16-bit 1.5k-word LM, special purpose registers (*CR,* *FR*, and *AR*), and an address modifier (AM). In addition, the result of multiplier is bypassed to adder for MAC (multiply and accumulate) operation. The adder has the ability to perform the local memory addressing by adding the offset value stored in the *RF0* register and the address field of *WLD* (or *WST*) instruction. The embedded LM is used to store information such as weights, coefficients, image, and other data according to the applications. The instruction program is stored in the embedded program memory; on the other hand, the data are distributed in embedded local memories (LMs) and external data memory. The global data bus and ring bus allow data broadcast and PU-to-PU data transfer. The CU generates the control signals for all PUs and allows address jump and branch functions. The NFU is a look up table memory that realizes an arbitrary non- linear function. GRF (global register file) is used to store data from NFU. The data in GRF are to be broadcasted to PUs through the data bus or the ring bus. Logical Arithmetic Unit is designed to do arithmetic operations such as *barrel shift*,*NOT*, *OR*, *AND*, *XOR*, *NOR*, *NAND* and *NXOR* for various bitwise logic arithmetic computations. The Accumulate Register accumulates the results of the MAC command. The Count Register is a 12-bit counter and Flag Register is a 1-bit special purpose register to stop the operation of the PU. The Output Register is a special purpose register used to communicate the calculation result, or to store it in outside memory. The Ring Shift Register is used to broadcast data from the outside or to communicate between PUs. To lower the power and hardware design complexity, we reduced the overhead and memory access numbers by using the Register File. Particularly, each PU contains an AM which enables the proposed processor to have functionalities of both column-wise data fetch and row-wise data fetch. The importance to do so is that many linear algebra applications require series of matrix- by-vector and transposed matrix-by-vector multiplications. In ANNs, the matrix contains the synaptic weights and the vector does input values or error values. The matrix element accessing direction is dependent on the Processing state. AM model is an operational model of how an AM works on MLP with back-propagation. Here, a row of the forward weight matrix is allocated to each PU. The first is feed-forward operation, in which the network computes the equation,$u_i = \sum_j^n s_j \omega_{ij}$ and the second is error back-propagation that computes the equation,$e_j = \sum_i^m \delta_i \omega_{ij}$. From these two equations, the weights distributed over LMs should be accessed in two different modes; the row-wise for the feed-forward and the column-wise for the error back-propagation. In the proposed processor, three mechanisms, ring, bus, and AM are used for effective memory access for BP. For the process

of feed-forward operation, the address is broadcast to all PUs simultaneously through bus since the weights are stored in local memories in row order. In error back-propagation phase, the AM calculates a new address using modular operation for column base memory access. Previous error values are shifted to next PU through ring register.

4 Verification and Implementation Result

The functionality and the performance of the proposed HANNP are verified through instruction level simulation, HDL simulation and FPGA implementation. An instruction level simulator is developed with C^{++} language as a development tool. The HDL level simulation results were compared to instruction level simulation. First, in case of the 16-classes self-organization with Kohonen algorithm, each PU presents a class and the class mean value is stored in the register file in each PU. The input sample is broadcasted through global data bus and each PU computes the distance between class mean and the input sample. In order to determine the rank of distance to find the closest class, the *CR* in each PU decides the rank of its class. This is done by increasing *CR* when the each PU's distance is smaller than the data shifted into the PU. This shift operation is repeated until the ring shift complete one loop. According to the rank of each class, the class mean stored in each PU are modified when the PU's rank is higher than the predetermined value. Figure 3 shows the experimental result and shows the successful self-organization. The number of classes can be easily increased by assigning several classes per PU and

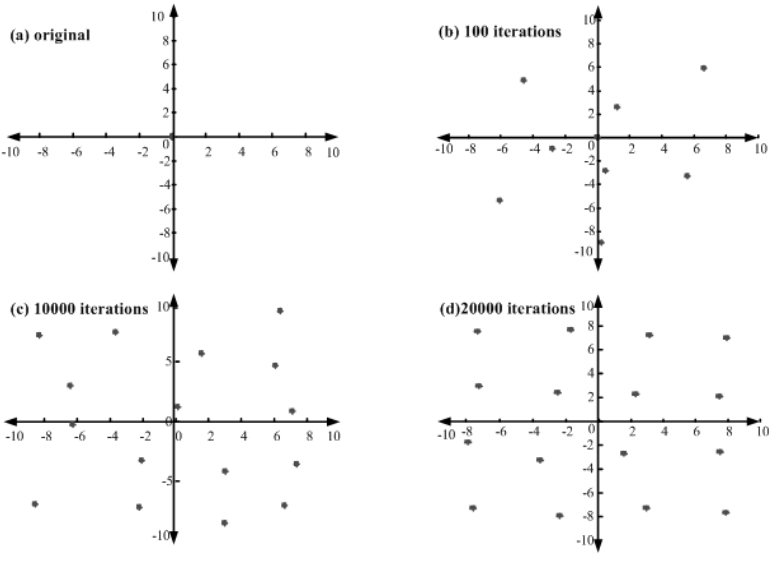


Fig. 3. Simulation result of Kohonen Self-Organized Map using the proposed processor

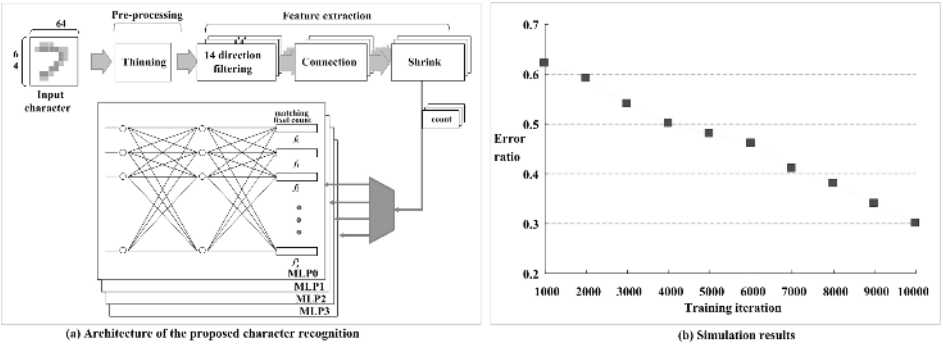


Fig. 4. Pattern recognition application using the proposed processor

store class mean in each LM. Figure 4(a) shows the overall architecture of the proposed character recognition system. The pre-processing and the feature extraction consist of 4 stages; *thinning*, *image filtering*, *connection*, and *shrinking*. These operations are based on two dimensional morphological filtering [5]. The *thinning* skeletonizes an input image while preserving its original shape. After that, the skeletonized image is filtered with 12 two-dimensional morphological feature filters like direction, angle, crossing, and T-crossing filters. Each feature extraction filtering is performed in each PU and the filter weight is stored in LM. The input image is broadcasted through the data bus and the filter output is stored in LM. During *image filtering* certain lines may be broken. These broken lines have to be reconnected by *connection* process that is a morphological dilation. Then the object in the resulting image is shrunken down until only one point remains for each object. The number of renaming point is counted for each feature filter and it represents how many corresponding features are in the input image. These numbers are used as an input vector for the classifier that is realized through MLP with back-propagation learning. Figure 4(b) shows the experimental result of the pattern recognition application. It shows that training was performed correctly. The operating clock is 55MHz and operating voltage is 3.3V. CU is implemented with 32-bit × 4Kword program memory, 32-bit × 16-word register file, program counter (PC) and 24-bit adder. NFU is consisted of 8 bit × 512-word memory and BUS is 32-cell 16-bit ring chain.The implemented recognition system is trained with 20 sets of 17 handwritten alphabets for l,770 iterations on incremental learning mode. Figure 4 shows that 4 MLPs successively trained. It took 24 seconds for pre-processing, feature extraction, classification, and learning. In order to compare its processing time, the application was implemented by using C^{++} program running on 2.8GHz Pentium IV personal computer with 1GB SDRAM, and its processing time was l4 seconds. The proposed processor showed no more than 1.7 times slower performance than PC-implementation, but nevertheless it run with relatively very slow operation clock of 55 MHz and small memory capacity of 256KB embedded SRAM. Suppose the

proposed processor is implemented as a chip using 0.18μ process technology, it is expected to operate at 400MHz clock speed, and then its computing power could be over 4 times faster than 2.8 GHz PC.

5 Conclusion

This paper proposed a high performance neural network processor that can efficiently support general digital signal processing as well as ANN computation. The proposed processor, HANNP, has an unique RISC-type customized 24 instruction set dedicated to the effective use of artificial neural network computing module and does not specialize it for any particular application. From the architectural point of view, the characteristic of the proposed processor is RISC structure with general digital processing units and neural network processing units. The Character recognition problems and Kohonen self-organization problems are applied to the processor in order to justify the validity of the design and the applicability of the processor to real engineering problems. The resulting designed processor provides high performance at computational complexity on its prime target application, biological sequence analysis, as well as on other digital signal processing applications.

References

1. Shiva, S.G. (ed.): Pipelined and Parallel Computer Architectures. Harper-Collins, New York. (1996)
2. Boulet, P., Fortes, J.A.B.: Experimental Evaluation of Affine Schedules for Matrix Multiplication on the MasPar Architecture. Proc. lst International Conf. on Massively Parallel Computing Systems. (1994) 452–459
3. Lam, K.D., Pattnaik, V., Seung-Moon, Y., Torrellas, J., Huang, W., Kang, Y., Zhenzhou, G.: FlexRAM: Toward an Advanced Intelligent Memory System. In Proceedings of International Conf. on Computer Design99. (1999) 192–201
4. Chong, F., Oskin, M., Sherwood, T.: Active pages: A Computation Model for Intelhgent Memory. Proc 25th Annual International Symposium on Computer Architecture. (1998) 192–203
5. Salembier, P., Brigger, P., Casas, J. R., Pardas, M.: Morphofogical Operators for Image Elllld Video Compression. IEEE Trans Image Process. 5(6) (1996) 881–898
6. Yentis, R., Zaghloul, M. E.: VLSl Implementation of Focally Connected Neural Networks for Solving Partial Differential Equations. IEEE Trans. Circuits Syst. I, Fundamental Theory Appl. 43(8) (1996) 687–690
7. Dongsun, K., Hyunsik, K., Hongsik, K., Gunhee, H., Duckjin, C.: A SIMD Neural Network Processor for Image Processing. ISNN 2005. LNCS 3497, (2005) 665–672

Fully-Pipelining Hardware Implementation of Neural Network for Text-Based Images Retrieval

Dongwuk Kyoung and Keechul Jung

HCI Lab., College of Information Science, Soongsil University, Seoul, South Korea
{kiki227, kcjung}@ssu.ac.kr
http://hci.ssu.ac.kr/

Abstract. Many hardware implementations cannot execute the software MLPs' applications using weight of floating-point data, because hardware design of MLPs usually uses fixed-point arithmetic for high speed and small area. The hardware design using fixed-point arithmetic has two important drawbacks which are low accuracy and flexibility. Therefore, we propose a fully-pipelining architecture of MLPs using floating-point arithmetic in order to solve these two problems. Thus our design method can implement the MLPs having the processing speed improved by optimizing the number of hidden nodes in a repeated processing. We apply a software application of MLPs-based text detection that is computed to be 1722120 times for text detection of a 1152×1546 sized image to hardware implementation. Our preliminary result shows a performance enhancement of about eleven times faster using a fully-pipelining architecture than the software application.

1 Introduction

In the text-based image indexing, text detection is important as a prerequisite stage for optical character recognition (OCR). We use multi-layer perseptrons (MLPs) to make a texture classifier that discriminates between text pixels and non-text ones [1-2]. Its application requires real-time processing and need to be implemented as hardware.

The hardware implementation of MLPs fundamentally uses a fixed-point or a floating-point arithmetic. Although the software implementation of MLPs uses the floating-point arithmetic for high accuracy, most of the hardware implementations use the fixed-point arithmetic because of small area, high speed and ease of implementation. The existing MLP applications implemented in software require data conversion from floating-point value to fixed-point value in order to utilize hardware implementation using fixed-point arithmetic.

Therefore a recent hardware implementation attempts to apply the floating-point arithmetic because the fixed-point arithmetic has two disadvantages: low accuracy and flexibility [3-4]. For example, A 10bit reduced-complexity VLSI circuit can be used in place of a 32bits floating-point processor to speed up some NN applications [3]. Wust [4] uses a floating-point representation for design parameters. This method shows good output performance with low complexity but since it inherently runs on fixed-point, it does not possess the large dynamic range capability of floating-point.

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1350 – 1356, 2006.

MLP is a parallel and distributed network of simple non-linear processing units interconnected in a layered arrangement. At present, there are three ways of parallel computing in hardware implementation of NN: layer parallelism, neuron parallelism, and link parallelism [5]. The computing unit of Xiaobin et al. [6] is designed for hardware implementation of MLP. In the computing unit, link parallelism is designed between the input layer and hidden layer of MLP; neuron parallelism is designed between the hidden layer and output layer. This design has high speed and small area using 8 bits fixed-point arithmetic. However the floating-point arithmetic is difficult for the neuron parallelism to adapt, because the floating-point arithmetic generally consists of a sequential circuit using pipelining[1] technique in order to improve speed.

We propose a fully pipelined design using floating-point arithmetic for real-time processing of MLPs-based text detection. The proposed architecture uses the link parallelism between the input layer and hidden layer and it uses the modified neuron parallelism between the hidden layer and output layer. Therefore, the proposed architecture has three advantages: the short cycle due to fully-pipelining architecture, high accuracy using floating-point arithmetic, high flexibility because software applications of NN use floating-point arithmetic. Our preliminary result shows a performance enhancement of about eleven times faster using the proposed design in FPGAs than the software application.

2 Hardware Implementation

The texture classifier consists of two-layers, such as input, hidden layer, and output layer. We use a 11×11 window as an input to the MLPs-based a texture classifier. The input values are computed by using floating-point arithmetic (i.e., addition unit and multiplication unit) and sigmoid unit.

2.1 Overall System Architecture

We illustrate the structure of simple two-layers MLPs before explaining an overall pipelined architecture. Fig. 1(a) shows a working process of the proposed architecture. All inputs and weight values were computed as addition, multiplication, sigmoid function. Hidden j works at one stage except the first node. All nodes of output layer are computed by using the modified node parallelism. The processing of all output nodes is completed after the processing of all hidden nodes, because we design the overall pipelined architecture of MLPs (Fig. 1(b)).

We will brief the data structure of floating-point, and also addition, multiplication and sigmoid function, before we illustrate the overall pipeline architecture in detail. Floating-point arithmetic of IEEE standard [7] is composes of a sign bit, 8 bits exponent and 23 bits significant.

[1] The Pipelining is the use of a pipeline which is the continuous and somewhat overlapped movement of instructions to the processor or in the arithmetic steps taken by the processor to perform an instruction.

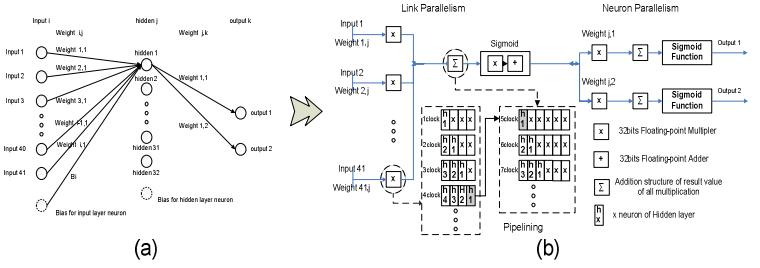


Fig. 1. Overall system: (a) A working process of MLP, (b) architecture of MLP

The floating-point operators (i.e., addition and multiplication) are designed in a pipelined architecture. The multiplication and addition are composed each of 6 and 5 steps. One of the important computational elements within a NN is the nonlinear activation function [8]. A commonly used type of nonlinear function is the sigmoid function which has the advantage of having a simple differential equation

$$f(x) = \frac{1}{1+e^{-x}}. \tag{1}$$

The sigmoid function is a very important part of hardware implementation of NN. Because sigmoid function should complete a process of nonlinear mapping and includes several types of operation such as exponent, addition and division, much time and large hardware resource should be consumed if it is realized directly. We implement a Piece Wise Linear (PWL) to solve this problem [9]. Only PWL for sigmoid function needs both addition and multiplication because PWL joins linear function. In this paper, we implement sigmoid function for simple implementation using 3 segment of PWL method(i.e., $y_1(x)=0$, $y_2(x)=(1/2)(1+x/2)$, $y_3(x)=1$). The proposed sigmoid function designs a pipelined architecture using pipelined units of floating-point arithmetic, such as addition and multiplication.

2.2 Link Parallelism Between Input Layer and Hidden Layer

The architecture of addition units is decided at the number of inputs. A symbol i is define the number of inputs. If $i = 2^n$, then the number of addition layer is n, and also, if $2^{n-1} < i < 2^n$, then the number of addition layer is n. Moreover, the number of adder units is $i/2$ and the number of multiply is $i/2+1$. The number of input values is 42 include a bias. In this case, The architecture of between input layer and hidden layer is designed like Fig. 2 (i.e., $2^5 < i < 2^6$). Fig. 2 shows 3 buffer units. Three buffers have to keep resulting values which don't processing in each layer. One buffer keeps as 6 stages because addition unit is designed as 6 stages.

If the number of any layer inputs is an even number, then the architecture is not need with a buffer, otherwise it is need with a buffer due to adder unit does not compute as one value. The pipelined architecture of between input layer and hidden layer spends one stage of every node except the first one.

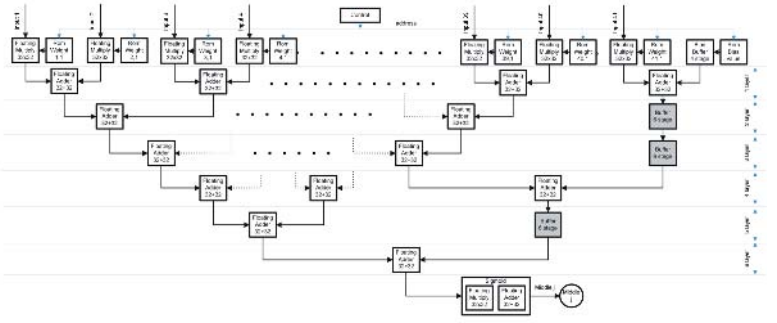


Fig. 2. Architecture of 41 inputs

2.3 Modified Node Parallelism Between Hidden Layer and Output Layer

The architecture of between hidden layer and output layer designs a different model between input layer and hidden layer. It is illustrated in Fig. 3 for the architecture between hidden layer and output layer. The architecture of output layer is designed for pipelining stage like Fig. 3. It consists of processing units as the number of output nodes. A processing unit for one of output nodes is composed of a multiply, a few adder include buffer, and a sigmoid unit. Fig. 4 shows the architecture of adders.

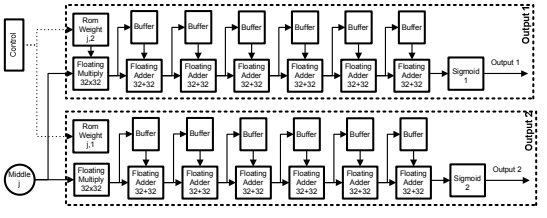


Fig. 3. Modified Node Parallelism between Hidden layer and Output layer

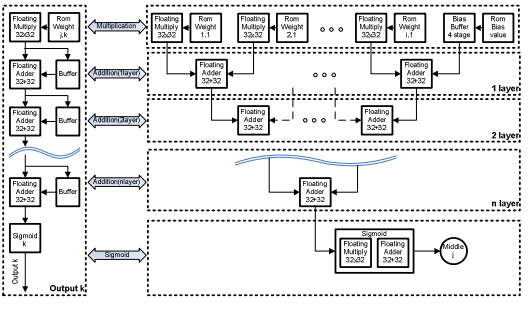


Fig. 4. Relation between architecture of both output layer and hidden layer

Addition unit is decided the number of hidden nodes (see Sect. 3.2). An addition un it with a buffer unit is able to compute a layer like Fig. 4. The buffer unit stores a float ing-point number for an input value of addition unit. A node in the output layer is com pute with a multiplication unit, a sigmoid function, and some addition units. Addition units, which are decided at the number of hidden nodes, work with buffers for the pip elining processing.

3 Results

In this paper, the overall pipelined architecture is implemented in FPGA of XC2VP100 provided by Xilinx. The VHDL with Xilinx ISE 7.1 as synthesis tool is the integrated development environment.

The MLP is the following: two-layers, 41 inputs, 32 nodes in the hidden layer and 2 nodes in the output layer. Table 1 shows analysis of two networks. The first processing performs each stage. Though the proposed architecture spends many stages in case of the first processing, the architecture spends a few stages (i.e. the number of hidden nodes) except the first one.

Table 1. Analysis of MLPs

Neural Network	Area (gate)	Frequency (MHz)	1st complete No. of Stages	1st complete Runtime (ns)
MLPs	1,426,312	54.259	111	2045.73

Table 2 shows the sequential processing result with a MLPS. The result of the MLPs is 32 stages (i.e., 32 nodes in the hidden layer). The runtime in between the first and the second is faster than the first one.

Table 2. Analysis of sequential processing for MLPs

Neural Network	2nd complete No. of Stages	3rd complete No. of Stages	…	1722120 rd co mplete No. of Stages	Runtime (ns)
MLPs	32	32	32	32	589.76

The MLPs tests processing time for a 1152×1546 sized image on a 2.4 GHz Pentium IV PC with 512 Mbytes of memory. It is computed to be 1722120 times for text detection of a 1152×1546 sized image (Fig. 5).

As shown in Table 3, we get proposed architecture using FPGA compared to CPU-only processing. Our preliminary result shows a performance enhancement of about eleven times faster using the proposed design in FPGAs than the software application.

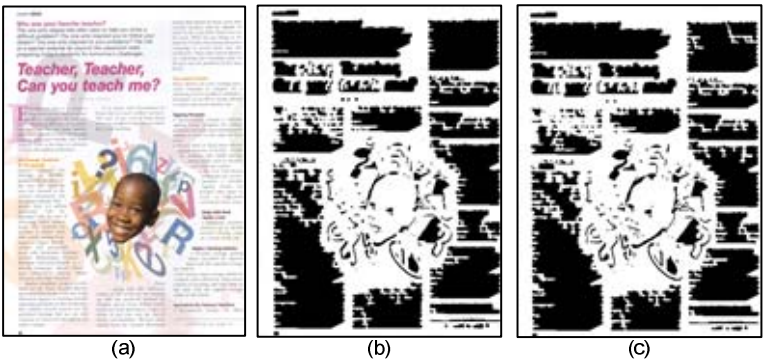


Fig. 5. Experimental Results: (a) test image, (b) result of CPU, (c) result of FPGA

Table 3. Processing times of MLPs-based a texture classifier

FPGA	CPU
1.04(s)	11.73(s)

4 Conclusions

Many NNs are implemented as hardware, because many NN applications require real-time processing. Hardware implementations of NN are very important that we use one of both floating-point and fixed-point. Floating-point arithmetic has a high accuracy a nd use with converting the data values of the general NN, otherwise fixed-point arith metic has a high processing speed and occupy a small area.

We propose the architecture to have a high processing speed, though we use floatin g-point arithmetic. This architecture is used a fully pipelining technique. The fully pip elining technique is able to work for stages which are the number of hidden layer nod es except the first result. Therefore, the fully pipelined architecture can improved a cy cle time.

Acknowledgement

This work was supported by the Soongsil University Research Fund.

References

1. Jung, K., Han, J.H.: Hybrid Approach to Efficient Text Extraction in Complex Color Images. Pattern Recognition Letters 25 (2004) 679-699
2. Jung, K., Kim, K.I., Kurata, T., Kourogi, M., Han, J.H.: Text Scanner with Text Detection Technology on Image Sequences. The 16th IEEE International Conference on Pattern Recognition 3 (2002) 11-15

3. Watkins, S.S, Chau, P.M: Reduced-complexity Circuit for Neural Networks. Electronics Letters Volume 31, Issue 19 (1995) 1644-1646
4. Wust, H., Kasper, K, Reininger, H.: Hybrid Number Representation for the FPGA-realization of a Versatile Neuro-processor. Euromicro Conference, 1998. Proceedings. 24th , 2 (1998) 694-701,
5. Ossoinig, H., Reisinger, E., Steger, C., Weiss, R.: Design and FPGA-Implementation of a Neural Network. Signal Processing Applications & Technology, Boston, USA (1996) 939-943
6. Ma, X., Jin, L., Shen, D., Yin, J.: A Mixed Parallel Neural Networks Computing Unit Implemented In FPGA. IEEE Int. Conf. Neural Networks & Signal Processing (2003) 14-17
7. IEEE Standard for Binary Floating-Point Arithmetic. ANSI/IEEE std 754-1985, New York, The Inst. Of Electrical and Electronics Engineers, Inc, Aug, 1985.
8. Hidvegi, A.: Implementation of Neural Networks in FPGA. http://www.sysf.physto.se/~attila/ANN.pdf (2002)
9. Basterretxea, K., Tarela, J.M., del Campo, I.: Approximation of Sigmoid Function and the Derivative for Hardware Implementation of Artificial Neurons. IEE Proc., Circuits Sys. 151 (2004)

FPGA Implementation of a Neural Network for Character Recognition

Farrukh A. Khan[1], Momin Uppal[2], Wang-Cheol Song[1],
Min-Jae Kang[3], and Anwar M. Mirza[4]

[1] Department of Computer Engineering, Cheju National University, Jeju 690-756, Korea
{farrukh, philo}@cheju.ac.kr
[2] Department of Electrical Engineering, Texas A&M University, College Station, TX, USA
momin@tamu.edu
[3] Department of Electronic Engineering, Cheju National University, Jeju 690-756, Korea
minjk@cheju.ac.kr
[4] Faculty of Computer Science and Engineering, GIK Institute, Topi, Pakistan
mirza@giki.edu.pk

Abstract. Neural Networks are usually implemented in software on sequential machines but when implemented in hardware, they are extremely fast due to the massive parallelism inherent in the hardware devices. Implementation of Neural Networks in Programmable Logic Devices such as FPGAs (Field Programmable Gate Arrays) gives us more flexibility since these devices are reconfigurable and their design can be altered whenever needed. The design proposed in this paper shows the implementation of perceptron neural network in FPGAs for the character recognition problem. The characters here are the English language alphabets which are input to the network and after training; they are tested for recognition. Each alphabet is tested for three different fonts. After implementation, the simulations are done and performance issues of the design are analyzed. The post-layout simulation gives excellent results even if some noise is introduced to the input patterns.

1 Introduction

Several researchers have attempted to implement neural networks in hardware for different applications during the past few years [1], [3], [4], [6], [7]. A detailed survey of the hardware implementation of neural networks is presented in [5]. In this paper, we have used the character recognition problem for implementation in FPGA. In our implementation, the preceptron model is used having one input layer and one output layer. The characters are English alphabets where each alphabet has three different fonts which are input to the neural network for recognition. These alphabets are represented in the form of a 5x5 matrix which means that there are 25 input neurons in the input layer of the neural network. As our example, we take four characters to be recognized by the network, therefore, we have four different categories to which each input vector may belong; hence there are four neurons in the output layer. The synaptic connections between the input and output layers come out to be 100 since each input neuron is connected to every other neuron in the output layer. The training input patterns and targets are converted to an appropriate form for the neural network to process which in this case is the bipolar representation. Each alphabet is having three

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1357 – 1365, 2006.

different fonts each represented in the form of a 5x5 grid as shown in Fig. 1(a). In the figure, the dark filled circles are represented as a 1 and the rest of the circles are represented as -1. The target values have also been represented in the same way. The perceptron model used in our problem is shown in Fig. 1(b).

All the inputs given to the network are represented in the form of vectors having values either 1 or -1. Each input is accompanied by a corresponding target value which is compared with the output coming from each output neuron. Since there are three different fonts to be recognized for each English alphabet, there are twelve input vectors and twelve target vectors. All these twelve inputs are used for the neural network to train itself. The response of each output unit is calculated which is the weighted sum of all the inputs and the bias coming to each output unit. The activation function is applied to each output value and then it is compared with the corresponding target value depending upon which it decides if the corresponding weights should be updated or not. The training process includes updating of weights and biases whenever the value of each neuron in the output layer is not equal to the corresponding target value. Finally, there are four test vectors which are used to test the trained neural network for the purpose of recognition.

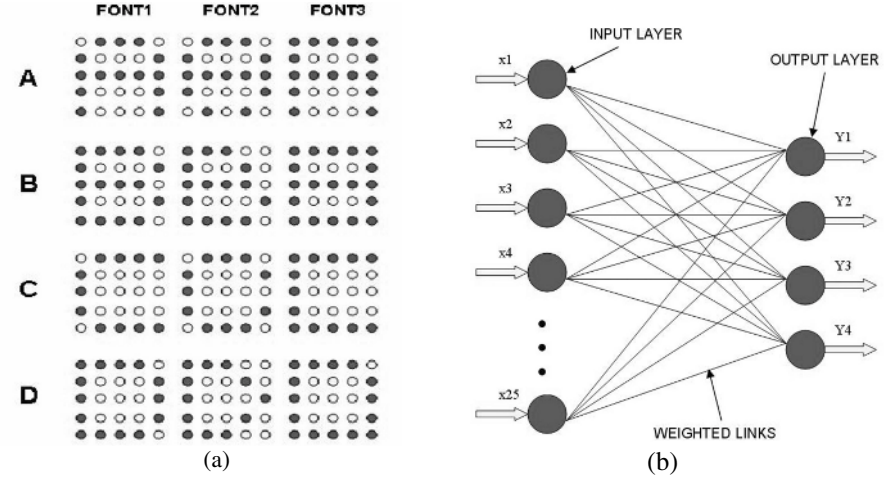


Fig. 1. (a) Input patterns with three different fonts for each character. (b) Perceptron model used for the character recognition problem.

The response of the output units can be calculated using the following equation [2]:

$$y\_in = b + \sum_i x_i w_i \tag{1}$$

The weights and corresponding bias are updated as follows:

$$w_{ij}(new) = w_{ij}(old) + \alpha t_j x_i \tag{2}$$

$$b_j(new) = b_j(old) + \alpha t_j \tag{3}$$

where, α is the learning rate parameter whose value in our case is kept as 1. The activation function used in our case is:

$$f(x) = \begin{cases} 1 & if \quad y\_in > \theta \\ 0 & if \quad -\theta \le y\_in \le \theta \\ -1 & if \quad y\_in < -\theta \end{cases} \tag{4}$$

The function has values 1, 0 or -1, depending on whether the value of the output is greater than θ , lies between $-\theta$ and θ or is less than θ. The value of the threshold θ in our case is kept zero.

2 FPGA Implementation

The FPGA implementation is done using Altera's MAX+PLUS II [9] environment (version 10.2). The device used is FLEX EPF10K70RC240-2 and the programming is done in Verilog HDL [10]. The Verilog code is compiled, synthesized and then the post-layout simulation and timing analysis is carried out. Post-layout simulation is done using the MaxPlus II Waveform Editor and the timing analysis is carried out using the MaxPLus II Timing Analyzer. There are basically two modules implemented in Verilog. One is basic-unit module and the other is top-level module. Further details of our design and Verilog code can be found in [4].

2.1 The Basic-Unit Module

The hardware design of our problem circulates around the most important part of the design termed as the basic-unit. The basic-unit comprises of a 4x1 multiplexer, four 4 bit registers, a 7 bit Add/Subtract unit and a 4 bit Add/Subtract unit. Input values are given to the 7 bit Add/Subtract unit which acts like a neuron of the neural network. It computes the response by adding the bias and the sum of corresponding weighted inputs. The important thing here is that in this design no multiplication is carried out by the Add/Subtract unit. Since all the inputs to the neural network are represented in bipolar form i.e., either 1 or -1, so we can take advantage of this bipolar representation of inputs in a way that if the input value is 1 which means x = 1 then the bias (which is also either 1 or -1) is simply added to the corresponding weight and when the value of x is -1, the weights are just subtracted from the bias. This is explained below:

The response of the output unit can be calculated by the following formula:

$$y\_in_j = b_j + \sum x_i w_{ij} \tag{5}$$

Here, When x = 1, then $y\_in_j = b_j + w_{ij}$ and
When x = -1 then $y\_in_j = b_j - w_{ij}$

So, in this way, we can easily eliminate the multiplication of the inputs with the corresponding weights which result in less memory usage and fast execution of the program since multiplication in hardware is a relatively complex process and it

consumes lot of memory space and increases the execution time. Each 4 bit register is used to store weights and these weights are selected by the multiplexer which are then input to the Add/Subtract unit for addition or subtraction with the bias value. There is another 4 bit Add/Subtract unit which is used to update the weights if the value of the output is not equal to the desired target value. The logic diagram of the basic-unit module is shown in Fig. 2.

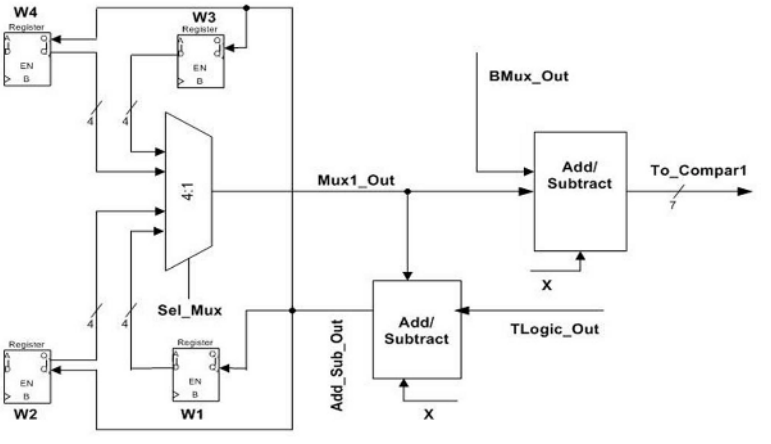


Fig. 2. Design of the basic-unit module

Since there are twenty five inputs in our design for the character recognition problem, so there would be twenty five different basic units connected to each other in a serial fashion. In the first Add/Subtract unit, the output is calculated from the addition or subtraction of the weight and the bias, which becomes the input to the next Add/Subtract unit and is again added or subtracted with the weight of that unit. This process repeats until we reach the twenty-fifth Add/Subtract unit where the total sum is calculated to give the response of the first output neuron. This process takes one clock cycle to compute the response of one output unit. Since there are four output neurons in the output layer of the network, therefore this process is repeated four times in order to get the output for four different neurons in the output layer which takes four clock cycles to get the output on four output neurons. As we are training the neural network for 12 different input patterns for three fonts each of the four English alphabets, so the whole process will be repeated twelve times to complete one iteration and one iteration would take 48 clock cycles. To sum up all this, the basic unit is mainly doing three things:

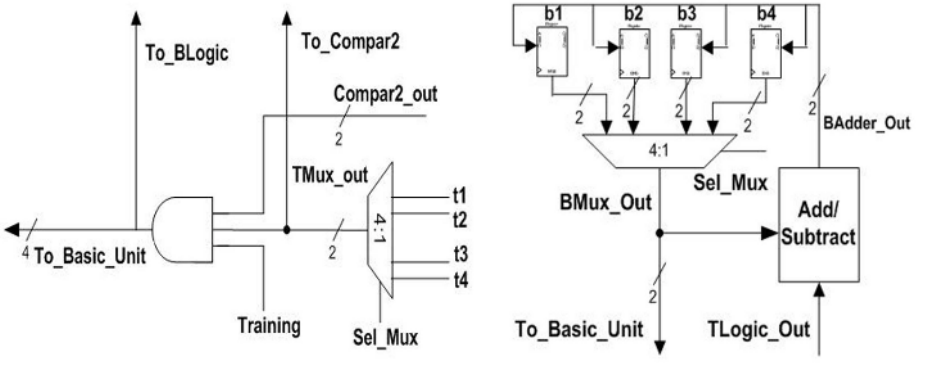
- It initializes the weights values and then passes them through the multiplexer to select one of them that is to be input to the 7 bit Add/Subtract unit.
- It calculates the response of the output unit by adding or subtracting bias with the selected weights and then calculates the overall output by summing up all the values coming out of the Add/Subtract units.

- Upon getting the value from the T-logic, it decides if the values of the weights should be updated or not. T-logic is responsible for providing the target values for comparison with the outputs. The new weights are computed in the 4 bit Add/Subtract unit.

2.2 The Top-Level Module

The top-level module contains the memory section where all the twelve input patterns are stored for training. Also, the four inputs for recognition of characters are stored in the same memory. This module also contains the control section which controls the smooth execution of the program and makes all these functions work in a proper manner. In short, in this module:

- All the 25 different instances of the basic-unit are instantiated.
- The activation function is applied i.e., the value of the output is checked against the threshold in the Comparator-1 and then the appropriate output value is set for further processing by the Comparator-2. The outputs are written at the end of training in the output registers.



(a) (b)

Fig. 3. (a) Design of the T-logic (b) Design of the B-logic

- The Comparator-2 compares the value sent by Comparator-1 with the corresponding target value.
- If the output is not equal to the target then the signal is sent from the T-logic to the basic unit's 4 bit Add/Subtract unit to update the weights. Also, the same signal is sent to the B-logic where the corresponding bias is updated.
- Memory section is located where all the inputs are stored.
- Control section is located which controls the smooth execution of the program.

The designs of T-logic and B-logic are shown in Fig. 3 (a) and (b) respectively. The simplified blocked diagram of the whole hardware design is shown in Fig. 4.

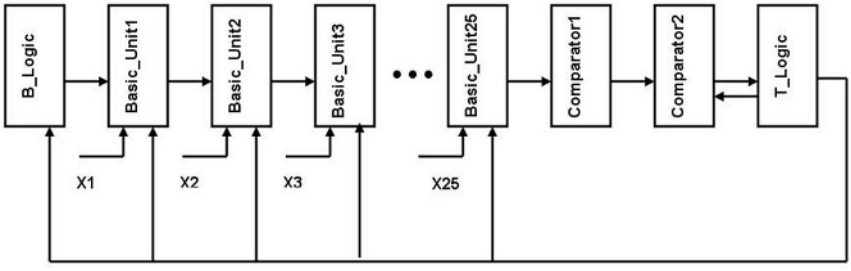


Fig. 4. Block diagram of the hardware design

3 Simulations

The software level design of the neural network for the character recognition problem has been tested in Matlab 6.1 [8] and the timing analysis is done. The hardware design is implemented in Verilog HDL and then the code is synthesized on the FLEX EPF10K70RC240-2 device in Altera's MAXPLUS II environment and the post layout simulation is carried out. After the simulation, the timing analysis is done. The results are also verified by doing simulations in the Veriwell simulator. The results obtained in our simulations cannot be exactly compared with other existing implementations of neural networks in FPGAs. The reason is that every implementation uses different simulation environment and application, and these applications are implemented on different FPGA devices. Therefore, we have shown here the results of the simulations done specifically for the character recognition application done on Altera's FLEX EPF10K70RC240-2 FPGA.

3.1 Software Implementation Results

In the Matlab implementation, the 12 input patterns are trained and then the recognition process is done on the following test patterns:

Test Pattern1 = [1 1 1 1 1 1 -1 -1 -1 1 1 1 1 1 1 1 -1 -1 -1 1 1 -1 -1 -1 1]
Test Pattern2 = [1 1 1 -1 -1 1 -1 -1 1 -1 1 1 1 1 -1 1 -1 -1 -1 1 1 1 1 1 -1]
Test Pattern3 = [-1 1 1 1 1 1 -1 -1 -1 -1 1 -1 -1 -1 -1 1 -1 -1 -1 -1 -1 1 1 1 1]
Test Pattern4 = [1 1 1 1 -1 1 -1 -1 -1 1 1 -1 -1 -1 1 1 -1 -1 -1 1 1 1 1 1 1]

The implementation gives the results as shown in Fig. 5. In the figure, *d1*, *d2*, *d3* and *d4* are the output vectors showing the four alphabets A, B, C and D recognized by the program. At the end of the results, the time elapsed represented by the variable *ans* during the execution of the whole program is also given i.e.,

Total time elapsed = 0.04 sec = 40,000 μ sec
The total time taken by the Matlab program to run the program is 40,000 μ sec.

3.2 Hardware Implementation Results

In Verilog, the input and test patterns were represented in hexadecimal form .i.e.,

x = 25'h1F8FE31; [1 1 1 1 1 1 -1 -1 -1 1 1 1 1 1 1 1 -1 -1 -1 1 1 -1 -1 -1 1]
x = 25'h1C97A3E; [1 1 1 -1 -1 1 -1 -1 1 -1 1 1 1 1 -1 1 -1 -1 -1 1 1 1 1 1 -1]
x = 25'h0F8420F; [-1 1 1 1 1 1 -1 -1 -1 -1 1 -1 -1 -1 -1 1 -1 -1 -1 -1 -1 1 1 1 1 1]
x = 25'h1E8C63F; [1 1 1 1 -1 1 -1 -1 -1 1 1 -1 -1 -1 1 1 -1 -1 -1 1 1 1 1 1 1]

```
>>
Total_Iterations =

     6

d1 =

     1    -1    -1    -1

d2 =

    -1     1    -1    -1

d3 =

    -1    -1     1    -1

d4 =

    -1    -1    -1     1

ans =

    0.0400
```

Fig. 5. Matlab recognition of alphabets A, B, C and D

The post-layout simulation results of the design for characters A, B, C, D in the MAXPLUS II environment are shown in the Fig.6.

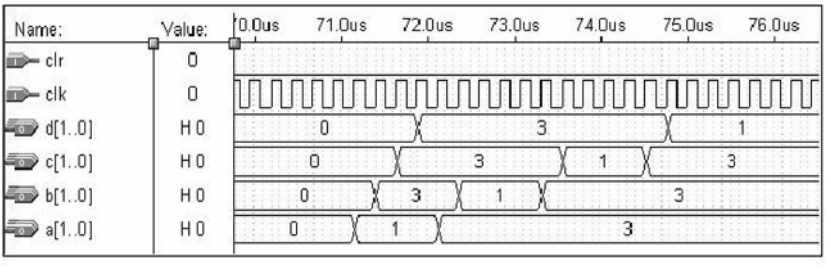


Fig. 6. Post-layout simulation showing recognition of A, B, C and D

In the figure, the alphabets A, B, C and D are recognized by the simulator. There are four 2-bit output registers a, b, c and d. Since the output is represented in bipolar form i.e., either the value is 1 or -1, so if the number is negative then it is represented

in 2's complement form. Hence, in the case of -1, we can write it as 11 which is the binary representation of hexadecimal 3. We get the output in the hexadecimal form because the inputs given to the neural network were also in hexadecimal form. The alphabet A is recognized as the target values for A are (1, 3, 3, 3) i.e., A is represented as having 1 in the first register and 3 in all the other registers. Similarly, for B the target values are (3, 1, 3, 3). Same rules apply to all other alphabets.

The same patterns for the characters A, B, C, D are also tested using the Veriwell simulator version 2.0. The results are shown in Fig. 7 which are exactly the same as we got with MAXPLUS II. For a different set of test patterns and a set of noisy pattern, we get the same results.

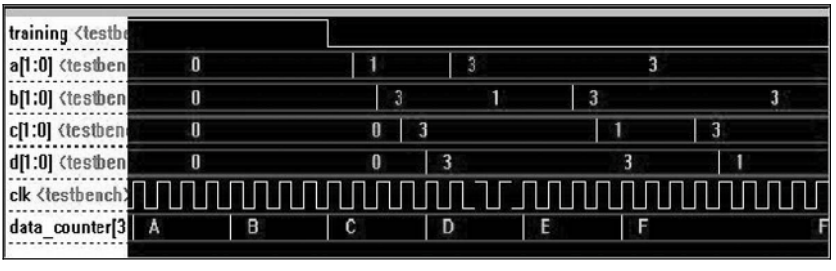


Fig. 7. Veriwell simulator results for characters A, B, C and D

This clearly shows that the neural network for the character recognition problem is implemented correctly in the hardware and it behaves exactly the same as it does in the software implementation. The timing simulation is run for 140 μsec since there are six iterations required for the neural net to train on 12 input patterns.

In the design, it requires one clock cycle to get one output at the single output unit. Since there are four output neurons in the output layer, so it takes 4 clock cycles to get output on the four output units for one input pattern. Since there are 12 input patterns for training, so we have,

12 x 4 = 48 Clock Cycles required for one iteration.

The network is trained in 6 iterations, so the clock cycles required for training are:

48 x 6 = 288 Clock Cycles

After training, we need 4 more clock cycles for each test pattern for recognition purpose. There are 4 test patterns so we need 16 more clock cycles. Therefore,

Total Clock Cycles required are: 288 + 16 = 304

For Timing Analysis, the Clock interval is taken to be 240 nsec since the minimum clock period calculated by the MAXPLUS II Timing Analyzer is 229.3 nsec at a frequency of 4.36 MHz.

Time Calculation for Training

Time Period for 1 Clock Cycle = 240 nsec

Time Period for 288 Clock Cycles = 240 x 288 = 69120 nsec = 69.12 μ sec

Hence the time required for the twelve input patterns for training the neural network is 69.12 μ sec.

Total Time Calculation

Time Period for 1 Clock Cycle = 240 nsec

Time Period for 304 Clock Cycles = 240 x 304 = 72960 nsec = 72.96 μ sec

Therefore, the total time required including the time required by the test patterns is 72.96 μ sec.

4 Conclusion

Neural network implementation for character recognition in software gives us good results when the characters are represented by less number of pixels. But when it comes to the recognition of characters having large number of input neurons, the software implementation results in slow execution of the program and the training process takes a much longer time for the neural network to learn. On the other hand, the hardware implementation gives us much faster results as compared to the implementation in software. Our work regarding the implementation of a neural network in FPGA clearly indicates that we can greatly enhance the speed of a neural network by implementing in hardware especially in programmable logic devices like FPGAs. By doing so, we can exploit their capabilities of inherent parallelism and reconfigurability. Moreover, if the higher-speed devices are used having greater number of logic gates, they can provide more flexibility and reliability by using the run-time reconfiguration capabilities. This way, we can introduce intelligence and adaptability in our design. The simulations are done in two different simulators which verify the validity of our hardware design. The simulations clearly show great enhancement in speed and high degree of performance. We get the correct results even if some noise is introduced in the input patterns. The hardware implementation is best suited for applications involving real-time processing.

References

1. Ferrucci, A.: ACME: A Field Programmable Gate Array Implementation of a Self adapting and Scalable Connectionist Network. Masters Thesis, University of California, Santa Cruz (1994)
2. Fausett, L.V.: Fundamentals of Neural Networks: Architectures, Algorithms and Applications. Prentice-Hall International (1994)
3. Khan, A.H.: Multiplier-free Feedforward Networks. Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN), Honolulu, Hawaii Vol. 3. (2002) 2698-2703
4. Khan, F. A.: Implementation of Neural Network in FPGAs. Masters Thesis, GIK Institute of Engineering Sciences and Technology, Topi, Pakistan (2003)
5. Liao, Y.: Neural Networks in Hardware: A Survey. Report, Department of Computer Science, University of California, Davis CA (2001)
6. Ossoinig, H.: Design and FPGA-Implementation of a Neural Network. Masters Thesis, Institute for Technical Informatics, Graz University of Technology, Austria (1995)
7. Seiffert, U.: Artificial Neural Networks on Massively Parallel Computer Hardware. Proceedings of European Symposium on Artificial Neural Networks, Bruges, Belgium (2002)
8. Matlab available at MathWorks Homepage: http://www.mathworks.com/products/matlab/
9. Altera Homepage: http://www.altera.com/
10. Verilog HDL Homepage: http://www.verilog.com/

A Silicon Synapse Based on a Charge Transfer Device for Spiking Neural Network Application

Yajie Chen[1], Steve Hall[1], Liam McDaid[2], Octavian Buiu[1], and Peter Kelly[2]

[1] Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool L69 3GJ, UK
[2] School of Computing and Intelligent Systems, University of Ulster, Londonderry, BT48 7JL, Northern Ireland, UK

Abstract. We propose a silicon synapse for spiking neural network application. In this endeavor, two major issues are addressed: the structure of the synapse and the associated behavior. This synaptic structure is basically a charge transfer device comprising of two Metal-Oxide-Semiconductor (MOS) capacitors the first of which stores the weight and the second controls its reading. In this work, simulation results prove that the proposed synapse captures the intrinsic dynamics of the biological synapse and exhibits a spike characteristic. The device operates at very low power and offers the potential for scaling to massively parallel third generation hardware neural networks.

1 Introduction

Biological research has accumulated a larger amount of knowledge of the structure and functions within the nervous systems of living organisms. A century has passed since the first introduction of the synapse which is a functional junction between nerve cells, at least one of which is a neuron. It is widely accepted that neurons are interconnected in a complex web-like structure and communicate with each other using pulses whose timing is used to encode information [1],[2]. Although nervous systems solve problems in real time in the way that scientists and engineers do not fully understand, a range of computational operations are possible with spiking neurons [3],[4]. Recently, much significant research has been carried out on the development and deployment of engineering equivalent implementations in both hardware and software that can be used to inspire new paradigms for real time computational networks. However, nervous systems are difficult to model and the mathematical behavior cannot be solved analytically. The performance of the most common implementations of such network on general-purpose computers is limited by the speed [5].

By contrast, semiconductor devices possess some similar attributes that facilitates the modeling of neuronal systems allowing the exploitation of the parallelism associated with these systems. However, current VLSI architectures fail to match the scale of biological networks because the fundamental building blocks (transistors) require complex circuitry to emulate a synapse [6],[7]. The single-transistor based synapses [8],[9] provide long-term nonvolatile analog storage and

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1366–1373, 2006.

local adaption in silicon, but the solid-state characteristics of these synapses are restrictive in the way they attempt to mimic real synaptic plasticity. Therefore, to advance the plausibility of neural networks realized in hardware, small geometry low power devices with behavioral characteristics comparable with real synapses must be developed.

In this paper, a compact hardware implementation of synapse is proposed as a core building block for spiking neural networks. Issues related to the modeling of the charge transfer and associated spike generation are investigated using the Silvaco software package.

2 Silicon Synaptic Model

Although biological synapses are known to exhibit extremely complex statistical behavior, only the first order models are considered in the development of neural based computational systems. Figure 1 shows a fragment of typical spiking neural network consisting of two neurons with a synaptic junction. The presynaptic neuron, i outputs a spike which is the input of the synaptic junction. At this junction, the spike is modulated according to the weight W_{ji}, and then transmitted to the postsynaptic neuron, j. The output of the synapse is known as the PostSynaptic Potential (PSP). It resembles a transient function where the rise and fall time constants are significantly different. This synaptic behavior is caused by the loading effect associated with the postsynaptic membrane time constants.

The synaptic junction is implemented as a floating gate charge transfer structure with charge storage capability, as shown in Fig.2. Essentially, it consists of two MOS capacitors in proximity and an output terminal. The charge storage is achieved by using the floating gate region of the first MOS capacitor where the stored voltage induces an inversion layer of charge at the oxide-semiconductor interface. Therefore, the stored voltage V_{ji} can be used as the synaptic weight. For the purpose of training within a network of neurons, charge would be added/removed from the floating gate through a tunneling process and suitable solutions are available from memory technology. This will be the subject of a further publication.

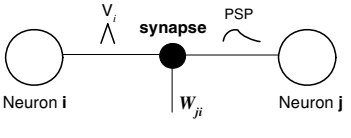


Fig. 1. Connection between two spiking neurons with a synapse

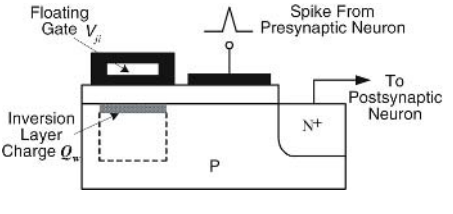


Fig. 2. Structure of the synaptic model with charge storage capability

The second MOS capacitor serves to clock the inversion layer of charge onto the postsynaptic neuron. The "clock" is the presynaptic neuron spike which releases the weighted charge package. Hence, the postsynaptic spike is modulated by the voltage stored in the body region of the first capacitor mimicking plasticity. The current flow between two capacitors is a transient, thus the power consumption can be considered negligible. Due to the requirement for close spacing of the MOS capacitors, the synaptic structure can be fabricated to deep-submicron dimensions.

3 Silicon Synaptic Operation with Spiking Behavior

In order to illustrate the principle of the proposed structure in detail, a simplified structure (floating gate is omitted) shown in Fig.3 is considered. The positive weight voltage V_{ji} controls the charge packet Q_{w} stored at the oxide-semiconductor interface of the first MOS capacitor. When V_{ji} exceeds the threshold voltage:

$$V_{\mathrm{T}} = V_{\mathrm{FB}} + 2\phi_{\mathrm{F}} + \frac{\sqrt{4\varepsilon_{\mathrm{s}}\varepsilon_0 q\phi_{\mathrm{F}} N_{\mathrm{A}}}}{C_{\mathrm{ox}}} \tag{1}$$

where V_{FB} is the flat-band voltage; N_{A} is the doping density of the substrate; C_{ox} is the oxide capacitance per unit area and other symbols have their usual meaning, the MOS capacitor operates in strong inversion causing a linear increase of the inversion layer charge arising from the thermal generation of electron-hole pairs in the depletion region. The presynaptic spike V_i controls the gate of the second MOS capacitor which will not be in thermal equilibrium as it operates in deep depletion state. A deeper potential well will be formed under the second gate in comparison to the first, causing an abrupt potential change between two gates if they are sufficiently close together. The charge packet Q_{w} therefore drifts laterally from the first capacitor to the second and subsequently to the output terminal which controls the membrane potential of the postsynaptic neuron. Note that the refractory period, which exists in a real neuron to establish its equilibrium membrane potential, could be modeled by the period of time required to reestablish an inversion layer under the first gate.

The charge transfer is driven predominately by the electrostatic forces associated with the presence of the charge and by the thermal forces responsible for diffusion [10]. Two sources of electric field are considered in acting on the charge: the self-induced field E_{S} due to the presence of the charges themselves, and the fringing field E_{F} due to the externally applied potentials on the electrodes [11],[12]. Thus, the basic equation for charge transfer in the silicon synapse can be written as the sum of three terms:

$$\frac{\partial n(x,t)}{\partial t} = \mu\frac{\partial}{\partial x}[n(x,t)E_{\mathrm{S}}(x,t)] + \mu\frac{\partial}{\partial x}[n(x,t)E_{\mathrm{F}}(x,t)] + \mu\frac{kT}{q}\frac{\partial^2 n(x,t)}{\partial x^2} \tag{2}$$

where $n(x,t)$ is the electron concentration per unit area in the storage well; x is the direction of charge propagation along the interface; μ is the surface mobility; $D = \mu\frac{kT}{q}$ is the thermal diffusion constant.

For a filled storage well, the charge transfer process is dominated by the self-induced field and is much faster than that due to thermal diffusion. During the transfer process, the charge packet Q_{w} will decay due to the lateral charge flow and the electron concentration will not be uniform as shown in Fig.3. Therefore, the potential well near the second MOS capacitor becomes deeper, and the self-induced field is in such a direction as to push the electrons towards the second capacitor. When the electron concentration drops to the point:

$$n = \frac{kTC_{\mathrm{ox}}}{q^2} \tag{3}$$

the self-induced field no longer dominates the charge transfer process and further decay is due to the thermal diffusion. In this situation, the decrease of the total charge is exponential with time constant:

$$\tau_{\mathrm{th}} = \frac{4L^2}{\pi^2 D} \tag{4}$$

Due to the coupling of the electrostatic potential caused by the interaction between adjacent electrodes, a fringing field is present during the charge transfer process even when the charge concentration is low and serves to speed up the charge transfer process. A simple estimate has shown that the fringing field is larger for thicker oxide, greater clock swing and smaller electrode length [11]. Towards the end of the transfer process, the device operates in the diffusion limit, as shown in Fig.4.

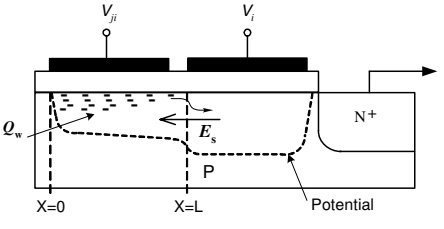


Fig. 3. Schematic drawing of a synaptic model showing the charge transfer process dominated by self-induced electric field

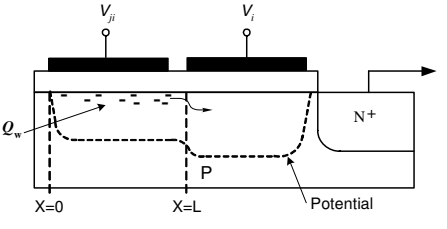


Fig. 4. Schematic drawing of a synaptic model showing the end of charge transfer process

Since the charge density in the inversion layer diminishes with time, the transfer of the charge will result in a spiking current at the output terminal. The characteristic of the spike depends on the charge concentration in the inversion layer and the time constant associated with depletion layer of the second capacitor. The charge-transfer dependent output will be followed by leakage arising from background thermally generated current, where the magnitude is insignificant compared to that of the spike. Note that the physical mechanisms realizing PSP and refractory period depend on the generation lifetime, τ_{g}, which can be engineered by appropriate defect engineering to produce the desired time response.

4 Simulation Study and Results Analysis

The Silvaco software package is employed to simulate the proposed synaptic device for spiking neural network application. A relatively large structure is used to demonstrate the basic principles. The N^+ output terminal is doped with density $10^{19}\,\mathrm{cm}^{-3}$, as shown in Fig.5. With 0.5 microns separation, two n-polysilicon gates are placed on the same p-type substrate which is doped with density $N_\mathrm{A} = 10^{15}\,\mathrm{cm}^{-3}$. The thickness of the oxide layer is set to 100 nm and the gate length is set to 4 microns in order to increase the fringing field. Thereafter, the DC and transient electrical calculations and the key parameters extractions are performed in turn. In our case, a fixed 5 V and a 3 V voltage are applied to the *Collector* and the first gate respectively. The transient voltage applied to the second gate is 5 V which is ramped over a period of 1 ns and left on until 100 ns with step 1 ps.

The biasing on the *Collector* enables the potential of the N^+ region to increase to 5.5 V as a result of the built-in potential. Compared to the electron concentration in the terminal and the acceptor concentration, the concentration under the first and second gates which are $10^{11}\,\mathrm{cm}^{-3}$ and $10^{9}\,\mathrm{cm}^{-3}$ respectively are negligibly small. The storage well of the synaptic weight is empty at this time.

A linearly increasing electron concentration due to the first gate biasing derived from the simulation is indicated in Fig.6 for voltage bigger than 0.5 V. The value is up to $3.7 \times 10^{17}\,\mathrm{cm}^{-3}$ which corresponds to $7.2 \times 10^{11}\,\mathrm{cm}^{-2}$ approximately. Clearly, the charge level is coincident with the theoretical value calculated by ignoring the depletion charge as follow:

$$n_0 = \frac{Q_\mathrm{inv}}{q} = \frac{(V_\mathrm{G} - V_\mathrm{T}) \times C_\mathrm{ox}}{q} = 6.2 \times 10^{11}\ \mathrm{cm}^{-2} \tag{5}$$

Accordingly, the storage well is full of electrons which are stored locally, due to the potential barrier and are ready to transfer.

When the 5 V transient voltage is applied to the second capacitor of the synapse, the associated MOS capacitor is driven into deep depletion and a deep potential well is abruptly formed. The electrons in the storage well therefore flow to the second MOS capacitor and subsequently to the output terminal. As described in the previous section, the self-induced and fringing fields dominate the transfer of most of the charge. Thermal diffusion is responsible for the transfer towards the end of the transient, and the fringing field enables the transfer of the remaining charge and speeds up the overall transfer process.

Figure 7 shows the profiles of surface potential along the oxide-semiconductor interface at different transient time. At the beginning of the charge transfer (0.5 ns) where the transient voltage is at 2.5 V, the potential well in the first MOS capacitor becomes oblique since the electrons are removed to the second capacitor forcing the right-hand end of first capacitor to go into deep depletion. At 1 ns where the transient voltage is 5 V, half of the charge packet has transferred to the second capacitor causing the increase of the electron

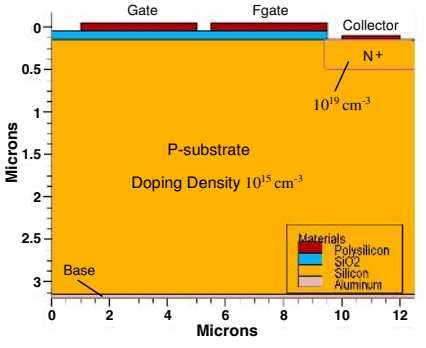


Fig. 5. Structure of silicon synapse with 4 microns electrode length and 0.5 microns spacing

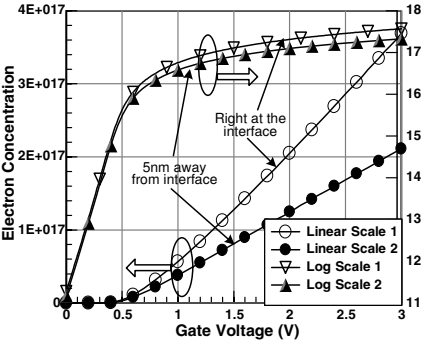


Fig. 6. Electron concentration at the interface of the storage well when weight voltage is applied

concentration there, as shown in Fig.8. The potential under the two gates increases while the potential at the output terminal decreases to about 4.8 V because of the charge transfer from storage well to the terminal. When the 5 V voltage has been applied for 5 ns, the concentration in the storage well becomes comparable to the acceptor density in the substrate which means the majority of the charge has been removed. The remaining charge continues to transfer by diffusion. As shown in Fig.7, the potential under the two gates at this time is constant with the potential at the terminal increasing towards its initial value, as there is a diminishing amount of charge coming from storage well. At the end of the transfer process, the potential returns to steady state and the electron concentration in the storage well is approximately 10^{10} cm^{-3}. The results obtained at 10 ns and 100 ns are almost the same indicating that the charge transfer process effectively stops at 10 ns. The results also indicate that the

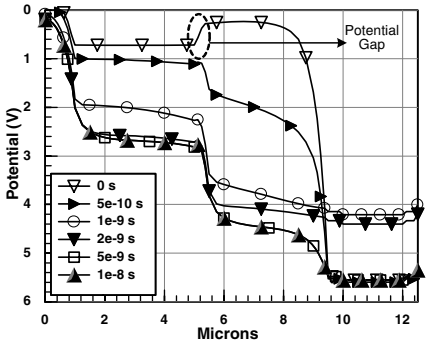


Fig. 7. Surface potential profiles at various transient stages

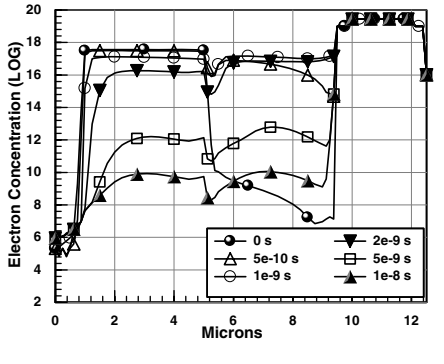


Fig. 8. Electron concentration profiles at various transient stages

electron concentration in the storage well is decreased greatly compared to the concentration shown in Fig.6.

The overall time dependency of the charge decay and the resulting current at the terminal is shown in Fig.9. If a 1 MΩ pull-up resistor and 5 V rail are connected to the output terminal, the spiking voltage can also be obtained as illustrated in Fig.10. When the electron concentration drops to $2.8 \times 10^{15}\,\mathrm{cm}^{-3}$ where the thermal diffusion constant equals to the effective drift constant, the charge transfer will be dominated by thermal diffusion. Therefore, it can be seen from Fig.9 that for times less than 2.9 ns, the charge decay is dominated by self-induced drift and the transfer efficiency achieves 99.24%. The drift dominated time is shorter than the theoretical value of 4 ns since the fringing field greatly speeds up the transfer process even though the ramp stage is responsible for a small amount of charge decay. Further decay due to thermal diffusion is exponential with constant $\tau_{\mathrm{th}} = 2$ ns for the remaining charge.

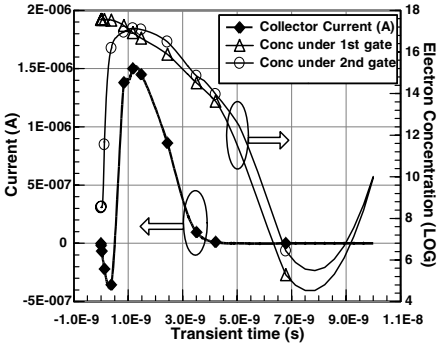


Fig. 9. Spiking current at the terminal and the electron concentration change over transient time

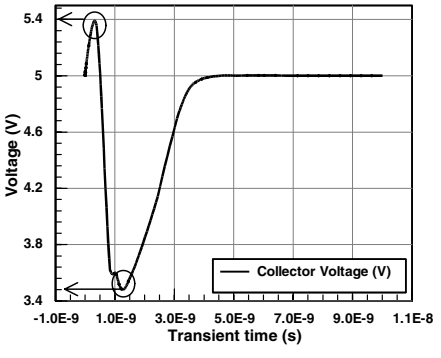


Fig. 10. Spiking voltage at the terminal obtained by connecting a 1 MΩ resistor and 5 V rail

The spiking current characteristic shown in Fig.9 indicates that the peak value is 1.5×10^{-6} A at 1.27 ns where the resulting voltage is 1.5 V. Due to the effect of the coupling capacitance between the second gate and the output terminal, the current initially decreases as indicated clearly in the figure. With increasing transient voltage, the potential gap disappears and the electrons flow towards the deep potential well causing the current rise. Over time, the remaining charge decays and the charge arriving at the output terminal diminishes. Therefore, after 1.27 ns the spiking current starts to decline. The results show that the spike characteristic of the proposed synapse depends on the charge transfer mechanism. In addition, the fall time of the spiking current is partly determined by the leakage current in the *pn* junction formed by the N^+/p-substrate junction, which depends on the electron/hole Shockley-Read-Hall recombination lifetimes. Therefore the carrier thermal generation lifetime in the substrate can be engineered to allow the fall time to be varied.

5 Conclusion

In this paper, a silicon synaptic model for spiking neural network applications was developed. The proposed silicon synapse is based on the charge transfer device with associated localized memory capability. The preliminary simulations show that the implementation of synapse represents the intrinsic dynamics of real synapse which is the dominant building block in spiking neural networks. Therefore the synaptic structure possess the potential of reducing the complexity associated with engineering the postsynaptic neuron circuits to mimic real PSP.

References

1. Maass, W., Bishop, C.M.: Pulsed Neural Networks. MIT Press (1999)
2. Gerstner, W., Kistler, W.: Spiking Neuron Models: Single Neurons, Populations, Plasticity. Cambridge University Press (2002)
3. Maass, W.: Networks of spiking neurons: The Third Generation of Neural Network Models. Electronic Colloquium on Computational Complexity TR96-031, Institute for Theoretical Computer Science, Technische Universitaet Graz, Austria (1996)
4. Thorpe, S., Delorme, A., VanRullen, R.: Spike-based Strategies for Rapid Processing. Neural Networks **14** (2001) 715–725
5. Heemskerk, J.N.H.: Neurocomputers for Brain-Style Processing: Design, Implementation and Application. PhD thesis, Leiden University, The Netherlands (1995)
6. Chicca, E., Badoni, D., Dante, V., et al.: A VLSI Recurrent Network of Integrate-and-Fire Neurons Connected by Plastic Synapses with Long-Term Memory. IEEE Trans. Neural Networks **14**(5) (2003) 1297–1307
7. Bofill-i-Petit, A., Murray, A.F.: Synchrony Detection and Amplification by Silicon Neurons with STDP Synapses. IEEE Trans. Neural Networks **15**(5) (2004) 1296–1304
8. Diorio, C., Hasler, P., Minch, B.A., Mead, C.: A Single-Transistor Silicon Synapse. IEEE Trans. Electron Devices **43**(11) (1996) 1972–1980
9. Diorio, C., Hsu, D., Figueroa, M.: Adaptive CMOS: From Biological Inspiration to Systems-on-a-Chip. Proceedings of the IEEE **90**(3) (2002) 345–357
10. Strain, R.J., Schryer, N.L.: A Nonlinear Diffusion Analysis of Charge-Coupled-Device Transfer. The Bell System Technical Journal **50** (1971) 1721–1740
11. Carnes, J.E., Kosonocky, W.F., Ramberg, E.G.: Drift-aiding Fringing Fields in Charge-Coupled Devices. IEEE J. Solid-State Circuits **SC-6**(5) (1971) 322–326
12. Carnes, J.E., Kosonocky, W.F., Ramberg, E.G.: Free Charge Transfer in Charge-Coupled Devices. IEEE Trans. Electron Devices **ED-19**(6) (1972) 798–808

Effect of Steady and Relaxation Oscillations in Brillouin-Active Fiber Structural Sensor Based Neural Network in Smart Structures

Yong-Kab Kim[1], Soonja Lim[1], and ChangKug Kim[2]

[1] School of Electrical Electronics & Information Engineering,
Wonkwang University,
344-2. Sinyong-Dong, Iksan, Chon-Buk 570-749, Korea
ykim@wonkwang.ac.kr
http://www.forlab.wonkwang.ac.kr
[2] Bioinformatics Div, National Institute of Agricultural Biotechnology, R.D.A.
225 Seodundong, Suwon, 441-707, Korea,
ncatkim@hanmail.net

Abstract. Research and development efforts are ongoing in the implementation of a totally integrated fiber-is-the sensor for structural and environmental sensing using the highly efficient Brillouin effect. Sensitivity has further improved by incorporating the smart structures for elevated actuation power, and logic memory or neural network capabilities. The presence of Brillouin and Kerr nonlinearity in the fiber together with inherent feedback delay has been shown to lead to instabilities and ultimately optical chaos. The deterministic nature of this process and its multiple bifurcations may be suppressed by a number of scheme, such as continues interference feedback, while its extremely large memory capacity exploited. These memories can be estimated as an optical logic function used for all-optic in-line switching, channel selection, amplification, oscillation in optical communication, optical logic elements in optical computation based on neural networks applications and hardware implementation.

1 Introduction

Optical fibers based on neural networks application and hardware implementation have been extensively used in optical communication systems [1], [2]. Recent interest has been also focused on using optical fibers as sensors since fiber parameters are sensitive to the fiber immediate environment [3], [4]. Important advances have been made in reducing optical losses in fibers, so the light signal can propagate in long haul transmission without requiring in-line amplifiers. The semiconductor and fiber optical amplifiers also present lossy and reflecting interfaces. Large input signals are thus required leading to nonlinear optical phenomenon in optical fibers, when signal power exceeds threshold. Specially, in the case of stimulated Brillouin scattering (sBs), part of the signal power is converted into reflected lightwave, traveling backwards towards the input of the fiber. The backward scattering nature of Brillouin scattering has long been viewed as an ultimate intrinsic loss mechanism in long haul fibers, since Brillouin threshold decreases with increasing effective fiber length. On the other hand, the very backscattering nature of this process and the existence of a threshold, provide

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1374 – 1379, 2006.

potential optical device functions, such as optical switching, channel selection, amplification, sensing, arithmetic and neural functions in optical signal processing, and neural network applications and hardware implementation. The theoretical and physical background of this nonlinear process has been well explained [5],[6]. The backward scattering scheme based on neural networks in optical fiber is shown in Figure 1.

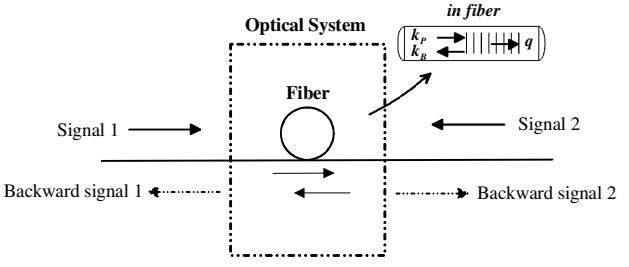


Fig. 1. Hardware implementations based Brillouin-active fiber with forward/backward scattering schemes

Active device in optical systems generally require the employment of nonlinearity, and possibly feedback for increased device efficiency. The presence of nonlinearity together with intrinsic delayed feedback has been repeatedly demonstrated to lead to instabilities and optical chaos [7], [8]. This phenomenon has extensively investigated by us for its potential detrimental effect to the Brillouin fiber sensor [9], [10].

Such a smart sensor system can potentially implement a massively parallel computational architecture with its attendant reduction in processing time while managing the complexity of the system, i.e. the sensing/actuation grid. Our SBS network would learn the correct "algorithms" by example during training and have the ability to generalize to untrained inputs after training is completed. The inputs to the network are the fiber optic sensor signal outputs, and the network outputs are the control signals for actuation controls. The true advantage of this system for application to smart sensor structures lies both in its capability to analyze complex sensor signal patterns and its speed in generating the appropriate control signal for the actuators. The key lies in the implementation of a neuron operation using SBS in optical fiber.

2 SBS Neuron Operation

The Brillouin scattering effect causes a significant proportion of the optical power traveling through the fiber transmission line to be converted into a reflected lightwave, shifted in frequency, traveling in the backward direction. The Brillouin effect can occur in a single pass through a long, low loss single mode fiber with launched power of only a few milliwatt, well within the operating range of communication systems. Chaos induced instabilities can be detrimental to an optical system in a number of ways: severe additional signal attenuation, multiple frequency shifts in some cases, and high intensity backward coupling in the transmission optics.

An artificial neuron, used in neural network research, can be thought of as a device with multiple inputs and single or multiple outputs in hardware implementations. The inputs to a neuron are weighted signals. The neuron adds the weighted signals, compares the result with a preset value, and activates if the sum exceeds threshold. In the nonlinear optical phenomenon, the system's combined weighted signals also produce an output if the weighted sum is greater than the threshold. A typical neuron, based control signal with input and output systems, is illustrated in Fig. 2.

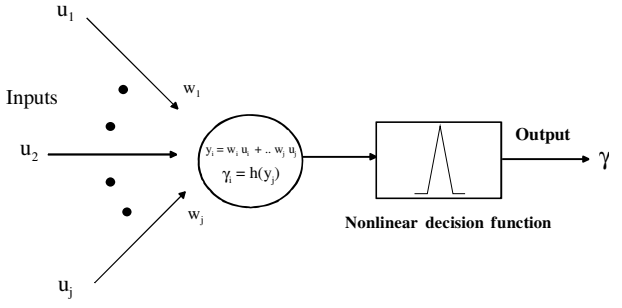


Fig. 2. A simplified multi-layered feedforward neural network. The physical resemblance on hardware implementations; optical fiber networks to nerve system and neural networks.

The system through SBS mixing combines weighted signals to produce an output if the weighted sum exceeds the threshold. The threshold decision has made by an individual neuron in conjunction with weighted inputs from other neurons. A theoretical SBS based neural network, utilizing SBS threshold sensing with an embedded sensor were explained [9], [10].

3 SBS Steady and Relaxation Oscillation

The effect of steady and relaxation oscillation, instabilities and chaos, are unavoidable in Brillouin scattering due to its intrinsic nonlinearity and feedback. Our effort is then to exploit them for device function and to suppress them by counter measures against chaos-causing parameters through the design for optimization on the one hand, and the promotion of such deterministic instabilities on the other for novel data transmission in optical communication systems. We have designed a setup for analyzing Brillouin instabilities configuration. A schematic of the setup for analyzing chaos is shown in Figure 3.

A stabilized *cw* probe laser operating at 1310 *nm* was used as a pump source for low scattering losses in the fiber, yielding a ≈13 *GHz* Brillouin scattering shift. Detection is also achieved with a 25*GHz IR* Photodetector Set (New Focus and an amplifier with 20*ps* impulse response) connected to a *HP* Oscilloscope. The temporal repetition rate of which corresponds to a pulse round-trip time in the fiber-ring taken to be less than 10 *nsec*.

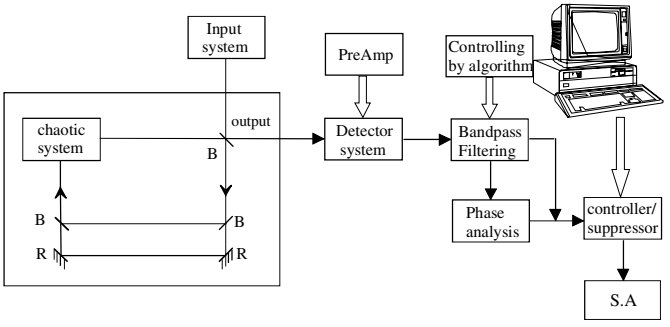


Fig. 3. Schematic diagram for controlling chaos induced instability in Brillouin-active fiber based neural network system. The optical implementation included a chaotic system configured. R is the mirror reflectivity and B is beam splitter.

Some levels of temporal instability and chaotic behavior in the backscattered intensity and also in its spectral line shift have been observed(see Fig.4). It is thus essential to know whether insertion of an amplifier in the ring will further destabilize the optical system. Since our proposed fiber sensor is based on monitoring the Brillouin spectral line shift with varying temperature and strain, the origin of the temporal chaotic behavior must be understood and its correlation to spectral line shift examined. The Brillouin signal is simultaneously displayed on a fast scope for better interpretation of the temporal process that leads to pulse train generation. The detected signal will also be viewed on a Microcomputer for comparison. The backward signal is detected with a fast detector as the laser pump power is progressively increased to maximum of 15~16 *mw*. When the pump power reaches a threshold value, a temporal structure arises in the backward signal, consisting of a periodic train of Brillouin-wave pulses as shown in Figure 4(a). The temporal repetition rate of which corresponds to a pulse round-trip time in the fiber-ring taken to be less than 10 nsec. The Brillouin pulse train amplitudes remain unstable, particularly just below pump threshold. When the observation is made using a long time scale (100μsec/division), the Brillouin output exhibits randomly distributed trains of periodic pulses. Partial stabilization of amplitude fluctuations is achieved as laser pump power approaches maximum value. These experimental features are shown in time domain in Fig. 4 (b) through (d). Instability can also occur at threshold power. The temporal evolution immediately above this threshold is periodic and at higher intensities can, for the case of a relatively broad Brillouin linewidth, become chaotic (see Fig.4 (d)). In the data presented, mechanical vibrations could be partially responsible for these Brillouin-temporal instabilities, because small amplitude fluctuations with similar frequencies were observed below the Brillouin threshold. The results attribute these Brillouin instabilities to phase fluctuations between direct and coupled pump intensity in the optical fiber systems.

At low power, the Brillouin instability can occur below SBS threshold. This is much lower than the power required for normal Brillouin process, involving single pump power. The temporal evolution immediately above threshold is periodic and at lower intensities can become chaotic. We propose to employ continuous optical feedback for control in which coherent interference of the chaotic optical signal with

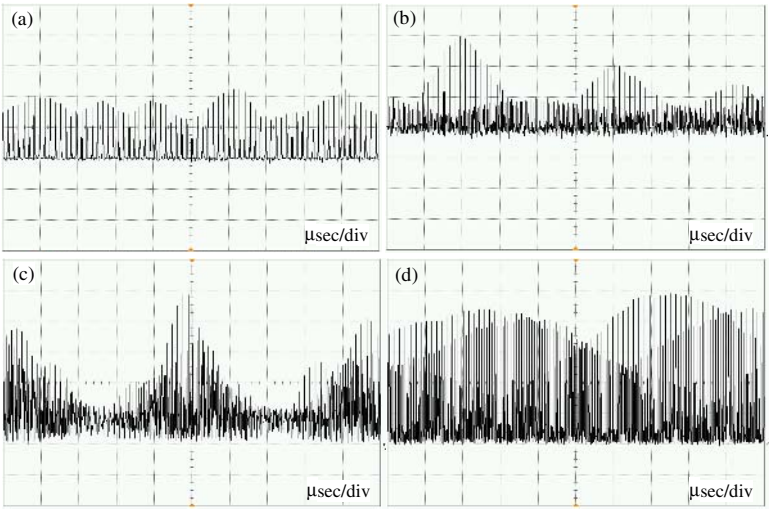


Fig. 4. Temporal structures of SBS instability vs. time; (a) before threshold, (b) immediately threshold, (c) above threshold, (d) high threshold with chaos. The whole time scale is used in 100 μsec/division.

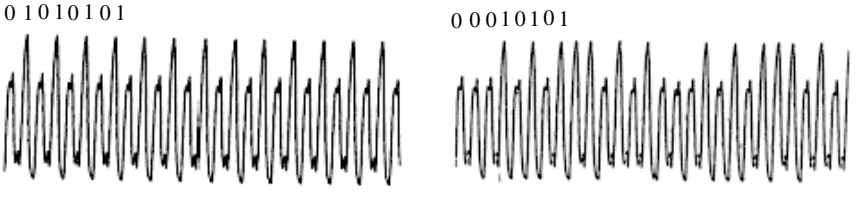


Fig. 5. Two examples of sequence of bifurcation are assigned by 0,1 symbols as (a) 0101010 and (b) 0001010

itself, when delayed, can achieve signal differencing for feedback. If suppressing by attractor proves to control chaos then, suppressing under natural chaos can be exploited as a means of sensing structural chaos.

The examples of sequence of suppression are assigned by 'low level' and 'high level' states. Multi-stable periodic states, as shown in Figure 5 (a) and (b), can lead to logic '0' or '1' and can in principle create large memory capacity as input bit streams (sequenced) in TDM network systems. Its implementation still requires much engineering improvements, such as arriving at a spatial resolution that is comparable to the references or speckle, and suppression of its tendency to chaos.

4 Conclusions

Control of SBS chaos-induced transient instability in optical systems leads to logic '*on*' or '*off*'' with multistable periodic states. It is theoretically possible to apply the

multi-stability regimes as an optical memory device for encoding/decoding messages and complex data transmission in optical communications systems. It can also in principle create large memory capacity.

Acknowledgement

This paper was supported by WonKwang University in 2005.

References

1. Gregory, D. Vanwiggeren, Roy, R.: Communication with Chaotic Lasers. SCIENCE **279** (1998) 1198-1200.
2. Hotate, K., Sean, S., L, Ong.: Distributed Dynamic Strain Measurement Using a Correlation-Based Brillouin Sensing System. IEEE. Photonics Letters **15** (2) (2003) 272-274.
3. Grossman, B., Alavie, T., Ham, F., Franke, F., Thursby, M.: Fiber-Optic Sensor and Smart Structures Research at Florida Institute of Technology. SPIE **1170** (1989) 213-218.
4. Bernini, R., Minardo, A., Zeni. L.: Stimulated Brillouin Scattering Frequency-Domain Analysis in a Single-mode Optical Fiber for Distributed Sensing. Optics Letters **29** (17) (2004) 1977-1979.
5. Koyamada, Y., Sato, S., Nakamura, S., Sotobayashi, H., Chujo, W.: Simulating and Designing Brillouin Gain Spectrum in Single-Mode Fibers. J. of Lightwave Tech. **22** (2) (2004) 631-639.
6. Tanemura, T., Takyshima, Y., Kikuchi, K.: Narrowband Optical Filter, with a Variable Transmission Spectrum, using Stimulated Brillouin Scattering in Optical Fiber. Opt. Lett. **27** (17) (2002) 1552-1554.
7. Harrison, R. G., Yu, D., Lu, W., and Ripley, P. M.: Chaotic Stimulated Brillouin Scattering Theory and Experiment. Physica D **86** (1995) 182-188.
8. Yu, D., Lu, W., Harrison, R. G.: Physical Origin of Dynamical Stimulated Brillouin Scattering in Optical Fibers with Feedback. Physical Review A, **51** (1) (1995) 669-674.
9. Kim, Y.K., Lim, S.J., Kim, H.Y., et al.: Implementation of Brillouin-Active Fiber Based Neural Network in Smart Structures. LNCS **3498** (2005) 987-991.
10. Kim, Y.K., Kim, J., Lim, S., et al.: Neuron Operation Using Controlled Chaotic Instabilities in Brillouin-Active Fiber based Neural Network. LNCS **3612** (2005) 1045-1051.

A Novel All-Optical Neural Network Based on Coupled Ring Lasers

Ying Chen[1], Qi-guang Zhu[2], and Zhi-quan Li[1]

[1] Institute of Electrical Engineering, Yanshan University, Qinhuangdao 066004, China
chenying@ysu.edu.cn, ysulzq@eyou.com
[2] Institute of Information Science and Engineering, Yanshan University,
Qinhuangdao 066004, China
zhu7880@ysu.edu.cn

Abstract. An all-optical neural network based on coupled ring lasers is proposed in this paper. Each laser in the network has a different wavelength, representing one neuron. The network status is determined by the wavelength of the network's light output. Inputs to the network are in the optical power domain. The nonlinear threshold function required for neural-network operation is achieved optically by interaction between the lasers. A simple laser model developed in the paper has illuminated the behavior of the coupled lasers. An experimental system is implemented using single mode fiber optic components at wavelengths near 1550 nm. A number of functions are implemented to demonstrate the practicality of the new network. From the experiment, a conclusion can be obtained that the neural network is particularly robust against input wavelength variations.

1 Introduction

In the recent years, telecommunications have been significantly advanced by the use of optical technology. Significant research effort has been focused on directly processing the transmitted optical information in optics, rather than requiring optical to electrical conversion and electronic processing. All-optical processing may provide many benefits such as flexibility in data rates and high speed. In particular, neural-network techniques have already been applied to the tasks of routing in telecommunication networks [1], [2]. In the future, these routing tasks could be performed completely in the optical domain [3].

An optical neural network that is for use in optical telecommunication systems must be compatible with the wavelengths used in telecommunications. Furthermore, it must be very robust and reliable to meet the strict bit error rate requirements, and operate at high speed.

There have been some attempts at obtaining fully optical neural networks using resonators and laser oscillators [4], [5]. These networks employed resonators based on photo-refractive materials and exploited gain competition between the transverse modes of the resonators.

In this paper, systems of coupled lasers are studied. It is shown that these systems can form an optical neural network that has an optical threshold function, and the network inputs are in the optical power domain. Each neuron in the network is

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 1380 – 1385, 2006.

represented by a distinct wavelength. The network status is determined by the wavelength of light output [6]. However, the system described here is not based on gain competition between lasing modes in a shared gain medium. Particularly, coupled ring lasers are considered which are implemented in single mode fiber-optic components at wavelengths near 1550 nm, and compatible with telecommunication systems. Furthermore, the system could potentially be integrated in a photonic integrated circuit, reducing the laser cavity round trip time and thereby satisfying high-speed requirements.

2 Principles of Operation

A simple ring laser implemented with single mode fiber-optic components is shown in Fig.1. The semiconductor optical amplifier (SOA) [7] acts as the laser gain medium and provides optical amplification of light traveling through it. The optical isolator allows light to travel in only one direction around the ring, thus ensuring lasing in only one direction. The wavelength filter ensures lasing at only one wavelength. The peak in the filter transmission spectrum specifies the wavelength of the laser. The coupler allows light to be coupled in and out of the ring laser.

First, we consider the operation of the solitary laser, that is, with the injected input power $P_{in} = 0$. For lasing to occur, the SOA must supply sufficient amplification or gain G, such that any losses incurred transmitting light from the SOA output back to the input are compensated for. The proportion of light transmitted from the SOA output around the loop to the SOA input is denoted by the transmittance around the loop T. T includes the transmittance of the isolator T_{iso}, the filter T_f, the coupler T_c, and losses due to component interconnections T_{com}.

$$T = T_{iso} T_f T_c T_{com} \tag{1}$$

The relation between power into the SOA P_s and power out of the SOA at the lasing wavelength P_{out} is

$$P_s = TP_{out} \tag{2}$$

As mentioned above, G must compensate for any losses in the loop for laser oscillation to occur. Once lasing occurs, G is then fixed at this specific threshold gain G_{th}

$$G_{th} = 1/T \tag{3}$$

For laser oscillation, the lasing wavelength must also satisfy the requirement that an integer number of wavelengths equal the optical length around the ring. The spacing $\Delta\lambda$ between adjacent wavelengths that satisfy this condition can be found to be

$$\Delta\lambda = \lambda^2 / n_g L_c \tag{4}$$

Here, L_c is the ring length, n_g is the refractive index in the ring, and λ is the wavelength at which lasing occurs. In our experiments, L_c is 10 m, resulting in a very small $\Delta\lambda$. The filter we use has a bandwidth much larger than $\Delta\lambda$, because they are much smaller than L_c, and the minimum bandwidth they can achieve is related to their length. Thus, it is assumed that there is always a wavelength in the filter pass-band that satisfies the ring optical length wavelength requirement. Hence, this condition on wavelength is not considered further in this paper. Furthermore, it is assumed that lasing only occurs at only one wavelength in the band-pass filter, which satisfies the optical length condition mentioned above.

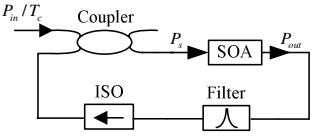


Fig. 1. The schematic diagram of a simple coupled ring laser

A neural network integrated on an optical integrated L_c circuit may be of the same order as the filter length. In this case, the choice of appropriate filter bandwidth in the optical integrated circuit design may be important.

Then, consider when external light via the coupler is also injected into the SOA

$$P_s = P_{in} + TP_{out} \tag{5}$$

Here, P_{in} is the externally injected light power arriving at the SOA input after passing the coupler. Furthermore, the wavelength of P_{in} is not at the lasing wavelength of the laser and it only passes through the SOA once, as it is blocked from making a trip around the ring by the filter. However, the wavelength of P_{in} should be sufficiently close to that of the laser, so that there is not a significant difference in SOA gain between the two wavelengths [8].

Given fixed operating conditions for the SOA, such as fixed injection current I, SOA parameters, and a fixed gain G_{th}, then, P_s is also fixed at a unique value. This value is denoted here TP_{tot}, where P_{tot} is the value of P_{out} when $P_{in} = 0$, and it has been assumed that I is sufficiently high so that G can reach G_{th}. Increasing P_s above TP_{tot} will cause G to decrease, and lasing will no longer occur as $G < G_{th}$. Thus P_{out} will fall to zero. Having P_s pegged at a unique value while maintaining sufficient gain for lasing, implies through (5) that P_{out} as a function of P_{in} is initially a straight line. The equation of the line can be found from (5) to be

$$P_{out} = P_{tot} - P_{in}/T \tag{6}$$

After P_{out} reaches zero, which occurs when $P_{in} = P_{tot}T$, it remains at zero when P_{in} is increased further.

The architecture of the neural network that is considered in this paper is shown in Fig.2. The neurons are interconnected by inhibitory connections, shown by the dotted curved lines. Inputs are connected to each of the neurons by unidirectional weighted synaptic connections. Fig.3 shows the realization of the neurons and the associated inhibitory interconnections mentioned above. The set of N neurons consists of coupled ring lasers. The arrayed waveguide grating (AWG) is an integrated optics device that provides N optical filters and additionally multiplexes the N outputs of the filters into a single output. Each filter in the AWG passes a different wavelength; thus, each ring laser lazes at a different wavelength, denoted λ_i, corresponding to input i of the AWG.

The polarization controllers (PCs) in each ring laser are used to control the polarization of the light flowing back to the input of the associated SOA. The gain through the SOAs is somewhat dependent on the polarization of the input light. For each SOA in each ring laser, the PC is adjusted so that light fed back to the SOA has the polarization necessary for maximum gain.

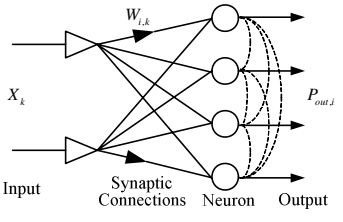


Fig. 2. The architecture of the all-optical neural network

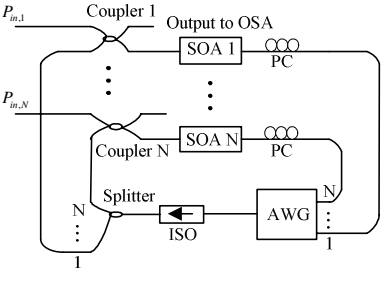


Fig. 3. Realization of the neutrons using SOAs and fiber optics

3 Experiment

3.1 Two Coupled Ring Lasers

For most of the experiments, the setup of Fig.3 was employed, in which $N = 4$. To demonstrate the threshold function of two lasers with approximately $T_{ii} = T_{ij}$, the setup shown in Fig.3 was employed, but with only lasers 1 and 2 switched on. The SOA injection currents were set asymmetrically to give $P_{tot,1}T_{11} > P_{tot,2}T_{21}$ and thus make laser 1 dominant whenever $P_{in,1} = 0$. The injection currents for SOA 1 and 2 were 132 and 150 mA, respectively, and with these currents $P_{tot,1} = 161$ mW, and $P_{tot,2} = 0.73$ mW. Note that the precise relation between $P_{tot,i}$ and the current of the specific SOA depends on the specific SOA characteristics, lasing wavelength, and losses around the ring.

External light with the wavelength of 1550.92 nm was injected into laser 1 via the coupler at the input of SOA 1. $P_{in,1}$ was varied from 0 to 0.32 mW. The values of $P_{out,1}$ and $P_{out,2}$ as a function of the injected light power are shown in Fig.4.

3.2 Recognition of 4-Bit Address

A target application of the network is in all optical processing of data packets in optical telecommunication systems. In these systems, the addresses of packets need to be recognized and routed to appropriate outputs. Current demonstrations of all-optical address recognition have address lengths of approximately four bits. Even if the address is larger, only a limited number of special patterns can be used [9]. Furthermore, the current methods do not provide very high contrast in the output decisions. Typically, electronic threshold is used after the address processor to clearly distinguish between patterns.

To demonstrate the ability of the network in address recognition, a network with two lasers and four inputs was trained to recognize the address 1010. The same methods that were used to obtain and set the input weights for the 2-bit input experiments described above were used for the 4-bit experiment. In particular, laser 2 and 4 out of the four lasers were employed. The implementation of the input synaptic connections was similar to that shown in Fig.5. The results of applying the 16 possible input vectors are shown in Fig.6. Only the input 1010 vector causes laser 2 to laze; for all other inputs, laser 4 lazes.

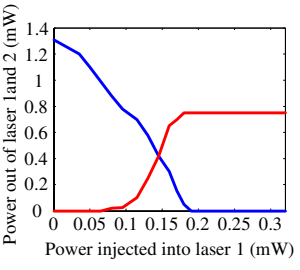


Fig. 4. Experimental results showing threshold when $T_{ii} = T_{ij}$

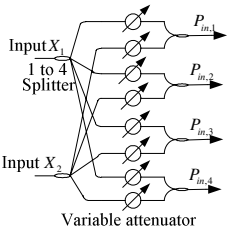


Fig. 5. Fiber optic implementation of input synaptic connections for a two-input four-laser system

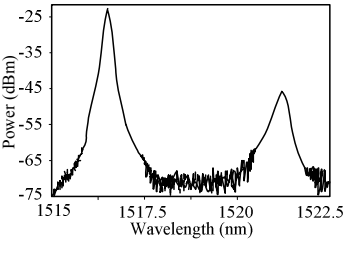


Fig. 6. Results of 4-bit address recognition experiment when input vector is1010

4 Conclusion

In this paper, a number of interesting systems based on coupled ring lasers have been studied. In particular, it was shown that two coupled lasers could provide a useful and controllable threshold or sigmoid function. A simple laser model was developed and used to explain how the coupled laser systems worked. Also the threshold functions were demonstrated experimentally. A number of 2- and 4-bit input logic functions were performed. The neural network was particularly robust against changes in input wavelength and environmental changes. This work has been targeted toward

telecommunication applications by implementing the networks at wavelengths near 1550 nm. If implemented in integrated optics, the neural network presented here could provide high-speed complex all optical logic functions required in telecommunication systems.

References

1. Giles, C.L., and Goudreau, M.W.: Routing in Optical Multistage Interconnection Networks: A Neural Network Solution. J. Lightwave Technol. 13(6) (1995) 1111-1115
2. Wang, C.J., and Weissler P.N.: The Use of Artificial Neural Networks for Optimal Message Routing. IEEE Network 9(2) (1995) 16-24
3. Frietman, E.E.E., Hill, M.T., and Khoe, G.D.: A Kohonen Neural Network Controlled All-optical Router System. Int. J. Comput. Res. 10(2) (2001) 251-267
4. Rosas, E, Aboites, V., Damzen, M.J.: FWM Interaction Transfer Matrix for Self-adaptive Laser Oscillators. Optics Communications 174(1-4) (2000) 243-247
5. Liu, Y., Hill, M.T., Calabretta, N., de Waardt, H., Khoe, G.D., Dorren, H.J.S.: Three-state All-optical Memory Based on Coupled Ring Lasers. IEEE Photonics Technology Letters 15(10) (2003) 1461-1463
6. Colak, S.B., Schleipen, J.J.H.B., and Liedenbaum, C.T.H.: Neural Network Using Longitudinal Modes of an Injection Laser with External Feedback. IEEE Trans. Neural Networks 7(6) (1996) 1389-1400
7. Zoiros, K.E., Houbavlis, T., Moyssidis, M.: Complete Theoretical Analysis of Actively Mode-locked Fiber Ring Laser with External Optical Modulation of a Semiconductor Optical Amplifier. Optics Communications 254 (4-6) (2005) 310-329
8. Hill, M. T., de Waardt, H., Khoe, G. D., Dorren H. J. S.: All Optical Flip-Fop Based on Coupled Laser Diodes. IEEE J. Quantum Electron. 37(3) (2001) 405-413
9. Calabretta, N., Liu, Y., Waardt, H.D., Hill, M.T., Khoe, G.D., and Dorren, H.J.S.: Multiple-output All-optical Header Processing Technique Based on Two-pulse Correlation Principle. IEEE Electron. Letters 37(20) (2001) 1238-1240

Author Index