Jun Wang
Xiaofeng Liao
Zhang Yi (Eds.)

LNCS 3497

Advances in Neural Networks – ISNN 2005

Second International Symposium on Neural Networks
Chongqing, China, May/June 2005
Proceedings, Part II

2 Part II

Springer

Lecture Notes in Computer Science 3497

Jun Wang Xiaofeng Liao Zhang Yi (Eds.)

Advances in Neural Networks – ISNN 2005

Second International Symposium on Neural Networks
Chongqing, China, May 30 - June 1, 2005
Proceedings, Part II

Springer

Volume Editors

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

Xiaofeng Liao
Chongqing University, School of Computer Science and Engineering
Chongqing, 400044, China
E-mail: xfliao@cqu.edu.cn

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

Library of Congress Control Number: 2005926239

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

ISSN 0302-9743
ISBN-10 3-540-25913-9 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-25913-8 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Typesetting: Camera-ready by author, data conversion by Olgun Computergrafik
Printed on acid-free paper SPIN: 11427445 06/3142 5 4 3 2 1 0

Preface

This book and its sister volumes constitute the proceedings of the 2nd International Symposium on Neural Networks (ISNN 2005). ISNN 2005 was held in the beautiful mountain city Chongqing by the upper Yangtze River in southwestern China during May 30–June 1, 2005, as a sequel of ISNN 2004 successfully held in Dalian, China. ISNN emerged as a leading conference on neural computation in the region with increasing global recognition and impact. ISNN 2005 received 1425 submissions from authors on five continents (Asia, Europe, North America, South America, and Oceania), 33 countries and regions (Mainland China, Hong Kong, Macao, Taiwan, South Korea, Japan, Singapore, Thailand, India, Nepal, Iran, Qatar, United Arab Emirates, Turkey, Lithuania, Hungary, Poland, Austria, Switzerland, Germany, France, Sweden, Norway, Spain, Portugal, UK, USA, Canada, Venezuela, Brazil, Chile, Australia, and New Zealand). Based on rigorous reviews, 483 high-quality papers were selected by the Program Committee for presentation at ISNN 2005 and publication in the proceedings, with an acceptance rate of less than 34%. In addition to the numerous contributed papers, 10 distinguished scholars were invited to give plenary speeches and tutorials at ISNN 2005.

The papers are organized into many topical sections under 20 coherent categories (theoretical analysis, model design, learning methods, optimization methods, kernel methods, component analysis, pattern analysis, signal processing, image processing, financial analysis, system modeling, control systems, robotic systems, telecommunication networks, incidence detection, fault diagnosis, power systems, biomedical applications, and industrial applications, and other applications) spanning all major facets of neural network research and applications. ISNN 2005 provided an international forum for the participants to disseminate new research findings and discuss the state of the art. It also created a pleasant opportunity for the participants to interact and exchange information on emerging areas and future challenges of neural network research.

Many people made significant efforts to ensure the success of this event. The ISNN 2005 organizers are grateful to Chongqing University, Southwest Normal University, Chongqing University of Posts and Telecommunications, Southwest Agricultural University, and Chongqing Education College for their sponsorship; grateful to the National Natural Science Foundation of China for the financial support; and to the Asia Pacific Neural Network Assembly, the European Neural Network Society, the IEEE Computational Intelligence Society, and the IEEE Circuits and Systems Society for their technical co-sponsorship. The organizers would like to thank the members of the Advisory Committee for their spiritual support, the members of the Program Committee for reviewing the papers, and the members of the Publication Committee for checking the papers. The organizers would particularly like to thank the publisher, Springer, for their cooperation in publishing the proceedings as three volumes of the Lecture Notes in Computer Science series. Last but not least, the organizers would like to thank all the authors for contributing their papers to ISNN 2005. Their enthusiastic contributions and participation were essential parts of the symposium with which the organizers were proud to be involved.

May 2005

Jun Wang
Xiaofeng Liao
Zhang Yi

ISNN 2005 Organization

ISNN 2005 was organized and sponsored by Chongqing University, Southwest Normal University, Chongqing University of Posts and Telecommunications, Southwest Agricultural University, and Chongqing Education College in cooperation with the Chinese University of Hong Kong. It was technically cosponsored by the Asia Pacific Neural Network Assembly, the European Neural Network Society, the IEEE Circuits and Systems Society, and the IEEE Computational Intelligence Society. It was financially supported by the National Natural Science Foundation of China and K.C. Wong Education Foundation of Hong Kong.

General Chair

Jun Wang, Hong Kong, China

Advisory Committee Co-chairs

Shun-ichi Amari, Tokyo, Japan
Jacek M. Zurada, Louisville, USA

Advisory Committee Members

Zheng Bao, X'ian, China
Guoliang Chen, Hefei, China
Ruwei Dai, Beijing, China
Chunbo Feng, Nanjing, China
Walter J. Freeman, Berkeley, USA
Toshio Fukuda, Nagoya, Japan
Kunihiko Fukushima, Tokyo, Japan
Aike Guo, Shanghai, China
Zhenya He, Nanjing, China
Okyay Kaynak, Istanbul, Turkey
Frank L. Lewis, Fort Worth, USA
Yanda Li, Beijing, China
Erkki Oja, Helsinki, Finland
Tzyh-Jong Tarn, St. Louis, USA
Shoujue Wang, Beijing, China
Youshou Wu, Beijing, China
Bo Zhang, Beijing, China
Nanning Zheng, Xi'an, China

Steering Committee Chairs

Xiaohong Li, Chongqing, China
Yixin Zhong, Beijing, China

Steering Committee Members

Wlodzislaw Duch, Torun, Poland
Yinguo Li, Chonqing, China
Max Q.H. Meng, Hong Kong, China
Marios M. Polycarpou, Cincinnati, USA
Yuhui Qiu, Chongqing, China
Zhengqi Sun, Beijing, China
DeLiang Wang, Columbus, USA
Zhongfu Wu, Chongqing, China
Zongben Xu, Xi'an, China
Gary G. Yen, Stillwater, USA
Fuliang Yin, Dalian, China
Juebang Yu, Chengdu, China

Program Committee Co-chairs

Xiaofeng Liao, Chongqing, China
Zhang Yi, Chengdu, China

Program Committee Members

Shigeo Abe, Kobe, Japan
Sabri Arik, Istanbul, Turkey
Amit Bhaya, Rio de Janeiro, Brazil
Abdesselam Bouzerdoum, Wollongong, Australia
Jinde Cao, Nanjing, China
Laiwan Chan, Hong Kong, China
Ke Chen, Manchester, UK
Luonan Chen, Osaka, Japan
Tianping Chen, Shanghai, China
Yen-Wei Chen, Kyoto, Japan
Yiu Ming Cheung, Hong Kong, China
Zheru Chi, Hong Kong, China
Hyungsuk Cho, Dae Jeon, Korea
Andrzej Cichocki, Tokyo, Japan
Shuang Cong, Hefei, China
Chuanyin Dang, Hong Kong, China
Meng Joo Er, Singapore
Mauro Forti, Siena, Italy
Jun Gao, Hefei, China
Chengan Guo, Dalian, China
Ping Guo, Beijing, China
Zengguang Hou, Beijing, China
Baogang Hu, Beijing, China
Dewen Hu, Changsha, China
Jinglu Hu, Fukuoka, Japan
Danchi Jiang, Hobart, Australia
Licheng Jiao, Xi'an, China
Nikola Kasabov, Auckland, New Zealand
Hon Keung Kwan, Windsor, Canada
Irwin King, Hong Kong, China
Cees van Leeuwen, Tokyo, Japan
Xiaoli Li, Birmingham, UK
Yangmin Li, Macau, China
Yuanqing Li, Singapore
Yanchun Liang, Changchun, China
Lizhi Liao, Hong Kong, China
Chin-Teng Lin, Hsingchu, Taiwan
Ju Liu, Jinan, China
Qing Liu, Wuhan, China
Baoliang Lu, Shanghai, China
Hongtao Lu, Shanghai, China
Fa-Long Luo, San Jose, USA
Zhiwei Luo, Nagoya, Japan
Qing Ma, Kyoto, Japan
Satoshi Matsuda, Narashino, Japan
Tetsuo Nishi, Fukuoka, Japan
Stanislaw Osowski, Warsaw, Poland
Paul S. Pang, Auckland, New Zealand
Rudy Setiono, Singapore
Yi Shen, Wuhan, China
Daming Shi, Singapore
Peter Sincak, Kosice, Slovakia
Jianbo Su, Shanghai, China
Changyin Sun, Nanjing, China
Fuchun Sun, Beijing, China
Ron Sun, Troy, USA
Johan Suykens, Leuven, Belgium
Ah Hwee Tan, Singapore
Ying Tan, Hefei, China
Dan Wang, Singapore
Lipo Wang, Singapore
Wanliang Wang, Hangzhou, China
Wei Wu, Dalian, China
Michel Verleysen, Louvain, Belgium
Hong Yan, Hong Kong, China
Mao Ye, Chengdu, China
Wen Yu, Mexico City, Mexico
Zhigang Zeng, Hefei, China
Huaguang Zhang, Shenyang, China
Liming Zhang, Shanghai, China
Liqing Zhang, Shanghai, China
Chunguang Zhou, Changchun, China

Special Sessions Chair

Derong Liu, Chicago, USA

Organizing Chairs

Guoyin Wang, Chongqing, China
Simon X. Yang, Guelph, Canada

Finance Chairs

Guangyuan Liu, Chongqing, China
Qingyu Xiong, Chongqing, China
Yu Wu, Chongqing, China

Publication Co-chairs

Yi Chai, Chongqing, China
Hujun Yin, Manchester, UK
Jianwei Zhang, Hamburg, Germany

Publicity Co-chairs

Min Han, Dalian, China
Fengchun Tian, Chongqing, China

Registration Chairs

Yi Chai, Chongqing, China
Shaojiang Deng, Chongqing, China

Local Arrangements Chairs

Wei Zhang, Chongqing, China
Jianqiao Yu, Chongqing, China

Secretariat and Webmaster

Tao Xiang, Chongqing, China

Table of Contents, Part II

7 Pattern Analysis

8 System Modeling

9 Signal Processing

10 Image Processing

11 Financial Analysis

Table of Contents, Part III

12 Control Systems

13 Robotic Systems

14 Telecommunication Networks

15 Incidence Detection

16 Fault Diagnosis

17 Power Systems

18 Biomedical Applications

19 Industrial Applications

20 Other Applications

Table of Contents, Part I

1 Theoretical Analysis

2 Model Design

3 Learning Methods

4 Optimization Methods

5 Kernel Methods

6 Component Analysis

A New Approach for Classification: Visual Simulation Point of View*

Zongben Xu[1], Deyu Meng[2], and Wenfeng Jing[1]

[1] Institute for Information and System Science, Xi'an Jiaotong University
Xi'an 710049, China
zbxu@mail.xjtu.edu.cn

[2] School of Electronic and Information Engineering Xi'an Jiaotong University
Xi'an 710049, China

Abstract. Classification is a fundamental problem in data mining, which is central to various applications of information technology. The existing approaches for classification have been developed mainly based on exploring the intrinsic structure of dataset itself, less or no emphasis paid on simulating human sensation and perception. Understanding data is, however, highly relevant to how one senses and perceives the data. In this talk we initiate an approach for classification based on simulating the human visual sensation and perception principle. The core idea is to treat a data set as an image, and to mine the knowledge from the data in accordance with the way we observe and perceive the image. The algorithm, visual classification algorithm (VCA), from the proposed approach is formulated. We provide a series of simulations to demonstrate that the proposed algorithm is not only effective but also efficient. In particular, we show that VCA can very often bring a significant reduction of computation effort without loss of prediction capability, as compared with the prevalently adopted SVM approach. The simulations further show that the new approach potentially is very encouraging and useful.

1 Introduction

Data mining (DM) is the main procedure of knowledge discovery from database (KDD), aiming at discovery of useful knowledge from large collections of data. The classification is the fundamental problem in DM aiming at finding a discriminant rule from a set of experiential data with multiple labels generated from an unknown but fixed distribution, and then, according to the rule found, categorizing any new input datum. The problem has attracted extensive attention in past decades due to their very wide-scope applications in scientific research and engineering. Typical examples are, for instance, any applications related to handwritten digit recognition [1], speaker identification [2], and face detection[3].

Various approaches for classification have been developed. Roughly four types of approaches are known and widely applied nowadays, which are the statistical

* This research was supported by the NSFC project under contract 10371097.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 1–7, 2005.

methods(say, Bayes classifier [4]), the discriminant analysis methods (say, linear, nonlinear and Fisher discriminant analysis methods [5]), the iterative methods (say, boosting algorithms [6]) and the computational intelligence methods including, particularly, the fuzzy system, neural network and support vector machines (SVMs) ([7][8]). All these approaches have led to many efficient algorithms. Most of them are efficient, however, only for small or middle sized data sets. The applicability for very large data set is still a problem in application. Other problems very often met include the robustness and the model selection problems: most algorithms are very sensitive to sample neatness and model parameters, and there is no general rule to specify the model parameters involved.

In this paper, we will initiate a new approach for classification, with an expectation of resolving some of the problems mentioned above. The approach is motivated by an observation that the existing approaches have been developed mainly based on exploring the intrinsic structure of data set itself, less or no emphasis paid on human sensation and perception. Understanding data is, however, highly relevant to how one senses and perceives the data. So it should be very helpful to develop an approach based on mimicking human sensation and perception principle. The aim of present research is to make such idea clear and, furthermore, develop a new approach through simulating human visual sensation and perception. Our core idea is to treat a data set as an image, and mine the knowledge from the data in accordance with the way we observe and perceive the image.

The paper is outlined as follows. The visual sensation and perception principle is briefly summarized in section 2, and, then, based on the principle, a new approach for classification problem, called Visual Classification Algorithm (VCA), is developed and a series of simulation results of VCA, in contrast to the prevalently used SVM, are provided in section 3. Some concluding remarks are given in section 4.

2 Visual Sensation and Perception Principle

Human observe an object through sensing light strength of the object that forms an image on the retina. One can perceive objects from different distances because of the fact that the shape of the crystalline lens of eye can be automatically accommodated to focus the image of the objects. There exists an interval within which shape of the lens varies one can observe the object clearly (otherwise, it becomes blurry). Accommodating shape of the lens is functionally equivalent to varying the distance between object and eye (Figure 1).

How an image in retina varies with the distance between object and eye (or equivalently, with the shape of crystalline lens)? The scale space theory developed by Witkin [9] and Koenderink [10] provides us a mathematic answer, which is also supported by neurophysiologic evidences in animals and psychophysical experiments in man directly [11].

2.1 Scale Space Theory

The theory says that if $p(x) : R^n \to R$ is an original image whose gray degree at x is $p(x)$ and $P(x, \sigma)$ is the image appeared in retina at scale σ (that can be

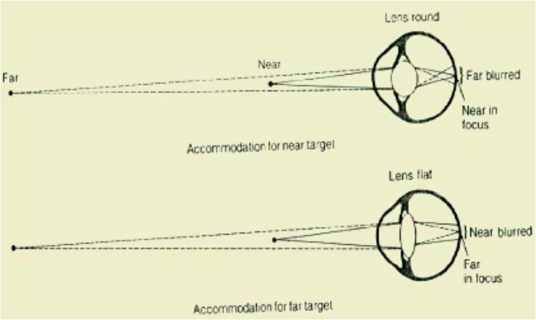


Fig. 1. Accommodation(focusing) of an image in the retina

understood as the distance between object and eye, or the curvature of crystalline lens), then $P(x, \sigma)$ obeys to

$$\left\{ \begin{array}{l} \frac{\partial P}{\partial \sigma} = \Delta_x P \\ P(x, 0) = p(x) \end{array} \right. \tag{1}$$

or, equivalently,

$$P(x, \sigma) = p(x) * g(x, \sigma) = \int_{\Omega} p(x - y) g(y, \sigma) dy \tag{2}$$

where '$*$' denotes the convolution operation, $g(x, \sigma)$ is Gaussian function and σ is the scale parameter. In the theory, (x, σ)-plan is called the scale space and $P(x, \sigma)$ is the scale space image corresponding to scale σ.

The scale space theory provides us a very useful way to explore the structure of an image through analyzing the interconnections among images at different scales. Such an idea has been explored by Leung, Zhang and Xu([12]) for clustering. We will apply the same idea to classification in this presentation. To this end, some further knowledge on category of cell responses in retina is needed.

2.2 Cell Responses in Retina

According to physiological experiments, human can only sense and perceive change of light. In retina, only three types of cell responses exist ([13]): 'ON' response, which is the response to arrival of a light stimulus; 'OFF' response, which is the response to removal of a light stimulus and 'ON-OFF' response, which is the response to the hybrids of 'ON' and 'OFF' (because both presentation and removal of the stimulus may simultaneously exist). For a single small spot of light (a fixed light spot), if it causes 'ON' /'OFF' response on the retina, the cells in the retina with 'ON' /'OFF' response forms a Gaussian-like region (called 'ON' /'OFF' region) and all cells outside of the region then forms an 'OFF'/'ON' region (see Figure 2 for an illustration). Consequently, as for as multiple spots of lights are concerned, different 'ON' , 'OFF' and 'ON-OFF' regions

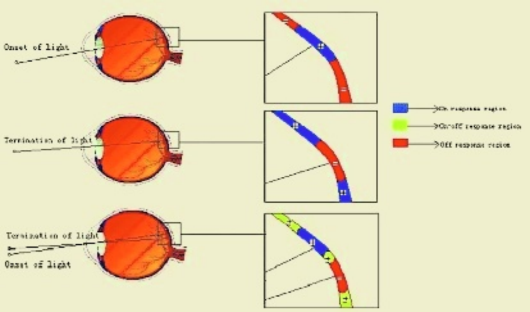


Fig. 2. Three types of cell responses in the retina

may coexist in the retina. Particularly, the ‘ON-OFF’ region intuitively forms narrow boundary between ‘ON’ region and ‘OFF’ region.

Such category information on cell responses will constitute the basis of our to-be-developed VCA for classification problem in the next section.

3 Visual Classification Algorithm

We first explain how a classification problem can be transformed into an image analysis problem and how the visual sensation and perception principle can be applied to analyze the image, providing a reasonable solution of the classification problem. Due to that any multiple-label classification problem can be directly deduced to a series of two-label problems, only is the two-label classification problem considered below.

Thus, let be given an two-label experiential training data set $D = \{x_i^+\}_{i=1}^{N_+} \cup \{x_i^-\}_{i=1}^{N_-}$ that is generated from an unknown but fixed distribution on $\varXi$, where $\varXi$ is a larger data space embodying D, we are asked to determine a discriminant rule on $\varXi$ such that any datum in $\varXi$ can be categorized.

Our main strategies are as follows.

(i) To view every positive (negative) training sample $x_i^+(x_i^-)$ as a spot light with unit strength $\delta(x - x_i)(-\delta(x - x_i))$, causing an ‘ON’(‘OFF’) response in retina (where $\delta(x)$ is a Dirac function). Consequently all the data forms an image

$$p(x) = \frac{1}{N_+ + N_-}(\sum_{i=1}^{N_+} \delta(x - x_i^+) + \sum_{i=1}^{N-}(-\delta(x - x_i^-))) \tag{3}$$

This defines the original image on which the image analysis will be carried and it further bridges the classification problem and the image analysis task.

(ii) To apply the scale space theory to yield a family of blurred images, $P(x, \sigma)$, of $p(x)$. That is, we will define

$$P(x, \sigma) = p(x) * g(x, \sigma), \sigma \geq 0 \tag{4}$$

(iii) For each fixed scale σ_0, view the "+" class as the 'ON' response region, "-" as the 'OFF' region, and the boundary as the 'ON-OFF' region in retina. Correspondingly, the discriminant function is defined by $sgn(P(x, \sigma_0))$ and the classification boundary defined by $\mathit{\Gamma} = \{x : P(x, \sigma_0) = 0\}$.

The strategies result in a family of discriminant functions $\{sgn(P(x, \sigma_0)) : \sigma_0 \geq 0\}$. According to the visual sensation and perception principle, there should be an interval within which the scale σ_0 varies we can always observe the image clearly (that is, the discriminant functions can classify the data properly).The problem is then how to determine such an interval, or, more specifically, a suitable scale σ^* at which the $sgn(P(x, \sigma^*)$ can perfectly classify the date.

To find σ^*, we first observe that when the scale σ gets very small (more precisely, for a fixed classification problem there exits an $\varepsilon > 0$ such that when $\sigma < \varepsilon$), $sgn(P(x, \sigma))$ can always categorize every samples correctly, but no any or has very poor generalization capability; However, when σ gets very large (more precisely, $\sigma > N$ for a large constant N), $P(x, \sigma)$ nearly becomes constant, meaning that it is unable to classify the sample data any more. This observation shows that the expected scale σ^* must be in a bounded interval $[\varepsilon, N]$.

There are still an infinite number of possibilities to choose the scale. If only finite number of scales from which σ^* is selected, the well-known cross-validation approach ([14]) can be applied. So our next step is to select finite number of scales from $[\varepsilon, N]$. This asks for an appropriate discretilization scheme for scales. We propose to adopt the scheme deduced from Weber's law in physiology. The Weber's law says that a person can not recognize the difference between two images whose fraction for line length of the scale parameters is smaller than 0.029 ([13]). By the law, a reasonable discretilization scheme can then be defined as

$$\mathit{\Delta}\sigma = 0.029 \tag{5}$$

With the above discretilization scheme, we thus can obtain a finite number of scales $\{\sigma_i : i = 1, 2, ..., M\}$ where $M = (N - \varepsilon)/\mathit{\Delta}\sigma$, or, correspondingly, a finite number of discriminant functions

$$\{sgn(g(x, \sigma_i))\} : i = 1, 2, ..., M \tag{6}$$

Applying any cross-validation approach (say, 5-fold across method [15]) to $\{\sigma_i : i = 1, 2, ..., M\}$ can then give the expected scale σ^*.

To summarize the above strategies and procedures, the visual classification algorithm (VCA) we propose in this talk can be formally stated as follows.

Visual Classification Algorithm:

Step I (*Scale determination*): Using a cross-validation method combined with the discretization scheme (5)to determine an appropriate scale σ^*;

Step II (*Visual classification*): Compute $P(x, \sigma^*)$ according to the scale space formulae and get the discrimant function $sgn(f(x, \sigma_0)$.

To demonstrate the feasibility and high efficiency of the VCA, 11 groups of IDA benchmark problems (cf. http://ida.first.gmd.de/ raetsch/data/benchmarks.htm) have been used to test in contrast to the SVM with Gaussian kernel

Table 1. The statistics of 11 benchmark problems used in simulations

Problems	Input dim	Size of training set	Size of training set
Banana	2	400	4900
Broast-cancer	9	200	77
Diabotis	8	468	300
Flare-sola	9	666	400
German	20	700	300
Heart	13	170	100
Image	18	1300	1010
Thyroid	5	140	75
Titanic	3	150	2051
Twonorm	20	400	7000
Waveform	21	400	4000

Table 2. Performance comparison between VCA & SVM

Problems	Training time		Prediction error	
	SVM(s)	VCA(s)	SVM(%)	VCA(%)
Banana	4501.41	7.83	11.53±0.66	10.81±0.51
Broast-cancer	773.46	6.63	26.04±4.74	24.82±4.07
Diabotis	7830.93	13.68	23.79±1.80	25.84±1.81
Flare-sola	20419.71	24.23	32.43±1.82	35.01±1.72
German	24397.02	41	23.61±2.07	25.27±2.39
Heart	538.11	4.3	15.95±3.26	17.22±3.51
Image	1346476	129.7	2.96±0.60	3.62±0.63
Thyroid	368.43	3.13	4.80±2.19	4.35±2.34
Titanic	403.59	3.53	22.42±1.02	22.31±1.00
Twonorm	449.84	15.63	2.96±0.23	2.67±0.39
Waveform	4586.25	18.75	9.88±0.44	10.64±0.98
Mean	12825	24.40	16.01±1.71	16.60±1.76

that has been accepted as one of the most promising classification methods nowadays. The dimensions, sizes of training datasets and test datasets related to the problems are listed in Table 1. The performance of VCA applied to the datasets is shown in Table 2 as compared with the SVM. In the simulations, the five-fold cross validation are commonly used to select the scale in VCA and the spread parameter in SVM.

From Table 2 we can see that VCA and SVM both can very successfully classify the data sets and predict new data well. However, as far as the training time is concerned, the VCA outperforms the SVM significantly. Actually, it is observed from the table that without increasing of misclassification rate (i.e., loss of generalization capability), VCA only costs 0.2% times computation effort of SVM, that is, it has brought an approximately 500 times save of computation cost of the SVM. This shows the very high efficiency of the proposed VCA.

4 Concluding Remarks

The existing approaches for classification have mainly been developed based on exploring the intrinsic structure of dataset, less or no emphasis paid on simulating human sensation and perception. We have initiated an approach in this work for classification based on human visual sensation and perception principle. The core idea is to model the blurring effect of lateral retinal interconnections based on the scale space theory and cell response classification facts. The preliminary simulations have demonstrated that the new approach potentially is very encouraging and useful.

The main advantage of the new approach is its high efficiency. It very often brings a significant reduction of computation effort without loss of prediction capability, especially as compared with the prevalently adopted SVM approach.

There are still many problems remains for further research. Some of them, for instance, are: (i) assessing theoretically the approach proposed (say, analyzing the properties like convergence and complexity); (ii) applying the more dedicated nonlinear scale space theory for further efficiency speed-up.

References

1. Burges, C.J.C., Schölkopf, B.: Improving the Accuracy and Speed of Support Vector Learning Machines. In M.Moze, M. Jordan and T. Petsche, editors, Advances in Neural Information Processing Systems. Vol 9. MIT Press (1997) 375-381
2. Schmidt, M.: Identifying Speaker with Support Vector Networks. In Interface '96 Proceedings, Sydney (1996)
3. Yang, M.H., Kriegman, D.J., Ahuja, N.: Detecting Faces in Images: A Survey. IEEE Transactions on Pattern Analysis and Machine Intellegence, **24** (2002) 34-58
4. Rajan, J., Rayner, P.: Time Series Classification Using the Volterra Connectionist Model and Bayes Decision Theory. 1993 IEEE International Conference on Acoustics, Speech, and Signal Processing, **1** (1993) 27-30
5. Fukunaga, K.: Introduction to Statistical Pattern Recognition. Academic Press, 2nd edition, San Diego (1990)
6. Ratsch, G., Mika, S., Scholkopf, B., Muller, K.: Constructing Boosting Algorithms from SVMs: An Application to One-class Classification. IEEE Transactions on Pattern Analysis and Machine Intelligence, **24** (2002) 1184 - 1199
7. Bezdek, C., Pal, K. (eds.): Fuzzy Models for Pattern Recognition. IEEE Press, New York (1992)
8. Simpson, P.: Fuzzy Min-Max Classification with Neural Networks. 1991 IEEE Conference on Neural Networks for Ocean Engineering, (1991) 291 - 300
9. Witkin: Scale-space Filtering Proc. 8th IJCAI. (1983) 1019-1022
10. Koenderink, J. J.: The Structure of Image. Biological Cybernetics, **50** (1984) 363-370
11. Hubel, D.H.: Eye, Brain, and Vision. Scientific Am. Library, New York (1995)
12. Leung, Yee, Zhang, Jiangshe, Xu, Z.B.: Clustering by Scale Space Filtering. IEEE Transactions on Pattern Analysis and Machine Intelligence, **22** (2000) 2-15
13. Coren, S., Ward, L., Enns, J.: Sensation and Perception. Harcourt Brace College Publishers (1994)
14. Acton, S., Bovik, A.: Piecewise and Local Image Models for Regularized Image Restoration Using Cross-validation. IEEE Transactions on Image Processing, **8** (1999) 652 - 665
15. Rasch, G.: Robust Boosting via Convex Optimization. PhD thesis, University of-Potsdam, Neues Palais. vol.10. 14469 Potsdam, Germany. October (2001)

A Novel Classifier with the Immune-Training Based Wavelet Neural Network*

Lei Wang[1], Yinling Nie[1], Weike Nie[2], and Licheng Jiao[2]

[1] School of Computer Science and Engineering, Xi'an University of Technology
Xi'an 710048, China
{leiwang,yinlingnie}@xaut.edu.cn
http://202.200.112.7/computer

[2] School of Electronic Engineering, Xidian University
Xi'an 710071, China
{wknie,lchjiao}@mail.xidian.edu.cn

Abstract. After analyzing the classification and training ability of a wavelet neural network (WNN), a novel WNN learning scheme integrating immunity based evolutionary algorithm (IDEA) is proposed, in which, IDEA is an evolutionary algorithm with an embedded immune mechanism. When WNN is used as a classifier, the process of seeking the least mean-square error (LMS) of an optimal problem is equivalent to that of finding the wavelet feature with maximal separability, namely, maximizing its separable division. On the other hand, with the capability of robust learning of its evolutionary process, IDEA is able to eliminate local degenerative phenomenon due to blindfold behaviors of original operators in the existing evolutionary algorithms. In the case of the twin-spiral problem, experimental simulation shows the feasibility of WNN training with the IDEA based learning algorithm.

1 Introduction

Computer science has a healthy tradition of stealing nature's good ideas [1], as a good example, the model of artificial neural network (ANN) comes from the inspiration of biological neural systems. With ANN's rapid development in theory and wide application in practice, it is gradually realized that the activation function of hidden layer nodes and the learning algorithm for network training, are two main factors affecting network's classification performance and the convenience of its application. For selecting the node function, some researchers look into the wavelet theory, which provides an adjustable time-frequency decomposition of signals [2]. Such kind of node function is able to adaptively extract features from objects, however, its parameter's selection usually depends on designer's experience or experiments for many times. Except this, the process of network training is prone to get stick in local minimum, which makes it not converge. In view of improving the learning robustness, researchers try to use the genetic algorithm (GA) which is a global learning algorithm

* This research is supported by National Science Foundation of China under grant n°60133010.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 8–13, 2005.

with high robustness [3], however, its two main operators not only give every individual evolutionary opportunities but also cause certain degeneracy, and therefore lack universality. Furthermore, GA neglects the assistant function of background characteristics and prior knowledge. For this reason, a global optimization algorithm based on immune strategy is proposed for improving algorithmic efficiency [4].

Based on the considerations of how to select the node function in hidden layer and how to improve the training algorithm, a novel model of integrating wavelet neural network (WNN) with immunity based evolutionary algorithm (IDEA) is proposed, with the aim to simplify the conditions of utilization, realize associational recollection, and improve the classification performance. In which, IDEA comes from the research results of artificial immune system (AIS), a novel direction being paid more and more attention. Most of the inspiration for this research has been drawn from the adaptive immune system (IS) in nature. The adaptive IS is so-called because it adapts or "learns" to recognize specific kinds of pathogens, and retains a "memory" of them for speeding up future responses [5].

2 Wavelet Neural Network

A WNN has many similarities with a universal ANN in structure [6]. Without loss of universality, we suppose the number of neurons in input layer is *n*, that in hidden layer is *N*, that in output layer is *C*; the number of elements in training set is *M*; the dimension of the input signal matrix *S* is of dim$S = n\times M$, that of the output signal matrix *V* is of dim$V = C\times M$, that of the wavelet matrix *H* is of dim$H = N\times n$, and that of the wavelet flexing parameter matrix *A* and the wavelet shifting parameter matrix *B* are respectively of dim$A$ = dim$B = N\times 1$; that of connecting weights matrix *W* in output layer is of dim$W = C\times N$ and that of adjustable bias matrix θ in output layer is of dim$\theta = C\times 1$. For a multi-input and multi-output function f: $R^n \rightarrow R^c$, A WNN can be expressed as:

$$v_i = \sigma(u_i) = \sigma\left[\sum_{k=1}^{N} W_k \sum_{t=1}^{n} s_i(t) h\left(\frac{t-b_k}{a_k}\right)\right]. \tag{1}$$

Suppose the target set is $\{\omega_1, \omega_2, \cdots, \omega_c\}$, and $\overline{\omega}_i$ denotes a target is not in ω_i. For units in output layer, they satisfy V = $\sigma(O)$, where σ is Sigmoid function and *O* is the real output matrix. Suppose *D* is the expected output, and since $\sigma(\cdot)$ is monotone and reversible, there should exist such a relation between *V* and *O*, i.e., dim$O$ = dim$D = C\times N$, and,

$$b_{ik} = \begin{cases} b_1 & S_k \in \omega_i \\ -b_2 & S_k \in \overline{\omega}_i \end{cases},$$

where S_k is the *k*th column vector in the matrix *S*, and b_1, b_2 are both constants. As usual, a cost function of network training is defined as LMS of real outputs, i.e,

$$E = \frac{1}{MC}\sum_{k=1}^{M}\left\|d_{jk} - O_{jk}\right\|^2 = \frac{1}{MC}\left\|D - O\right\|^2, \tag{2}$$

where O_{jk} denotes the *j*th component of the *k*th learning sample, and d_{jk} means the *j*th component of the expected output vector of the *k*th learning sample, and,

$$O = W\hat{H} + \theta I_M^T,$$

where $\hat{H} = HS$ is a matrix of feature vector, I_M is a M-dimension column vector of each component *I*. Through analysis, in view of *θ* and *W*, the process of minimizing *E* in accordance with LMS can be expressed as

$$E_{\min\left|\theta=\hat{\theta},W=\hat{W}\right.} = (b_1 + b_2)^2\left[\frac{1-\sum_{j=1}^{C}P_j^2}{C} - tr(S_T{}^{+}S_B)\right] = (b_1 + b_2)^2\left[\frac{1-\sum_{j=1}^{C}P_j^2}{C} - J\right], \tag{3}$$

where $tr(\cdot)$ means the matrix-tracing operation, $S_T = \frac{1}{M}\sum_{k=1}^{M}(\hat{h}_k - m_{\hat{h}})(\hat{h}_k - m_{\hat{h}})^T$, $S_B = \sum_{j=1}^{C}P_j{}^2(m_h^j - m_{\hat{h}})(m_h^j - m_{\hat{h}})^T$, and m_h^j is a mean vector of wavelet feature from the *j*th learning sample. Here S_T can be also regard as an estimation of a sample mix-spreading matrix defined in the feature space based on the sample set, and S_B as an estimation of the spreading matrix within a sample class defined in the feature space. Therefore, it can be seen that minimizing *E* for wavelet parameters *A* and *B* with minimizing *E* for *W*, is equivalent to maximizing *E* for *A* and *B*, also equivalent to maximizing *J* for *A* and *B*, i.e.

$$\max\text{:}\, J(A,B) = tr[S_T{}^{+}S_B]. \tag{4}$$

From the above optimization problem, we can get $J_{\max} \geq J$ (the equal mark holds if the starting values of *A* and *B* have already been extremums). Then from eqn.2, we get

$$E_{\min} = (b_1 + b_2)^2\left[\frac{1-\sum_{j=1}^{C}P_j{}^2}{C} - J_{\max}\right]. \tag{5}$$

If $J_{\max} = \frac{1}{C}\left(1-\sum_{j=1}^{C}P_j{}^2\right)$, then the separable LMS descends to 0. Therefore, we can conclude the following two points: 1, for a wavelet neural network classifier, the process of finding LMS is equivalent to the process of seeking the wavelet feature with maximal separability and maximizing the separable division of the wavelet feature; and 2, for the feature extracting layer, the aim is to make J=$tr[S_T$+$S_B]$ maximal, that is to make the separable measure $J=J_{\max}$. For the output layer, the aim is to make a division with maximal separability for the output (i.e., the extracted signal wavelet feature) in feature extracting layer.

3 Immune Strategy Based Network Training Algorithm

As written in introduction, the process of network training with *LMS* method is prone to get stick in a local minimum, and therefore makes the learning suboptimal. That is why GA is employed as the network training algorithm. Due to the generated degeneracy of classical operators, GA still needs many improvements, especially the ability of long term memory, diversity of operators and stability of evolution. Based on this consideration, the immunity based Evolutionary Algorithm – IDEA, is proposed in order to improve the whole efficiency of an algorithm.

IDEA is specially designed for dealing with such an optimal problem as:

$$\max\left\{F(x)\middle| x\in \Omega\subset R^n\right\}, \tag{6}$$

where $\Omega\subset R^n$ is a compact set and $F(x)$ is the fitness function in Ω that is continuous and greater than zero. Our purpose is to find a quasi-optimal solution $x^*\in\Omega$ such that the error between x^* and the theoretical optimum satisfies the prescribed precision ($\Delta=10^{-l}$). The algorithm is as follows:

Step 1: Initialization. First, set the precision of solutions in accordance with the demand. Due to the calculating error, we cannot ask the quasi-optimal solution x^* is equal to the theoretical optimum. Here the precision Δ is supposed to be 10^{-l}. Then, create randomly N individuals in Ω. Each component of an individual is a decimal with l bits, and all of these individuals construct the initial population A_0.

Step 2: Differentiation. Calculate the fitness of all individuals in present population A_k. If halt conditions are satisfied, then stop and output the results; or else, continues.

Step 3: Mutation on the present parent A_k and obtain the results B_k. Suppose $A_k=(a^1,\cdots,a^N)$, $a^i\in\Omega\subset R^n$, $a^i=(a_1^i,\cdots,a_n^i)$, i=1, ⋯ , N. The mutation on a^i means to generate the new individual b^i in the following fashion:

$$b_j^i=a_j^i+\xi_j^i, j=1,\cdots,n\,, \tag{7}$$

where ξ_j^i is the greatest decimal with l bits not more than the stochastic variable $\eta_j^i \sim N(0,\sigma_{ij}^2)$, and the variance σ_{ij}^2 is calculated according with the following:

$$\sigma_{ij}=\frac{T_k M_j+m_j}{T_k+f(a^i)}, \tag{8}$$

where T_k is the series as same as that in the annealing selection, and T_0 should be enough great so as to ensure all individuals have more evolutionary chance in the initial stages. M_j and m_j relate to the shape, size and the other factors of Ω, and $m_j\neq 0$.

Step 4: Clonal Selection. At first, updating of the memory cells set. The memory set is composed of *m* individuals (in fact, it is also a special population). In the current population, φ (φ <<μ) individuals with the strongest vitality are selected to replace the φ individuals in the memory set that are the most weak. Here, the word of vitality is a concept similar to the ordinarily said *fitness*. For an example, as to the problem of finding the maximum of a function, the real value of an individual can be just regarded as the vitality of itself. Secondly, take the density-based selection, i.e., compose the current population and the memory set to a temporary population, and then calculate the every individual's fitness as follows [7]:

$$\begin{aligned} Fitness(i) &= vitality(i) + \zeta\,\frac{vitality(i)}{MaxVitality}\cdot vitality(i) \\ &+ \delta(1-Ds)\left(1+\operatorname{sgn}(\theta-Ds)\frac{vitality(i)}{MaxVitality}\right)\cdot vitality(i) \end{aligned} \quad , \tag{10}$$

where δ and ζ are two adjustable parameters in (0,1). θ is a threshold value also in (0,1) that is used for determining if an individual's vitality is strong or not. *MaxVitality* means the maximal vitality value of the current population. sgn(·) is the symbol function and *Ds* suggests the density of the individuals with stronger vitality, i.e.,

$$Ds = \frac{\sum_i \left(\theta\cdot MaxVitality \le vitality(i) \le MaxVitality\right)}{\mu+\varphi}\,. \tag{11}$$

After clonal selection, let *k*=*k*+1 and then return to Step 2.

4 Simulations

As we know, the twin-spiral topic belongs to a non-linear problem of classification, which is demanded to classify exactly the points which belong to two different spiral curves. Usually, the polar coordinates function of a spiral curve is given as:

$$\rho = \alpha\theta + \beta \quad , \tag{12}$$

where, α and β are both constants, and respectively suggest the angular velocity of a spiral curve and the beginning distance to the polar.

4.1 Classification of the Twin-Spiral Problem

As the two elements of any point (*x*, *y*) are same in essence, or equivalent, the wavelet base function can be selected from one dimension wavelet models. Therefore, the following function is considered as the wavelet base function, i.e.,

$$\psi(t) = \cos 1.75t\cdot\exp(-\frac{t^2}{2})\,. \tag{13}$$

In this simulation, 640 points are first generated with random noise which respectively belong to two spiral lines ρ_1 and ρ_2 (320 points per each). In which, the angular velocities of the two spiral lines are same, and are both 4, in addition, the beginning distance are respectively 1 and 7. Half of the sample points are alternately

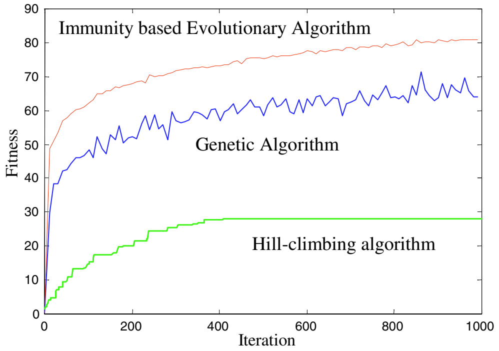


Fig. 1. Comparison among three different training algorithms

selected as the training data, the rest are used for test. When the trained network is used for classifying the testing data, there are about 13~17 points which are made error classification every time, and the correct ratio is 95.3125% (average value). If without noise, then the ratio is near to 100% (also average value).

4.2 Comparison Among the Training Algorithms

For the Wavelet network training, we select another two algorithms for such purpose except for IDEA, one is genetic algorithm, and the other is the hill-climbing algorithm. Under the same conditions, we use the trained WNN network for this aimed problem, and IDEA's results are shown as the above, while GA's correct ratio is about 81.2500% (average value) and the hill-climbing algorithm's is only 57.3725% (average value). Due to different training process, shown as Fig.1, the training with GA is not sufficient, and it seems not finished within 1000 iterations (because of some degradation, its efficiency drops during anaphase), while the hill-climbing process gets into a local minimum status without further ado in the end.

References

1. Klarreich, E.: Inspired by Immunity. Nature, **415** (2002) 468-470
2. Szu, H. H., Kadambe, S.: Neural Network Adaptive Wavelets for Signal Representation and Classification. Optical Engineering, **9** (1992) 1907-1916
3. Rudolph, G.: Convergence Analysis of Canonical Genetic Algorithms. IEEE Trans. on Neural Networks, **5** (1994) 96-101
4. Jiao, L. C., Wang, L.: A Novel Genetic Algorithm Based on Immunity. IEEE Trans. on Systems, Man, And Cybernetics-Part A: Systems and Humans, **30** (2000) 552-561
5. De Castro, L. N., Von Zuben, F. J., de Desus, Jr F. A.: The Construction of a Boolean Competitive Neural Network Using Ideas from Immunology. Neurocomputing, **50** (2003) 51-85
6. Zhang, Y. N.: Researches of Intelligent Target Identification. Report of Post-doctoral Research in Xidian University. Xi'an (1999)
7. Liu, K. S., Cao, X. B., Zheng, H. R., Wang, X. F.: Solving TSP Based on Immune Algorithm. Computer Engineering of China. **26** (2000) 1-16

Fisher Subspace Tree Classifier Based on Neural Networks

Dongyue Chen, Xiaodan Lu, and Liming Zhang

Dept. Electronic Engineering, Fudan University, Shanghai 200433, China
{022021021,lmzhang}@fudan.edu.cn

Abstract. This paper proposes a multi-neural network classification based on fisher transformation. The new method improves HDR[1] (Hierarchical discriminate regression) method proposed in 2000, which can classify the training set from coarse to fine by non-linear dynamic clustering for high-dimension data. In proposed method a fisher subspace replaces K-L subspace of HDR that simplifies the Hierarchical tree. Simulation results show that our method is better than HDR on recognition ratio and time cost.

1 Introduction

As the development of multimedia technology the high-dimension data such as image, video and audio become as the dominant data sources in pattern recognition. So there are lots of challenges for computation complexity and time consuming in pattern recognition and neural network area.

In 2000, a regression method, called HDR for high-dimension data, is proposed by Weng [1]. It has good performance for high-dimension data because HDR deals with the data in K-L subspace by using SDNLL (Size-Depend Negative-log-likelihood) distance which is good for image data. Since HDR has tree-like structure, it is more efficient on the time and space cost in NN classifier. Its search time is a log-function of training samples' number. However, it also has something to improve. In K-L subspace of HDR information of samples' classification is lost, so HDR is not good enough when the training set is small.

Here we uses a two-layer neural network based on fisher subspace to perform the node's split in HDR. This network can realize non-linear clustering using a non-linear output function that is similar to SDNLL. Fisher space uses class-label's information completely, so it is more efficient than HDR.

2 Hierarchical Discriminate Regression (HDR)

HDR is a subspace learning method by tree which use the natural information and class-label at the same time and classify the projection of data on low-dimension subspace. So HDR can deal with the high-dimension data more efficiently.

Given a training set $L=\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}, i=1,2,\ldots n$, $\boldsymbol{x}_i$ is a input vector and $\boldsymbol{y}_i$ is the label vector of $\boldsymbol{x}_i$. The tree building steps is outlines as follows.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 14–19, 2005.

Step1: Clustering of *y*: divide *L* into *k* clusters according to *y*, *L*={ $l_1, l_2 ... l_k$ }, Here n_i is the number of samples in class l_i. *L*={(x_{ij}, y_{ij})},*i=1,2, ...k*, *j=1,2,...*n_i. Fig.1 (left picture) shows the case of k=3. The corresponding *x*-cluster is shown in middle of Fig.1

Step2: Mapping ***x*** to low dimension space: Compute mean vector $\boldsymbol{m}_{ix}$ of ***x*** for every cluster l_i, *i=1,2, ...k* and compute the K-L transform matrix ***W*** by $\boldsymbol{m}_{ix}$ *i=1,2, ...k* ,and project all samples $\boldsymbol{x}_{ij}$, *i=1,2, ...k*, *j=1,2,...*n_i to the K-L subspace, $\boldsymbol{x'}_{ix}=\boldsymbol{W}\boldsymbol{x}_{ix}$, $\boldsymbol{m'}_{ix}=\boldsymbol{W}\boldsymbol{m}_{ix}$, where $\boldsymbol{x}_{ij}\in R^d$ and $\boldsymbol{x'}_{ij}\in R^{k-1}$, *k*-1<<*d*. The covariance $\boldsymbol{C'}_{ix}$ of class l_i in the low subspace is:

$$\mathbf{C}'_{ix}=\frac{1}{n_i}\sum_{j=1}^{n_i}(\mathbf{x}'_{ij}-\mathbf{m}'_{ix})(\mathbf{x}'_{ij}-\mathbf{m}'_{ix})^T. \tag{1}$$

Step3: Clustering of ***x'***: Suppose that $\boldsymbol{x'}=\boldsymbol{x'}_{ij}$, For every sample ($\boldsymbol{x}_{ij}$, $\boldsymbol{y}_{ij}$)∈L assign sample ($\boldsymbol{x'}_{ij}$, $\boldsymbol{y'}_{ij}$) into cluster i^* , where:

$$i^*=\underset{i}{\operatorname{argmin}}(d_{SDNLL}(\mathbf{x}',i)),i=1,2,\cdots,k. \tag{2}$$

$$d_{SDNLL}(\mathbf{x}',i)=\frac{1}{2}(\mathbf{x}'-\mathbf{m}'_{ix})^T\,\Sigma_i^{-1}(\mathbf{x}'-\mathbf{m}'_{ix})+\frac{1}{2}\ln(|\Sigma_\mathbf{i}|)\qquad\cdots\cdots(1). \tag{3}$$

where, $\Sigma_i=w_e\rho_i{}^2\boldsymbol{I}+w_m\boldsymbol{S}_w+w_g\boldsymbol{C'}_{ix}$, $\rho_i=\left\|\boldsymbol{x}'-\boldsymbol{m}'_{ix}\right\|$, $\boldsymbol{S}_w$ is the global covariance of L in subspace. w_e,w_m,w_g are constants constrained by $w_e+w_m+w_g=1$. $\Sigma_i\in R^{k-1\times k-1}$. In the subspace, compute the new mean vector $\boldsymbol{m'}_{ix}$ and covariance $\boldsymbol{C'}_{ix}$ of every updated cluster for *i=1,2,* ···*k* which is as the parameters of the k new sub-nodes. Send all the samples in cluster *i* to the *i*th new sub-node, shown in Fig 1(right picture). Thus this node is split as k sub-nodes.

Step4: Check every new split sub-node. If every sample in this sub-node has the same ***y***, this sub-node is called a leaf-node, stop the division of this sub-node. Otherwise repeat step1 to step 3 to split these sub-nodes until all new sub-nodes are leaf-node. The structure of HDR tree shows in Fig 2.

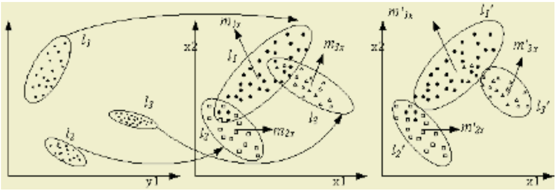


Fig. 1. From left to right, Left shows the ***y*** in L is divided into 3 clusters. Middle is the corresponding ***x*** clusters according to ***y***. Right is the ***x'*** clusters in subspace.

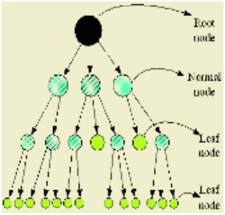


Fig. 2. The structure of HDR tree.

Fig 1 from left to right shows the process of the node split when the samples are 2-d. Note that the dimension reduction is not shown in this Figure for the simple case. The test algorithm of HDR state as follows:

Given a HDR tree T and a tested sample $\boldsymbol{x}$, find the output $\boldsymbol{y}$ with the search width r

Step1: Compute the SDNLL distance between x and every sub-node. Select at most r sub-nodes that have the smallest SDNLL distance, which are called active nodes.

Step2: Save all leaf-nodes in active nodes in memory and calculate the SDNLL distance between x and the remaining active nodes. Select at most top r sub-nodes that have the smallest SDNLL as the new active node.

Step3: Repeat step1 and step2 till the all active nodes are leaf-nodes. Find the sample $\{x^*,y^*\}$that has the nearest Euclidean distance with the tested sample x in all leaf-nodes and output result y^*. Notice the y can be any kind output, such as persons' label in ID recognition and robot's moving requirement in robot navigation.

The search time of HDR is log-function of training set's size due to the tree structure.

3 Fisher Subspace Tree Classifier Based on Neural Networks

Although HDR has great performance, its K-L subspace is obtained only by the mean value of every cluster that may be lost much classification information. This paper proposes a Fisher subspace to replace the K-L subspace and use two-layer neural network perform the node's split in HDR.

Fisher transformation was proposed by R.A.Fisher in 1936. It has many applications in pattern recognition[2]. The idea is find a subspace to increase the distance-between- classes and decrease the distance-in-class of the sample set in this subspace. A given training set $L=\{(\boldsymbol{x_i}, \boldsymbol{y_i})\},i=1,2,...N$, is divided into k clusters, $\{ l_1,l_2...l_k \}$, l_i have n_i samples, $L=\{(\boldsymbol{x_{ij}}, \boldsymbol{y_{ij}})\},i=1,2, \cdots k, j=1,2,\cdots n_i$. $N = n_1 + n_2 + \cdots + n_k$. Let $\boldsymbol{M}$ be the mean vector of all samples in set L. The mean vector and covariance matrix of $\boldsymbol{x}$ in class l_i are denoted by $\boldsymbol{m_i}$ and $\boldsymbol{C_i}$ respectively. Then the Fisher transform matrix is defined as $\boldsymbol{F}=[\boldsymbol{F_1}, \boldsymbol{F_2}, \cdots, \boldsymbol{F_{k-1}}]$, $\boldsymbol{F_i}$ is the ith eigenvector of matrix J=$(\boldsymbol{S_w})^{-1}\times \boldsymbol{S_b}$. Where:

$$\boldsymbol{S}_w = \frac{1}{N}\sum_{i=1}^{k}\sum_{j=1}^{n_i} n_i(\boldsymbol{x}_{ij}-\boldsymbol{m}_i)(\boldsymbol{x}_{ij}-\boldsymbol{m}_i)^T \cdots (5) \tag{4}$$

$$\boldsymbol{S}_b = \frac{1}{N}\sum_{i=1}^{k} n_i(\boldsymbol{m}_i-\boldsymbol{M})(\boldsymbol{m}_i-\boldsymbol{M})^T \cdots (6) \tag{5}$$

In this paper, we proposed a two-layer neural network based on Fisher subspace to realize the nodes split in HDR. This two-layer neural network is shown in Fig.3.

In Fig.3, C_{in}^i, $i=1,2, \cdots,d$, represents the ith input neuron. C_{mid}^j ,$j=1,2, \cdots,k$-1 is the jth middle-layer neuron. Weight matrix $\boldsymbol{W}=[\boldsymbol{w_1}, \boldsymbol{w_2}, \cdots,\boldsymbol{w_{k-1}}]$ corresponds to Fisher transform matrix $\boldsymbol{F}$. C_{out}^i, $i=1,2, \cdots,k$ represent the kth clusters in the sample set L.

$\boldsymbol{M}=[\boldsymbol{m_1}, \boldsymbol{m_2}, ...,\boldsymbol{m_{k-1}}]$, $\boldsymbol{m_i}$ correspond to the mean vector of cluster i.

Similar to the node's split in HDR, clustering samples in set L into k clusters based on $\boldsymbol{y}$ and calculate the mean vector $\boldsymbol{m}_{ix}$ and covariance matrix $\boldsymbol{C}_{ix}$ of $\boldsymbol{x}$ for cluster i,

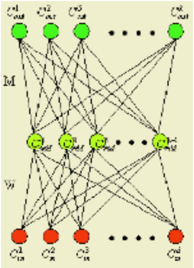


Fig. 3. Two-layer neural network classification

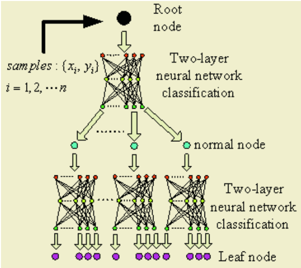

Root node
samples : $\{x_i, y_i\}$
$i = 1, 2, \cdots n$
Two-layer neural network classification
normal node
Two-layer neural network classification
Leaf node

Fig. 4. Modified HDR tree structure

i=1,2,…k. Use the Eq.(4),(5) to get the Fisher transform matrix $\boldsymbol{F}\in \mathrm{R}^{d\times (k-1)}$. Initialize the weights $\boldsymbol{W}$ with $\boldsymbol{F}$, $\boldsymbol{w}_i=\boldsymbol{F}_i$; initialize the weights $\boldsymbol{M}$ with $\boldsymbol{m}_{ix}$, $\boldsymbol{m}_i=\boldsymbol{F}^T\times\boldsymbol{m}_{ix}$, $\boldsymbol{m}_i\in\mathrm{R}^{k-1}$ $i=1,2, \cdots,k$. Several variables such as S_w, $\boldsymbol{C'}_{ix}$ in SDNLL are saved as the inside parameters of the output neuron C_{out}^i. When the input signal is $\boldsymbol{x}$, the output of C_{out}^i is:

$$f_i(x) = -\frac{1}{2}(\boldsymbol{W}\boldsymbol{x}-\boldsymbol{m}_i)^T \textstyle\sum_i^{-1}(\boldsymbol{W}\boldsymbol{x}-\boldsymbol{m}_i) - \frac{1}{2}\ln\left(\left|\sum_i^{-1}\right|\right), i=1,2,\cdots,k \,. \qquad (6)$$

Where, $\sum_i$ was defined in section 2. Input all the samples in training set L into the neural network of root node one by one. Every sample will be assigned into the output neuron with biggest value. Every output neuron connects a sub-node. Use the new mean vector and covariance of $\boldsymbol{x}$ in cluster *i* to update $\boldsymbol{m}_i$ and all parameters in C_{out}^i. The weight matrix $\boldsymbol{M}$ between middle-layer and output layer in update period is use unsupervised learning proposed by Kohonen in [3]. Construct a new two-layer neural network to perform the node split for each non-leaf sub-nodes as shown in Fig. 4. Tree-search procedure in our method is similar to HDR: Given a tree *T* and a sample $\boldsymbol{x}$, find the output $\boldsymbol{y}$ with the parallel search width *r* like classical HDR, but the processing between nodes and their sub-nodes is replaced by neural networks .

4 Experiments and Results

Three experiments on different databases are to test our proposed method and to compare with classical HDR.

Experiment (1): Pose identification for face set of Germany male. There are 100 persons with 7 different poses for each as database. One of them is shown in Fig. 5. We select some persons as the training set and remaining people as test set. The results are shown in Table1. We can see that the two methods spend almost the same time in tree-building and search, but our method is better than HDR obviously in error rate.

Table 1. Pose regression experiment results Using Germany male face database

No of training vs. test	Error rate		Training time		Test time	
	HDR	Mine	HDR	Mine	HDR	Mine
560 vs. 140	1/140	0/140	1.943s	1.810s	1.600ms	1.100ms
490 vs. 210	1/210	0/210	1.623s	1.611s	1.400ms	1.200ms
420 vs. 280	0/280	0/280	1.312s	1.242s	1.200ms	0.989ms
350 vs. 350	0/350	0/350	1.001s	0.980s	1.100ms	0.991ms
280 vs. 420	4/420	0/420	0.741s	0.741s	1.200ms	0.920ms
210 vs. 490	5/490	0/490	0.530s	0.561s	1.100ms	0.849ms
140 vs. 560	18/560	5/560	0.271s	0.401s	0.940ms	0.831ms

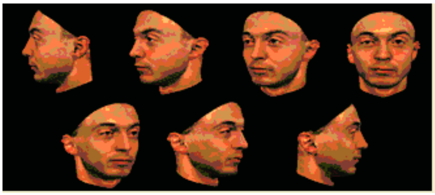


Fig. 5. Germany male face database

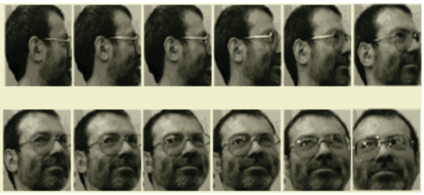


Fig. 6. UMIST face database

Table 2. ID check experiment results using UMIST face data base

No of training vs. test		525 vs.50	475 vs. 100	425 vs. 150	375 vs. 200
Error rate: Mine vs. HDR		0.0% vs. 3.2%	1.2% vs. 6.0%	0.4% vs. 7.2%	1.3% vs.7.2%
Time cost of Training/test	HDR	3.746s/2.2ms	3.415s/2.1ms	3.004s/2.1ms	2.815s/2.2ms
	Mine	3.766s/2.0ms	3.532s/2.0ms	3.140s/1.9ms	3.112s/2.1ms

Experiment (2): ID check using UMIST database. There are 575 pictures for 20 persons in UMIST face database as seen in figure 6. We randomly select some pictures as the test set and building the Tree by the others. The experiment results comparing with different training set and test set are shown in Table 2, which also shows that the improved method is better than HDR with K-L subspace. Two curves of comparison between improved method and HDR is shown in figure 7 in the case of 425 training pictures and 150 testing pictures. The Tree's inputs are these features of face pictures after PCA. From figure 7 we see that the proposed method has higher recognition ratio than HDR whatever the number of principle components is.

Experiment (3): Robot with video camera moving. The collected pictures are clustered into three classes: two different persons stand in front of the robot and no person. According to the scenes in front of the robot and use them training the tree (HDR and our method) for robot moving. When robot sees a scene he can search the tree and find his correct moving direction: no person, he go forward, others are stop or go back. We test our method on the robot (Figure 8). The robot can run normally and verdict whether there are someone in front of him. The simulation results of the video obtained from robot are shown as Table 3.

Table 3. The results of Mobile robot visual navigation

No of training vs. test		1061 vs. 200	961 vs. 300	861 vs. 400	761 vs. 500
Error rate: Mine vs. HDR		2.62% vs. 4.88%	2.50% vs. 3.25%	3.19% vs. 4.50%	3.50% vs. 5.20%
Time cost of Training/test	HDR	5.741s/1.450ms	5.537s/1.500ms	4.977s/1.600ms	5.037s/1.700ms
	Mine	5.092s/1.750ms	5.077s/1.667ms	4.813s/1.500ms	4.873s/1.800ms

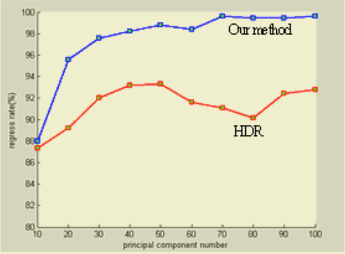


Fig. 7. Comparing our method with HDR in different PC features

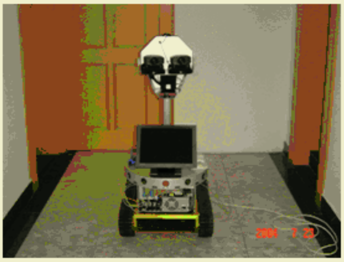


Fig. 8. Intelligent Robot

References

1. Hwang, W., Weng, J..: Hierarchical Discriminant Regression. IEEE Trans. on Pattern Analysis and Machine Intelligence, **22** (2000)
2. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection, **19** (1997)
3. Yin, H., Nigel, M., Allisnson: On the Distribution and Convergence of Feature Space in Self-Organizing Maps. Neural Computation, Massachusetts Institute of Technology, **7** (1995) 1178-1187

Classification Algorithms Based on Fisher Discriminant and Perceptron Neural Network

Hu Yang and Jianwen Xu

College of Science, Chongqing University 400030, China
yh@cqu.edu.cn

Abstract. In this paper, we exploit a new method of implementing mining classification, i.e., Fisher classification algorithm. In comparison with the decision-tree ID3 algorithm and its improved algorithm that is based on the criterion of choosing the split attributes according to information gain ratios and simple Bayes classification algorithm, we find that Fisher classification algorithm has a higher predictive accuracy and relatively less computation effort. Due to the sensitiveness of these methods mentioned above to noise, we propose a perceptron neural network classification algorithm, which has the stronger noise-rejection ability.

1 Introduction

Mining the classification characters is an important research field in data mining [1], which is now being widely used in commerce. The goal of classification is to learn a classification function or model, which is also often called a classifier [2]. There are a lot of methods for constructing classifiers, such as statistical methods, machine learning methods and neural network methods.

As a classifier, decision tree is a directed tree without rings. However, it has proved to be an NP problem to find the best decision trees with the fewest leaf nodes and the smallest node depth [3]. So what we can do is to find relatively better solutions with various heuristic algorithms. In the common ID3 classification algorithm, it defines information gains [2] of training data's attributes with the concept of entropy. We should firstly compute all attributes' information gains while constructing decision tree, and choose the attribute with the biggest information gain as a tree node, then set up decision tree branches according to its various values; Secondly, based on training data contained in different branches, we recursively use the method above to set up decision tree nodes and branches, until all the subsets' training data belong to the same category or there are no attribute left for partition.

The entropy function of ID3 algorithm is not very ideal, having the defect of leaning to attributes with more values. To solve this problem, Quinlan proposed an improved algorithm [4] of ID3, which is based on the criterion of choosing the split attributes according to information gain ratios.

Bayes classification is a method based on Bayes theorem and the maximal posterior probability criterion, and its particular feature is that we should take the prior probability into consideration. For each instance, we should firstly compute its posterior probabilities under all categories, and then determine its category according to the maximal posterior probability criterion.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 20–25, 2005.

2 Fisher Classification Algorithm

Like Bayes algorithm, Fisher classification algorithm is also based on existing statistical methods [5]. Its basic idea is to reduce dimensions, namely substituting all initial variables with fewer discriminants, and then determine the categorie of each instance with these discriminants. In Fisher classification algorithm, we should firstly deduce discriminant functions from the known training data, the general form of which is: $Y=a_1X_1+a_2X_2\cdots a_nX_n$, while Y denotes discriminant values, X_i denote attribute variables and a_i is the coefficient of each variable. Secondly, we need to back substitute the values of each independent variable into discriminant functions and determine each instance's category. Furthermore, we could get the predictive accuracy by comparing the results obtained from Fisher classification algorithm with the true results.

Specifically, let x_{ij} be p-dimensional data in category C_i, $y_{ij}=a'x_{ij}$, while $a=(a_1,a_2,\cdots a_p)'$. In fact, if given a, then y_{ij} is just the discriminant value. Let $\bar{y}_i$ and $\bar{y}$ be the mean values of y_{ij} in category C_i and in all categories respectively, namely $\bar{y}_i=\frac{1}{n_i}\sum_{j=1}^{n_i}y_{ij}=a'\bar{x}_i$, $\bar{y}=\frac{1}{n}\sum_{i=1}^{k}\sum_{j=1}^{n_i}y_{ij}=a'\bar{x}$, while $n=\sum_{i=1}^{k}n_i$, $\bar{x}_i=\frac{1}{n_i}\sum_{j=1}^{n_i}x_{ij}$, $\bar{x}=\frac{1}{n}\sum_{i=1}^{k}n_i\bar{x}_i$. We use the variance analysis technique to seek some transform a, which can separate various categories most. The quadratic sum matrix between categories of y_{ij} is: $\sum_{i=1}^{k}n_i(\bar{y}_i-\bar{y})^2=\sum_{i=1}^{k}n_i(a'\bar{x}_i-a'\bar{x})^2=a'Ba$, with k-1 degrees of freedom; The quadratic sum matrix in categories of y_{ij} is $\sum_{i=1}^{k}\sum_{j=1}^{n}(y_{ij}-\bar{y}_i)^2=\sum_{i=1}^{k}\sum_{j=1}^{n}(a'x_{ij}-a'\bar{x}_i)^2$, with n-k degrees of freedom, while $B=\sum_{i=1}^{k}n_i(\bar{x}_i-\bar{x})(\bar{x}_i-\bar{x})'$, $E=\sum_{i=1}^{k}(x_{ij}-\bar{x}_i)(x_{ij}-\bar{x}_i)'$. So the test statistic of single variance analysis is: $F=\frac{a'Ba/(k-1)}{a'Ea/(n-k)}$. Let $\Delta(a)=\frac{a'Ba}{a'Ea}$, then the bigger $\Delta(a)$ is, the bigger the differences between categories are, so we should choose the transform a, which makes $\Delta(a)$ reach its biggest value. It has proved that the biggest value of $\Delta(a)$ is just the biggest eigenvalue of $E^{-1}B$. Let it be λ_1, and its corresponding standardized eigenvector is just the transform we want to seek [6]. Thus we get the Fisher discriminant $y_1=v_1'x$. We could establish several discriminants similarly when it has numerous categories, namely $y_i=v_i'x, i=1,2,\cdots p$, while v_i is the corresponding standardized eigenvector of λ_i, which are the positive eigenvalues of $E^{-1}B$. After getting the discriminants, we use the Fisher discriminant criterion to determine instances' categories, which is: if $\sum_{j=1}^{r}(y_j-\bar{y}_{lj})^2=\min_{1\le l\le k}\sum_{j=1}^{r}(y_j-\bar{y}_{lj})^2$, then $x\in C_1$, while $\bar{y}_{ij}=v_j'\bar{x}_i$, and r is the number of discriminants.

3 Examples and Comparison Among Classification Algorithms

The problem is that the managers of some golf course want to determine whether it's suitable for playing golf according to the weather conditions. Based on training data (see Ref. [2]), now we construct classifiers with ID3 algorithm, its improved algorithm, simple Bayes classification algorithm and Fisher classification algorithm respectively, and then make comparison between these methods.

3.1 ID3 Classification Algorithm

In this problem, based on the computing results of information gains of attributes, we should choose attribute Outlook as the first split attribute to partition the initial training data. Then choose attribute Humidity and Temperature as split attributes. Finally we could choose attribute Windy and obtain the following final decision tree model (See Figure 1. Ellipses show inner nodes and rectangles show leaf nodes).

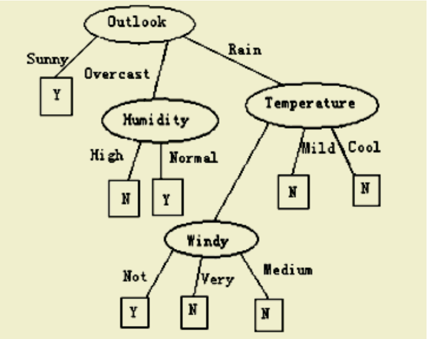


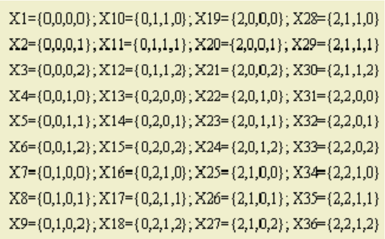
Fig. 1. The final decision tree model of ID3 algorithm

From the final decision tree model, we can extract some classification rules, such as: If Outlook=Sunny, Then category=Yes. It's apparent that the attribute Outlook is an important index. In this problem, we can also obtain the model shown in Figure 1 with the improved algorithm of ID3 based on the criterion of choosing split attributes according to information gain ratios.

3.2 Simple Bayes Classification Algorithm

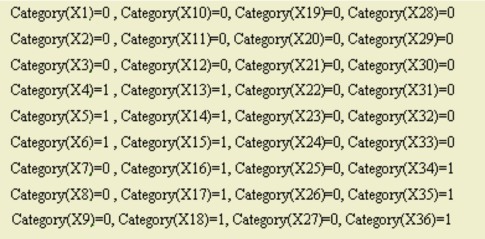
Now we use the simple Bayes classification algorithm to solve the golf problem. Its knowledge can be expressed by some discriminant function *Category*(*X*), the definition domain of which are all possible combinations of values of non-category attributes, the range of which are composed of all the values of the category attribute. For convenience, we use A_1, A_2, A_3, A_4 and C to denote the attribute Outlook, Temperature, Humidity, Windy and Category respectively. The value Overcast, Sunny, Rain of A_1 are denoted by number 0, 1, 2, the values Hot, Mild, Cool by 0, 1, 2, the values High, Normal of A_3 by 0, 1, the values Not, Medium, Very by 0, 1, 2, the values No, Yes by 0, 1. C_1, C_2 denotes that C takes the value 1 and 0 respectively.

For any attribute X, if A_1=1, then it's easy to get that: $P(A_1=1|C_2)=0$, so $P(X|C_1)P(C_1)> P(X|C_2)P(C_2)$ whatever values the other attributes take, and we can get *Category(X)=1* according to the maximal posterior probability criterion. Furthermore, A_1, A_2, A_3, A_4 can take 2, 3, 2, 3 different other values, so it still exists 36 attribute combinations (see Figure 2).



```
X1={0,0,0,0};X10={0,1,1,0};X19={2,0,0,0}; X28={2,1,1,0}
X2={0,0,0,1};X11={0,1,1,1};X20={2,0,0,1}; X29={2,1,1,1}
X3={0,0,0,2};X12={0,1,1,2};X21={2,0,0,2}; X30={2,1,1,2}
X4={0,0,1,0};X13={0,2,0,0};X22={2,0,1,0}; X31={2,2,0,0}
X5={0,0,1,1};X14={0,2,0,1};X23={2,0,1,1}; X32={2,2,0,1}
X6={0,0,1,2};X15={0,2,0,2};X24={2,0,1,2}; X33={2,2,0,2}
X7={0,1,0,0};X16={0,2,1,0};X25={2,1,0,0}; X34={2,2,1,0}
X8={0,1,0,1};X17={0,2,1,1};X26={2,1,0,1}; X35={2,2,1,1}
X9={0,1,0,2};X18={0,2,1,2};X27={2,1,0,2}; X36={2,2,1,2}
```

Fig. 2. Attribute combinations



```
Category(X1)=0 , Category(X10)=0, Category(X19)=0, Category(X28)=0
Category(X2)=0 , Category(X11)=0, Category(X20)=0, Category(X29)=0
Category(X3)=0 , Category(X12)=0, Category(X21)=0, Category(X30)=0
Category(X4)=1 , Category(X13)=1, Category(X22)=0, Category(X31)=0
Category(X5)=1 , Category(X14)=1, Category(X23)=0, Category(X32)=0
Category(X6)=1 , Category(X15)=1, Category(X24)=0, Category(X33)=0
Category(X7)=0 , Category(X16)=1, Category(X25)=0, Category(X34)=1
Category(X8)=0 , Category(X17)=1, Category(X26)=0, Category(X35)=1
Category(X9)=0, Category(X18)=1, Category(X27)=0, Category(X36)=1
```

Fig. 3. The classification results of simple Bayes algorithm

For each combination in Figure 2, we should compute and compare its posterior probability under C_1, C_2, and then determine the value of its category. The final classification results are shown in Figure 3. Until now, the discriminant function *Category*(X) has been determined completely, and we can determine any instance's category with it, and then compare with the true results, we can get the accuracy of simple Bayes classification algorithm, which reaches 87.5% in the golf problem.

3.3 Fisher Classification Algorithm

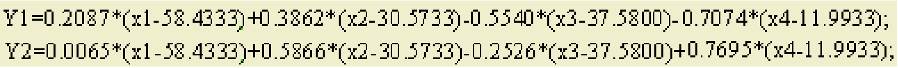
Now we use Fisher classification algorithm to solve the golf problem. For convenience, we quantify the value Sunny, Overcast, Rain with 1, 2, 3, the value Hot, Mild, Cool with 1, 2, 3, the value Normal, High with 1, 2, the value Not, Medium, Very with 1, 2, 3, the value Yes, No with 1, 2, and then deduce linear discriminant functions from the initial variables A_1, A_2, A_3, A_4 with Fisher classification algorithm. The discriminant coefficients are shown in Table 1.

From Table 1, we can get the Fisher discriminant function, which is:

$$Y=0.7283*\text{Outlook}-0.0041*\text{Temperature}+0.6835*\text{Humidity}+0.0488*\text{Windy}.$$

Finally, we get the results of Fisher classification algorithm (see Table 2, True shows the real categories, and Fisher shows the Fisher classification results). We can see that Fisher classification algorithm has a high accuracy, which reaches 95.8%, which is higher than that of the simple Bayes classification algorithm.

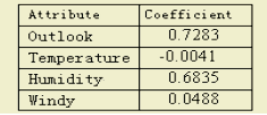
Now we use the famous Iris dataset [6] published by Fisher to further prove the effectiveness of our proposed method. Let x1, x2, x3, x4 be the calycine length, calycine width, petal length and petal width respectively. Then we can obtain the following two discriminant functions with Fisher classification algorithm:



```
Y1=0.2087*(x1-58.4333)+0.3862*(x2-30.5733)-0.5540*(x3-37.5800)-0.7074*(x4-11.9933);
Y2=0.0065*(x1-58.4333)+0.5866*(x2-30.5733)-0.2526*(x3-37.5800)+0.7695*(x4-11.9933);
```

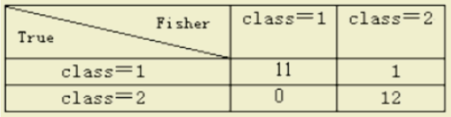
Finally, we can get the Fisher classification results shown in Table 3.

Table 1. Fisher discriminant coefficients



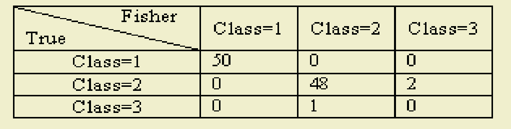
Attribute	Coefficient
Outlook	0.7283
Temperature	-0.0041
Humidity	0.6835
Windy	0.0488

Table 2. The results of Fisher classification algoritm



True \ Fisher	class=1	class=2
class=1	11	1
class=2	0	12

Table 3. The Fisher classification results of Iris dataset



True \ Fisher	Class=1	Class=2	Class=3
Class=1	50	0	0
Class=2	0	48	2
Class=3	0	1	0

From Table 3, it's obvious that the Fisher classification algorithm we proposed is effective and has relatively high predictive accuracy.

4 Perceptron Neural Network Classification Algorithm

In decision tree algorithm, we can only do the classification with one attribute at a time. Simple Bayes algorithm requires the attributes to be independent. All these limits make the algorithms above be weak in the noise proof ability. Neural network obtains knowledge through learning training data repeatedly with the features of intuition, robustness and noise proof, so it has been widely used in many fields.

Forward neural network is a common network configuration in the whole system of neural network. Now, perceptron is still a very useful kind of neural network model, especially for simple classification problems. Perceptron network is composed of input layer, hidden layer and output layer. When the network consists of only one hidden layer with S neurons, and it has R inputs $p(j)$, each of which is connected with the neurons through a set of weights $w(1,j)$. Let the bias of neuron i be $b(i)$, then the hidden layer has S outputs, and the output of neuron i is: $a(i)=f[n(i)]$, while $n(i)=\sum_{j=1}^{R} w(i,j)\times p(j)+b(i),\ i=1,2,\cdots,S$. Perceptron uses the symbolic threshold function, that is: if $n(i)<0$, then $a(i)=0$; else $a(i)=1$. The learning rules of perceptron neural networks are as follows: Considering a perceptron with given input vector P, output vector a and target vector t, we define $e=t-a$ to be its error, then the modified formulas of its weights and biases are as follows: $\Delta w(i,j)=e(i)p(j), \Delta b(i)=e(i)$, so the refreshed weights and biases are: $w(i,j)=w(i,j)+\Delta w(i,j),\ \ b(i)=b(i)+\Delta b(i)$. Neural networks should be trained before they are put into use, and then determine the weights and biases of network. The training processes of perceptron are as follows: we should firstly compute the error e and adjust the weights and biases; then recalculate the inputs under the function of new weights and repeat the process of adjusting the weights, until the outputs of the network equals the target values or the training time reaches the maximal value set previously.

Let's see an example. Some weather operator wants to predict whether the third day would rain through observing the humidity differences x_1, the pressure and tem-

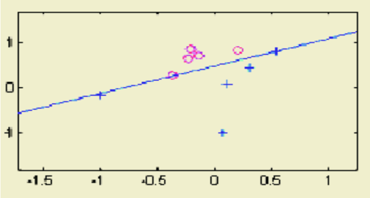


Fig. 4. Trainign data and classification results

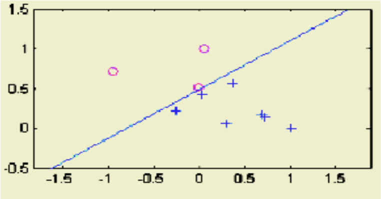


Fig. 5. Classification results of trained network

perature difference x_2 in two continuous days before. The collected data for training are as follow (see Ref. [6], 1 denotes rain and 0 the contrary).

Now we choose the first ten data as training data to train networks, and the latter ten to test the trained perceptron neural network. We should first let each digit divide the training data's maximal value to condense them to be in the interval -1~1.The training data and trained results are shown in Figure 4. Finally, we employ the latter ten data to test the trained network and get the classification results shown in Figure 5. From Figure 5, we can see that the trained perceptron network is effective, and its accuracy reaches 80%.

5 Conclusions

From the results of example analysis, we find that Fisher classification algorithm, decision tree algorithm and simple Bayes algorithm are both effective and have strong consistency in classification. Compared to simple Bayes algorithm, Fisher classification algorithm has a relatively higher predictive accuracy. Perceptron neural network gets the correspondence of its input data to the true categories by adjusting the weights. The results of example analysis show that it's effective for simple classification problems. The merits of neural network lies in its ability to classify untrained data and its strong noise-rejection ability, which makes it play a more and more important role in data mining [1].

References

1. Han, J. kamber M.: Data Mining. Morgan Kaufmann Publishers Inc (2001)
2. Shi, Z.Z.: Knowledge Discovery. University of Tsinghua Press, Beijing (2002)
3. Liu, X.H.: Optmizational Algorithm of Decision Tree. Software Transaction, **9** (1998) 797-800
4. Quinlan, R.: Programs for Machine Learning. Morgan Karfmann, San Mateo (1993)
5. Yang, H., Liu, Q.S., Zhong, B.: Mathematical Statistics. High Education Press of China, Beijing (2004)
6. Wang, X.M.: Applied Multi-analysis. University of Shanghai Finance and Economics Press, Shanghai (2004)

Robust Classification of Immunity Clonal Synergetic Network Inspired by Fuzzy Integral

Xiuli Ma, Shuang Wang, and Licheng Jiao

Institute of Intelligent Information Processing, Xidian University
Xi'an, Shannxi 710071, China
xlma@mail.xidian.edu.cn

Abstract. The selection of prototype pattern vectors and the reconstruction of order parameters are choke points in Synergetic Neural Network (SNN). We have improved the performance of SNN on these two points by Immunity Clonal Strategy (ICS) and applied them to classification successfully. But how to aggregate them remains an open question. Inspired by decision fusion mechanism, a new Immunity Clonal Synergetic Network is proposed to solve the two problems simultaneously. With the use of fuzzy integral, not only are the classification results combined but that the relative importance of the different networks is also considered. Experiments show that the presented algorithm has higher recognition rate and is more robust to classification.

1 Introduction

In the late 1980s, Haken proposed to put synergetic theory into the area of pattern recognition and computer science [1] and then presented a new theory on neural networks, namely, Synergetic Neural Network (SNN). Compared with other traditional neural networks, SNN is constructed from up to down and its remarkable characteristic is not having pseudo-state. In recent years, its learning algorithms and the reconstruction methods of order parameters are being widely studied.

On the selection of prototype pattern vectors, SCAP [2], C-Means clustering, LAIS [3], genetic algorithm (GA) and Immunity Clonal Strategy [4] are applied to get prototype pattern vectors. These learning algorithms improve the performance of SNN to a certain extent but order parameters got by these methods are unreasonable at some degree. Hu discovered this and proposed the reconstruction of order parameters [5] in concept. Then many reconstruction algorithms are presented on award-penalty learning mechanism [3], GA and Immunity Clonal Strategy [6] etc. These methods increase the searching efficiency of SNN. However, SCAP is used to get prototype pattern vectors, which affects the performance of SNN.

Considering that, how to give attention to the two aspects at the same time remains an open question. In this paper, a new Immunity Clonal Synergetic Network inspired by decision fusion is proposed which aggregates the two aspects of SNN by fuzzy integral. The advantage of fuzzy integral is that not only are the classification results combined but that the relative importance of the different networks is also considered.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 26–31, 2005.

Fuzzy integral is also superior to other voting techniques such as Majority and Borda Count [7]. Experiments show the new algorithm can improve recognition rate greatly.

2 Immunity Clonal Synergetic Network Inspired by Fuzzy Integral

2.1 Synergetic Neural Network

Haken's synergetic approach [1] to pattern recognition is based on a parallel between pattern recognition and pattern formation. Supposing that the M prototype pattern vectors v_k and the status vectors q are N-dimensional column vectors with zero mean and unit length, M is less than N for the sake of linear independence of prototype pattern vectors. A dynamic equation [1] can be given as follows:

$$\dot{q} = \sum_{k=1}^{M} \lambda_k (v_k^+ q) v_k - B \sum_{k \neq k'} v_k (v_{k'}^+ q)^2 (v_k^+ q) - Cq(q^+ q) + F(t) \cdot \tag{1}$$

where λ_k is attention parameters and v_k^+ is the adjoint vectors of v_k that satisfies: $(v_k^+, v_{k'}) = v_k^+ v_{k'} = \delta_{kk'}$. Order parameters are defined as

$$\xi_k = (v_k^+, q) = v_k^+ q \,. \tag{2}$$

Corresponding dynamic equation of order parameters is

$$\dot{\xi} = \lambda_k \xi_k - B \sum_{k' \neq k} \xi_{k'}^2 \xi_k - C\Big(\sum_{k'=1}^{M} \xi_{k'}^2\Big)\xi_k \cdot \tag{3}$$

The strongest order parameter will win and desired pattern will be found out.

2.2 Immunity Clonal Synergetic Neural Networks

The selection of prototype pattern vectors and the reconstruction of order parameters are choke points in SNN. We have improved SNN's performance on these two points in paper [4] and [6]. But the two methods have their own disadvantages.

In SNN, the learning problem can be reduced to how to get prototype pattern vectors and adjoint vectors. We can get prototype pattern vectors first and then compute adjoint vectors and vice versa. In paper [4], a new learning algorithm based on Immunity Clonal Strategy (ICS) is presented, which has automatic balance ability between exploration and exploitation and is not easy to get into local optimum. Experiments on Iris data and remote sensing images show that the new method has not only a faster convergent speed but also a higher classification accuracy rate. But the obtained order parameters are unreasonable at some degree.

In paper [6], a novel reconstruction algorithm of order parameters based on ICS is presented, which combines the self-learning ability of SNN with the global searching performance of ICS to construct linear transform and then realize reconstruction. The tests on IRIS data and Brodatz texture show that the proposed method can positively find a new set of reconstruction parameters and enhance the classification accuracy rate greatly. But the prototype pattern vectors are got by SCAP, which is simple and rapid for it gets prototype pattern vectors by averaging the training samples simply.

Hereinafter, a new Immunity Clonal Synergetic Network inspired by decision fusion is proposed to solve the two problems, i.e. the solution of prototype pattern vectors and the reconstruction of order parameters, simultaneously.

2.3 Immunity Clonal Synergetic Network Inspired by Fuzzy Integral

Fuzzy integral provides the importance of each classifier that is measured subjectively. Final decision is integrated from the evidence of classifiers for each class and the importance of classifiers defined by users or generated from training data.

The fuzzy integral introduced by Sugeno [8] and the associated fuzzy measures [9] [10] provide a useful way for aggregating information, which is defined as follows.

Definition 1: Let X be a finite set of elements. A set function $g:2^x \to [0,1]$, with $g(\phi)=0$, $g(X)=1$ and $g(A)\le g(B)\ if\ A\subset B$, is called a fuzzy measure.

Definition 2: Let X be a finite set, and $h:X\to[0,1]$ be a fuzzy subset of X. The fuzzy integral over X of the function h with respect to a fuzzy measure g is defined by

$$h(x)\circ g(\cdot)=\max_{E\subseteq X}\left[\min\left(\min_{x\in E}h(x),g(E)\right)\right]\cdot \tag{4}$$

The calculation of the fuzzy integral is as follows. Let $Y=\{y_1,y_2,\cdots,y_n\}$ be a finite set and let $h:X\to[0,1]$ be a function. Suppose $h(y_1)\ge h(y_2)\ge\cdots\ge h(y_n)$, if not, Y is rearranged so that this relation holds. Then a fuzzy integral, e, with respect to a fuzzy measure g over Y can be computed by

$$e=\max_{i=1}^{n}\left[\min\left(h(y_i),g(A_i)\right)\right],\ A_i=\{y_1,y_2,\cdots,y_i\}\cdot \tag{5}$$

Note that when g is a g_λ-fuzzy measure [8],[9], the value of $g(A_i)$ can be determined recursively as

$$g(A_1)=g(\{y_1\})=g^1\cdot \tag{6}$$

$$g(A_i)=g^i+g(A_{i-1})+\lambda g^i g(A_{i-1}),\ \ for\ 1<i\le n\ \cdot \tag{7}$$

λ is given by solving the equation

$$\lambda+1=\prod_{i=1}^{n}(1+\lambda g^i)\quad \lambda\in(-1,+\infty)\ and\ \lambda\ne 0\ \cdot \tag{8}$$

This can be easily calculated by using the $(n-1)st$ degree polynomial and finding the unique root greater than -1.

Let $C=\{c_1,c_2,\cdots,c_N\}$ be a set of classes, $Y=\{y_1, y_2,\cdots, y_n\}$ be a set of classifiers, and $h_k:Y\rightarrow[0,1]$ be partial evaluation of the object A for class c_k. $h_k(y_i)$ is an indication of how certain we are in the classification of object A to be in class ω_k using the network y_i. The set Y is sorted by the value of $h_k(y_i)$ for each class in descending order. A_{ki} means a set of former i elements in Y for the class k.

$$Final\ class=\arg\max_{c_k\in C}\left[\max_{i=1}^{n}[\min(h_k(y_i),g(A_{ki}))]\right]\cdot \tag{9}$$

Algorithm [7]: Network fusion by fuzzy integral

```
        Calculate $\lambda$;
        For each class $\omega_k$ do
            For each network $y_i$ do
                Calculate $h_k(y_i)$;
                Determine $g_k(\{y_i\})$;
            End_for
            Compute the fuzzy integral for the class;
        End_for
        Determine the final class;
```

In the final step, the class ω_k with the largest integral value can be chosen as the output class.

3 Experiments

In order to verify the proposed method, we compare it with three methods including the learning algorithm of SNN based on ICS (SNNppv), the reconstruction of order parameters in SNN based on ICS (SNNoprc) and RBFNN (Radial Basis Function Neural Network), a more stable and commonly used method on classification. Subsequently, two types of data, Iris data and Brodatz textural images, are employed to test the performance of the algorithms above.

Experiment 1. IRIS data is selected to test the performance of the proposed method. This data set has 150 data in all, which is made up of 3 classes, and each datum possesses 4 attributes. We randomly select one third of the data from each class as training data and the remainder as testing data.

Experiment 2. Seven similar textural images, D16, D21, D53, D68, D77, D78 and D105, are chosen from Brodatz. Every image of size 640×640 is segmented into 100 nonoverlapping images as a class. We also randomly select one fourth of the samples from each class as training data and the remainder as testing data. Simultaneously, textural feature extraction method [11] is used to decompose images into 3 lays and 10-dimensional features are obtained.

In the two experiments, the parameters are set empirically as follows. The initial population is 5 and mutation probability is 0.1. The clonal scale is determined by the affinity. The halt condition is the classification accuracy rate keeping unchanged for several times. The fuzzy densities are represented by recognition rates of networks on training data and the corresponding λ is given in Table 1 and 3 respectively. The recognition results of the two experiments are shown in Table 2 and 4.

Table 1. Fuzzy densities and the correpsonding λ on Iris data

	SNNppv	SNNoprc	RBFNN	λ
g	0.8542	0.8750	0.8958	-0.9980

The g values, the degree of importance of each network, are assigned based on how well these networks performed on training data and they are obtained from the statistical average values of 10 times. From Table 1, it can be seen that the degree of importance can be interpreted as a plausibility value for Banon showed that $\lambda < 0$ if g is a plausibility measure [12].

Table 2. The recognition rates of different methods on Iris data

(%)	SNNppv	SNNoprc	RBFNN	Fuzzy Integral
Class 1	100.00	97.06	88.24	100.00
Class 2	73.53	88.24	97.06	88.24
Class 3	100.00	97.06	100.00	100.00
Average rate	91.18	94.12	95.10	96.08

Table 3. Fuzzy densities and the correpsonding λ on Brodatz texture

	SNNppv	SNNoprc	RBFNN	λ
g	0.8743	0.9086	0.8971	-0.9988

We also assign the g values from the statistical average values of 10 times. From Table 3 we can see that g is a plausibility measure for $\lambda < 0$ [12].

Table 4. The recognition rates of different methods on Brodatz texture

(%)	SNNppv	SNNoprc	RBFNN	Fuzzy Integral
Class 1	100.00	98.67	89.33	98.67
Class 2	100.00	100.00	100.00	100.00
Class 3	100.00	90.67	96.00	97.33
Class 4	93.33	94.67	96.00	97.33
Class 5	98.67	100.00	100.00	100.00
Class 6	88.00	94.67	100.00	97.33
Class 7	100.00	97.33	98.67	100.00
Average rate	97.14	96.57	97.14	98.67

From Table 2 and 4, it can be concluded that the new method has better recognition rate and is more robust because the decision fusion scheme, fuzzy integral, searches for the maximal grade of agreement between the objective evidence and the expectation and considers the relative importance of the different networks simultaneously. It should be pointed out that the worst result out of 10 times is chosen to conduct the experiments.

4 Conclusions

In this paper, we propose a new Immunity Clonal Synergetic Network inspired by fuzzy integral, which aggregates the improved Synergetic Neural Networks (SNN) by decision fusion scheme. Three networks, i.e. SNNppv, SNNoprc and RBFNN, are fused by fuzzy integral, which not only combines the classification results but also considers the relative importance of the different networks. Experiments show that the presented algorithm has better performance and higher recognition rate. But how to select networks and how to aggregate their outputs by some alternative aggregating mechanism deserve our further research.

References

1. Haken H.: Synergetic Computer and Cognition–A Top-Down Approach to Neural Nets. Springer-Verlag, Berlin (1991)
2. Wagner T., Boebel F.G.: Testing Synergetic Algorithms with Industrial Classification Problems. Neural Networks, **7** (1994) 1313-1321
3. Wang H.L.: The Research of Application of Image Recognition Using Synergetic Neural Network. Ph.D. Dissertation, Shanghai Jiao Tong University, China (2000)
4. Ma X.L., Jiao L.C.: An Effective Learning Algorithm of Synergetic Neural Network. Lecture Notes in Computer Science, Springer-Verlag, Berlin , **3173** (2004) 258-263
5. Hu D.L., Qi F.H.: Reconstruction of Order Parameters in Synergetics Approach to Pattern Recognition. J. Infrared Millim. Waves, **17** (1998) 177-181
6. Ma X.L., Jiao L.C.: Reconstruction of Order Parameters Based on Immunity Clonal Strategy for Image Classification. Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg , **3211** (2004) 455-462
7. Cho S.B., Kim J.H.: Combining Multiple Neural Networks by Fuzzy Integral for Robust Classification. IEEE Transactions on Systems, Man, and Cybernetics, **25** (1995) 380-384
8. Sugeno, M.: Fuzzy Measures and Fuzzy Integrals: A Survey. Fuzzy Automata and Decision Processes. Amsterdam: North Holland (1977) 89-102
9. Leszeynski, K., Penczek, P., Grochulskki, W.: Sugeno's Fuzzy Measures and Fuzzy Clustering, Fuzzy Sets and Systems, **15** (1985) 147-158
10. Yager, R.R.: Element Selection from A Fuzzy Subset Using The Fuzzy Integral. IEEE Trans. Syst. Man Cybern, **23** (1993) 467-477
11. Hu, Z.L., Guo, D.Z., Sheng, Y.H.: Extracting Textural Information of Satellite SAR Image Based on Wavelet Decomposition, Journal of Remote Sensing, **5** (2001) 423-427
12. Banon, G.: Distinction Between Several Subsets of Fuzzy Measures, Fuzzy Set & Systems, **5** (1981) 291-305

An Improved Optimal Pairwise Coupling Classifier

Roger Xu, Tao Qian, and Chiman Kwan

Intelligent Automation, Inc., 15400 Calhoun Drive, Suite 400, Rockville, MD 20855
{hgxu,tqian,ckwan}@i-a-i.com

Abstract. The Optimal Pairwise Coupling (O-PWC) classifier was proposed and used for data classification because of its excellent classification performance [1]. A key step in the O-PWC algorithm is to calculate a number of posterior probabilities, which was achieved using an iterative procedure in [1],[2]. In this paper, we will present an analytical solution to the problem of finding the posterior probabilities. As a result, the computational efficiency of the O-PWC algorithm will be significantly improved, which will be shown by one numerical example.

1 Introduction

For a K-class classification problem, the one-against-one method is being widely used. In this method, K(K-1)/2 two-class classifiers are constructed. Each of them is utilized to discriminate two of the K-classes. MaxVoting (MV) and Pairwise Coupling (PWC)[2] are two common ways to combine the results from these binary classifiers. MV considers the output of each classifier as a binary decision and selects the class that wins the most votes. In PWC, however, each of the binary classifier outputs a posterior probability, called pairwise probability, for a given test sample. PWC combines these pairwise probabilities into K posterior probabilities. PWC selects the class that has the largest posterior probability. In many applications PWC performs better than MV.

However, the original PWC has some drawbacks. When a test sample is classified by one of the K(K-1)/2 classifiers and it doesn't belong to either of the two involved classes of this classifier, the probabilistic measures of the test sample against the two classes become meaningless and therefore may damage the final fused output of PWC. One new approach called optimal PWC (O-PWC)[1] was proposed to tackle this problem. The key idea of O-PWC is as follows. For the same K-class classification problem, an array of K O-PWC classifiers is constructed, each of which is the most reliable and optimal for one corresponding class in the sense of cross entropy or square error. The final decision is made by combining the results of these K O-PWCs. In a face recognition application considered in [1], O-PWC yields 98.11% of recognition accuracy whereas MV 94% and PWC 95.13%, respectively. However, the original O-PWC is computationally more complex than PWC. First, only one PWC classifier needs to be constructed for a K class classification problem in PWC whereas K similarly formulated O-PWCs need to be formed in O-PWC. Second, for each O-PWC, an iterative procedure is required in [1] to calculate the K posterior probabilities. In contrast, only one iterative procedure is needed in PWC.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 32–38, 2005.

In this paper, we will present a simple analytical solution to the problem of finding the K posterior probabilities. Thus, our approach is computationally more efficient than the iterative procedures. However, it should be noted that the uniqueness of the O-PWC structure has made the analytical solution possible. In other words, this solution can not be applied to the calculation of the posterior probabilities in PWC.

The paper is organized as follows. We will describe the O-PWC classification algorithm in Sec. 2. In Sec. 3, we will detail our analytical solution to the problem of finding the K posterior probabilities. One numerical example will be given to show the computational efficiency of the analytical solution. Concluding remarks will be given in Sec. 4.

2 Review of The O-PWC Classification Algorithm

2.1 The PWC Algorithm

For K classes, the probability of $\vec{x}$ belonging to class c_i, given $\vec{x}$ is in either class c_i or c_j, can be written as a pairwise probability:

$$r_{ij} = p(c_i | \vec{x}, \vec{x} \in c_i \cup c_j), \ j \neq i . \tag{1}$$

Going through all K(K-1)/2 binary classifiers, a *pairwise probability matrix*, can be produced.

To couple the pairwise probability matrix into a common set of probabilities P_i, Haste and Tibshirani in [3] proposed a Pairwise Coupling (PWC) method. They introduced a new set of auxiliary variables:

$$\mu_{ij} = \frac{P_i}{P_i + P_j} \tag{2}$$

and found P_i's such that the corresponding μ_{ij}'s are in some sense "close" to the observed r_{ij}'s. The Kullback-Leibler divergence between μ_{ij} and r_{ij}:

$$l(\vec{p}) = \sum_{i<j} n_{ij} \left[\log \frac{r_{ij}}{\mu_{ij}} + (1 - r_{ij}) \log \frac{1 - r_{ij}}{1 - \mu_{ij}} \right] \tag{3}$$

is selected as the closeness measure in [3]. Minimizing this function can find P_i's. An iterative procedure is proposed to solve such constrained minimization problem in [3]:

Step 1. Initialize P_i, and compute corresponding μ_{ij};

Step 2. Repeat until convergence:

$$P_i \leftarrow P_i \cdot \frac{\sum_{j \neq i} n_{ij} r_{ij}}{\sum_{j \neq i} n_{ij} \mu_{ij}} \tag{4}$$

renormalize the P_i, and recompute the μ_{ij}.

Step 3. $P \leftarrow P / \sum P_i$

For simplicity, the weights are assumed to be equal, that is, $n_{ij} = 1$ for all i, j. A non-iterative estimate of P can be obtained simply as:

$$P_i = \frac{2}{K(K-1)} \sum_{j \neq i} r_{ij}, \quad i, j = 1, 2, \ldots, K \tag{5}$$

Let the posterior probabilities of $\vec{x}$ be $\vec{P}(\vec{x}) = (P_1, \ldots, P_k)$. The final decision rule is: $d(\vec{x}) = \arg \max_i [P_i(\vec{x})]$.

2.2 The O-PWC Classification Algorithm

The following weight matrix W was introduced in [1] for optimal classifier construction:

$$W = \begin{bmatrix} - & W_{12} & W_{13} & \cdots & W_{1K} \\ W_{21} & - & W_{23} & \cdots & W_{2K} \\ W_{31} & W_{32} & - & \cdots & W_{3K} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ W_{K1} & W_{K2} & W_{K3} & \cdots & - \end{bmatrix} \tag{6}$$

where, $W_{ij} = W_{ji}, \quad i, j = 1, \ldots, K, i \neq j$. This weight matrix reflects the influence of each pairwise classifier to the final decision for a given pattern $\vec{x}$. In PWC, the weights are all assumed to be 1, which means that all binary classifiers' contributions to the final decision are the same. In fact, given a testing pattern $\vec{x} \notin c_i \cup c_j$, the pairwise probability r_{ij} is absolutely irrelevant to $\vec{x}$ because the corresponding binary classifier which is used to discriminate class c_i and c_j has not been trained with data from the true class. Consequently, using it to find $\vec{P}$ is very likely to damage the result of the calculation. In other works, not all binary classifiers are useful and relevant to the final decision for a given pattern. The results from some classifiers are meaningless, or even harmful.

The O-PWC was introduced in [1] to avoid the classification inaccuracy caused by the irrelevant pairwise probabilities as mentioned above. The optimal weight matrix for class c_i is defined as:

$$W^i = \{W_{pq}\} \tag{7}$$

which satisfies:

$$\begin{cases} W_{pq} = 1, & if \quad p = i \quad or \quad q = i \\ W_{pq} = 0, & otherwise \end{cases}, \quad p, q = 1, 2, \ldots, K, \ p \neq q \tag{8}$$

Considering a 5-class classification problem, the *optimal weight matrix* for class c_2 and c_4 can be represented as follows respectively:

$$00\begin{bmatrix} - & 1 & 0 & 0 & 0 \\ 1 & - & 1 & 1 & 1 \\ 0 & 1 & - & 0 & 0 \\ 0 & 1 & 0 & - & 0 \\ 0 & 1 & 0 & 0 & - \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} - & 0 & 0 & 1 & 0 \\ 0 & - & 0 & 1 & 0 \\ 0 & 0 & - & 1 & 0 \\ 1 & 1 & 1 & - & 1 \\ 0 & 0 & 0 & 1 & - \end{bmatrix}$$

This means that, given a pattern $\vec{x} \in c_i$, only those binary classifiers, which involve class c_i, are considered. The rest binary classifiers will be ignored. Hence, all pairwise probabilities in *O-PWC^i* are relevant to class c_i.

Given a pattern $\vec{x}$, the output of *O-PWC^i* can be represented as a probability vector: $\vec{P}^{c_i}(\vec{x}) = (P_1^{c_i}, P_2^{c_i}, \ldots, P_K^{c_i}), \quad i \in \{1, 2, \ldots, K\}$. Each of its components represents the probability of $\vec{x}$ belonging to the respective class.

In [1], the afore-described iterative procedure and W^i were proposed and used to construct the O-PWC classifier for c_i. After K O-PWC classifiers were constructed, the final classification decision can be made based on minimal square errors or cross entropy [1].

3 An Analytical Solution

It is obvious that iterative procedures are computationally costly as a number of iterations are needed for convergence. This may affect the real-time implementation of the algorithm depending on how fast the solution converges. In this section, we will present an analytical way to construct the O-PWC for C_i. In other words, a formula will be derived to solve $\vec{P}^{c_i}(\vec{x}) = (P_1^{c_i}, P_2^{c_i}, \ldots, P_K^{c_i}), \quad i \in \{1, 2, \ldots, K\}$.

First let us denote the original PWC matrix as R and assume that the *pq*th entry of R, r_{pq} satisfies: $0<r_{pq}<1.\ p=1,2,\ldots K,\ q=1,2,\ldots,K.$ After applying the weight matrix to R and taking the fact that $r_{pq}+r_{qp}=1$ into account, we can form an O-PWC matrix R^{c_i} for class c_i, $i=1,2,\ldots,K$ The *pq*th entry of R^{c_i} satisfies:

$$\begin{cases} r_{pq}^{c_i} = r_{pq}, & if \quad p = i \quad or \quad q = i \\ r_{pq}^{c_i} = 0, & otherwise \end{cases}, \quad p, q = 1, 2, \ldots, K,\ p \neq q \tag{9}$$

Reconsidering the 5 class classification problem, the O-PWC matrix for class c_2 is:

$$R^{c_2} = \begin{bmatrix} - & r_{12} & 0 & 0 & 0 \\ r_{21} & - & r_{23} & r_{24} & r_{25} \\ 0 & r_{32} & - & 0 & 0 \\ 0 & r_{42} & 0 & - & 0 \\ 0 & r_{52} & 0 & 0 & - \end{bmatrix} \tag{10}$$

In both PWC and the orignal O-PWC classification algorithms [1], the iterative procedure was used to solve the constrained minimization problem of finding posterior probabilities. The computation time of such solutions depends on the choice of convergence error thresholds. After scrutinizing the O-PWC algorithm, we have found that finding posterior probabilities in O-PWC becomes a problem of solving a set of unconstrained linear equations after the optimal weight matrix is applied to R for each class. Moreover, the solution is unique and simple.

Let us consider constructing an O-PWC classifier for class 2 for the same 5 class classification problem. Based on (10) and $\sum_{j=1}^{5} P_j^{C_2} = 1$, the following 5 linear independent equations are obtained:

$$\begin{gathered} r_{12} = \frac{P_2^{C_2}}{P_1^{C_2} + P_2^{C_2}},\ r_{32} = \frac{P_2^{C_2}}{P_3^{C_2} + P_2^{C_2}}, \\ r_{42} = \frac{P_2^{C_2}}{P_4^{C_2} + P_2^{C_2}},\ r_{52} = \frac{P_2^{C_2}}{P_5^{C_2} + P_2^{C_2}} \text{ and} \\ P_1^{C_2} + P_2^{C_2} + P_3^{C_2} + P_4^{C_2} + P_5^{C_2} = 1 \end{gathered} \tag{11}$$

After we take into account the fact that $r_{pq}+r_{qp}=1$, $p=1,...K$, $q=1,...,K$, the solution to (11) becomes:

$$P_k^{c_2} = \begin{cases} \frac{r_{2k}}{r_{k2}} P_2^{c_2} & k = 1, 2, \cdots 5, k \neq 2 \\ \frac{1}{1 + \sum_{j=1, j\neq 2}^{5} \frac{r_{2j}}{r_{j2}}} & k = 2 \end{cases} \tag{12}$$

It is straightforward to extend (12) to the K class classification problem. For class c_i, $i=1,2,...,K$, the solution of $P_k^{c_i}$, $k=1,2,...,K$ is as follows:

$$P_k^{c_i} = \begin{cases} \frac{r_{ik}}{r_{ki}} P_i^{c_i} & k = 1, 2, \cdots K, k \neq i \\ \frac{1}{1 + \sum_{j=1, j\neq i}^{K} \frac{r_{ij}}{r_{ji}}} & k = i \end{cases} \tag{13}$$

To calculate all the K posterior probabilities for class c_i, we only need 2K-1 divisions and K-1 additions. Compared with the iterative procedure, our solution is not only computationally more efficient but also mathematically more elegant.

The following numeric example will show the improved computational efficiency of the analytical solution over the iterative procedures. A random pairwise probability matrix was generated for 5 classes as follows:

$$r = \begin{bmatrix} - & 0.5492 & 0.2841 & 0.1072 & 0.7269 \\ 0.4508 & - & 0.7676 & 0.1951 & 0.0916 \\ 0.7159 & 0.2324 & - & 0.9216 & 0.3592 \\ 0.8928 & 0.8049 & 0.0784 & - & 0.0057 \\ 0.2731 & 0.9084 & 0.6408 & 0.9943 & - \end{bmatrix}$$

Using the iterative O-PWC procedure in [1], the output of O-PWC vectors for 5 classes was:

$$\begin{aligned}
\vec{P}_1 &= (0.07663984133669 \quad 0.06289660153882 \quad 0.19310757739254 \quad 0.63856164617774 \quad 0.02879433363081)\\
\vec{P}_2 &= (0.07356899721141 \quad 0.06037642852325 \quad 0.01827459430499 \quad 0.24904276997371 \quad 0.59873721005536)\\
\vec{P}_3 &= (0.06040869194258 \quad 0.50287937632761 \quad 0.15221033796954 \quad 0.01294559499882 \quad 0.27155599881062)\\
\vec{P}_4 &= (0.00064037697499 \quad 0.00129353273370 \quad 0.06273444732201 \quad 0.00533560832942 \quad 0.92999603473692)\\
\vec{P}_5 &= (0.61487704418754 \quad 0.02329549185090 \quad 0.12948678986034 \quad 0.00132538957989 \quad 0.23101528459744)
\end{aligned}$$

where the iteration stop threshold for the error of O-PWC outputs is 1.0×10^{-20}. The computation time in Matlab 6.5 for this iterative procedure was 0.658 seconds.

After decreasing the threshold for the error to 1.0×10^{-30}, the O-PWC output became:

$$\begin{aligned}
\vec{P}_1 &= (0.07663984127486 \quad 0.06289660148363 \quad 0.19310757719940 \quad 0.63856164643596 \quad 0.02879433360615)\\
\vec{P}_2 &= (0.07356899714034 \quad 0.06037642847004 \quad 0.01827459428808 \quad 0.24904276966926 \quad 0.59873721043228)\\
\vec{P}_3 &= (0.06040869192532 \quad 0.50287937645378 \quad 0.15221033793176 \quad 0.01294559499527 \quad 0.27155599869387)\\
\vec{P}_4 &= (0.00064037696382 \quad 0.00129353271113 \quad 0.06273444619985 \quad 0.00533560823654 \quad 0.92999603588866)\\
\vec{P}_5 &= (0.61487704423335 \quad 0.02329549184328 \quad 0.12948678981423 \quad 0.00132538957946 \quad 0.23101528452968)
\end{aligned}$$

The computation time was 1.043 seconds.

We applied the analytical solution to the same data set, the result was:

$$\begin{aligned}
\vec{P}_1 &= (0.07663984127486 \quad 0.06289660148363 \quad 0.19310757719940 \quad 0.63856164643596 \quad 0.02879433360615)\\
\vec{P}_2 &= (0.07356899714034 \quad 0.06037642847004 \quad 0.01827459428808 \quad 0.24904276966926 \quad 0.59873721043228)\\
\vec{P}_3 &= (0.06040869192532 \quad 0.50287937645378 \quad 0.15221033793176 \quad 0.01294559499527 \quad 0.27155599869387)\\
\vec{P}_4 &= (0.00064037696382 \quad 0.00129353271113 \quad 0.06273444619985 \quad 0.00533560823654 \quad 0.92999603588866)\\
\vec{P}_5 &= (0.61487704423335 \quad 0.02329549184328 \quad 0.12948678981423 \quad 0.00132538957946 \quad 0.23101528452968)
\end{aligned}$$

The computation time was only 0.015 seconds. The analytical solution is about 70 times faster than iterative algorithm in this case.

We applied the improved O-PWC algorithm to a NASA funded project on Electronic Nose data analysis. The results showed that O-PWC performed better than other conventional algorithms such as PNN, MLP and LVQ classifiers. Due to the NASA regulations, the simulation details are omitted here.

4 Conclusions

In this paper, we have provided an analytical solution to the problem of finding posterior probabilities for the O-PWC classifiers. Simulation results have shown that this solution is computationally more efficient than iterative procedures in [1].

It should be noted that the improvement we have made to the original O-PWC will not increase its classification performance. Moreover, for the general PWC classification problem, the analytical solution does not exist.

In the future, we will implement the improved O-PWC classification algorithm in applications, where both classification performance and computational efficiency are of main concern.

References

1. Li, Z., Tang, S.: Face Recognition Using Improved Pairwise Coupling Support Vector Machines. Proceedings of the 9th International Conference on Neural Information Processing (2002)
2. Platt, J.: Probabilistic Outputs for SVMs and Comparisons to Regularized Likehood Methods, Advances in Large Margin Classifiers. MIT Press (1999)
3. Hastie, T., Tibshirani, R.: Classification by Pairwise Coupling, NIPS*97 (1996)

Improvement on Response Performance of Min-Max Modular Classifier by Symmetric Module Selection[*]

Hai Zhao and Baoliang Lu

Department of Computer Science and Engineering, Shanghai Jiao Tong University
1954 Huashan Rd., Shanghai 200030, China
{zhaohai,blu}@cs.sjtu.edu.cn

Abstract. This paper aims to improve the response performance of min-max modular classifier by a module selection policy for two-class classification during recognition. We propose an efficient base classifier selection algorithm. We show that the quadratic complexity of original min-max modular classifier can fall onto the level of linear complexity in the number of base-classifier modules for each input sample under presented selection scheme. The experimental results indicate the effectiveness of our algorithm and verify our theoretical analysis.

1 Introduction

Two-class classification is one kind of basic classification problem. Many essential classification schemes often start from binary classifier and then adapt to multi-class classifiers. Therefore, an improvement on two-class problem processing will earn important value to pattern classification.

In this paper, we consider a decomposition policy of two-class classification task, which is one of two processing stages in min-max modular (M^3) neural network[1]-[2]. The policy is based on partition of training set, which make it possible that a large-scale two-class problem is divided into some smaller scale two-class subproblems for parallel processing.

However, one hand, it is obvious that such partition will yield much many modules, which cause serious performance problems. On the other hand, only partial sub-classifier modules will actually function for each input sample in classification. This activates us that it is possible to optimize the original combination procedure of classifiers only to check those necessary modules, instead of all produced sub-classifier modules for an input sample. This is to say, an additional module selection procedure can be introduced for improving the response performance of the classifier.

In this paper, we will present an efficient module selecting policy named symmetrical selection algorithm to improve the response performance of min-max modular classifier.

[*] This work was supported in part by the National Natural Science Foundation of China via the grants NSFC 60375022 and NSFC 60473040.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 39–44, 2005.

2 Task Decomposition and Module Combination

Consider a two-class classification problem, whose output coding of class labels are denoted by C_0 and C_1(as two values, $C_1 > C_0$), or equally, 0 and 1, which will be concise and not lose any generality for our algorithm description. Suppose the training set of class C_0 is decomposed into n subsets and the training set of class C_1 is divided into m subsets. By arranging those m and n training subsets in pairs, we obtain $m \times n$ training set pairs. Each pair is learned by a single binary sub-classifier. Therefore, a larger scale two-class problem can be decomposed into $m \times n$ smaller subproblems. We call the binary sub-classifier as base classifier.

Suppose all produced training set pairs are denoted by X_{ij}, for $i = 0, 1, ...,$ $m-1$, and $j = 0, 1, ..., n-1$. Without misunderstanding, we also express X_{ij} as the output of the corresponding base classifier.

Min-max combination defines how the outputs of those $m \times n$ base classifiers are combined into the solution to the original problem. Before combination, a grouping operation on $m \times n$ base classifiers should be done: these base classifiers, X_{ij}, for $j = 0, ..., n-1$, are defined as one "C_1 group" and i is defined as its group label, and those base classifiers, X_{ij}, for $i = 0, ..., m-1$, are defined as one "C_0 group" and j is defined as its group label.

Min-max combination of all base classifiers includes two stages: Firstly, the minimization rule is applied to each C_1 group to make the output of the group. Secondly, the outputs of all groups are integrated by the maximization rule to make the final output of the original two-class classification problem.

An outstanding merit for min-max modular classifier is that imbalanced classification can always be avoid from a simple decomposition if needed[4].

3 Symmetrical Module Selection Algorithm

We call a C_1 group whose all member base classifiers hold the class C_1 as a "winning group of class C_1" and a class C_0 group whose all member base classifiers hold the class C_0 as a "winning group of class C_0" . It is easy to demonstrate that in the original min-max combination procedure for one input sample, the fact that there is a winning group of C_1 in all $m \times n$ base classifiers will be sufficient and necessary to make the conclusion that the final combination result must be class C_1 according to the combining rules, vice versa, if any winning group of C_1 does not exist, then the final combination result must be class C_0. Thus, a min-max combination of base classifiers will be redescribed as such a search procedure to find a winning group of class C_1.

Similarly, we can also define one symmetrical *max-min* combination to perform a search procedure for a winning group of class C_0. Naturally, that there exists a winning group of class C_0 in all $m \times n$ base classifiers is a sufficient and necessary criterion for a combination result of class C_0. Symmetrical module selection algorithm, just like its name, is a procedure with consideration of the winning group of both class C_0 and class C_1, instead of one-sided search for the winning group of class C_1 in the original model[1],[2].

Basically, symmetrical module selection is based on the following idea: a) Notice that it is impossible that there exists one winning group of class C_1 and one winning group of class C_0 at the same time. The reason is that any winning group of class C_1, X_{ij}, for $j = 0, ..., n-1$, and any winning group of class C_0, X_{ij}, for $i = 0, ..., m-1$, must share one same base classifier, which can not output two different classification results at the same time. b) The symmetrical module selection algorithm is a procedure to continuously exclude those hopeless class C_1 or C_0 group. The existence of a winning group of class C_0 or class C_1 means final combination classification result is class C_0 or class C_1, respectively. Regarding all outputs of base classifiers as a binary matrix, a row can stand for a class C_1 group, and a column can stand for a class C_0 group. Thus, that a base classifier outputs 1 as classification result means that the column it locates loses the chance to be a winning group of class C_0. Symmetrically, that a base classifier outputs 0 as classification result means that the column it locates loses the chance to be a winning group of class C_1.

The symmetrical module selection algorithm is described below in detail. For convenience, we also say the first subscript of a base classifier is its row, and the second is its column, also a row means a class C_1 group and a column means a class C_0 group.

1. For $i = 0, ..., m-1$, set the tag $RF[i] = 0$, and for $j = 0, ..., n-1$, set the tag $CF[j] = 0$, which means all class one and zero groups are not excluded, respectively.
2. Set two counters, $RCounter = m$ and $CCounter = n$, respectively, which means the number of surviving class one and zero groups without excluded.
3. Set starting row and column, $i = 0$ and $j = 0$.
4. Repeat the following operations:
   - (a) If i is an excluded group label, that is, $RF[i] = 1$, then i increases continuously until i is not an excluded label any more or $i = m$.
   - (b) If j is an excluded group label, that is, $CF[j] = 1$, then j increases continuously until j is not an excluded label any more or $j = n$.
   - (c) If the counter, $RCounter = 0$, or $i = m$, then output the combination classification result class 0 and the algorithm ends here.
   - (d) If the counter, $CCounter = 0$, or $j = n$, then output the combination classification result class 1 and the algorithm ends here.
   - (e) Check the base classifier X_{ij}.
   - (f) If X_{ij} holds class C_1, then
     - i. Set the tag, $CF[j] = 1$.
     - ii. Subtract 1 from the counter, $CCounter = CCounter - 1$
     - iii. $j = j + 1$.
   - (g) If X_{ij} holds class C_0, then
     - i. Set the tag, $RF[i] = 1$.
     - ii. Subtract 1 from the counter, $RCounter = RCounter - 1$
     - iii. $i = i + 1$.

4 Performance Estimation

We now show that the number of checked base classifiers in symmetrical module selection algorithm for each input sample will be never larger than $m+n-1$.

In fact, the algorithm is equally a search procedure in a binary matrix with m rows and n columns, in which the start point is top left corner and the end point is bottom right corner. Each checking for a base classifier is equal to an access for one element in the matrix, which means one row or one column in the matrix must be excluded. In turn, the accessing of the algorithm gets to the next row or column without any backtracking. In summary, such search from top left corner to bottom right corner in this matrix mostly covers $m+n-1$ elements. Namely, the number of checked base classifier in symmetrical module selection for each input sample will not be larger than $m+n-1$, which is much better than the case, $m \times n$, under the original model.

5 Experimental Results

Three data sets shown in Table 1 from UCI Repository[3] have been chosen for this study. For a typical realization of min-max modular classifier, support vector machine(SVM) with RBF kernel is selected as base classifier[4]. Two parameters, C and γ, are set to 8 and 0.25 for Internet Ads and Adult data sets, respectively. These two parameters are set to 316.2 and 1 for Banana data set, respectively. The same numbers of samples for class C_1 and C_0 in every subsets are taken.

Table 1. Distributions of samples in three data sets

Data Set	Number of Training Samples			Number of Test Samples		
	Total	Class C_1	Class C_0	Total	Class C_1	Class C_0
Internet Ads	2100	1800	300	1179	1020	159
Adult	32561	24720	7841	16281	12435	3846
Banana	40000	21847	18153	490000	270553	219447

Experimental results are shown in Tables 2 through 4. The comparison between experimental results and theoretical bounds in three data sets are shown in Fig 1. Fig 1(a) shows 20 different experimental results, where n varies from 2 to 21 and m correspondingly varies from 12 to 129 for Internet Ads data set. Fig 1(b) shows 100 different experimental results, where n varies from 2 to 101 and m correspondingly varies from 6 to 317 for Adult data set. Note that Table 2 and Table 3 only show parts of experimental results shown in Fig.1(a) and Fig.1(b) for space limitation.

As expected, the experimental results show that our symmetrical module selection algorithm gives an outstanding performance improvement for testing procedure with respect to the case without any selection, while combining accuracies keep the same. The number of actual checked base classifier is also strictly

Table 2. Experimental results on Internet Ads data set

#Modules	m	n	Without Selection			Symmetrical Selection			
			#Modules	Time(ms)	acc.(%)	#Modules	Bound	Time(ms)	acc.(%)
14	129	21	2709	63656	90.16	29.04	149	26718	90.16
18	100	17	1700	44046	89.40	19.78	116	15484	89.40
23	78	13	1014	29890	89.31	15.07	90	8265	89.31
33	56	9	504	17859	89.57	10.62	64	3375	89.57
60	30	5	150	8421	89.90	5.97	34	1000	89.90

Table 3. Experimental results on Adult data set

#Modules	m	n	Without Selection			Symmetrical Selection			
			#Modules	Time(ms)	acc.(%)	#Modules	Bound	Time(ms)	acc.(%)
320	77	25	1925	7027	76.37	25.45	101	116	76.37
157	157	50	7850	15085	76.36	96.79	206	281	76.36
105	235	75	17625	22613	76.39	183.19	309	430	76.39
77	321	102	32742	37679	76.37	320.97	422	666	76.37

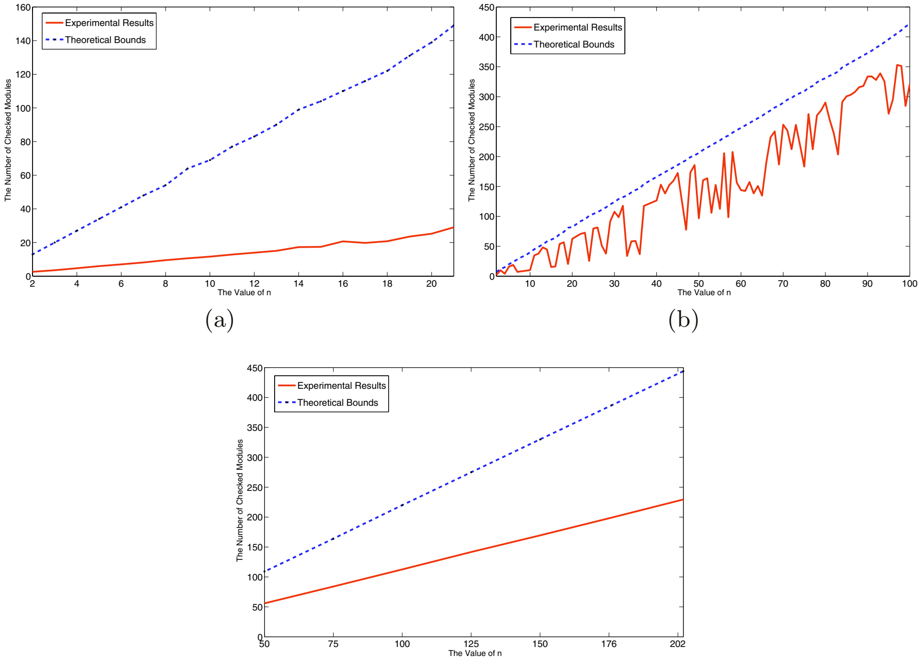
Table 4. Experimental results on Banana data set

#Modules	m	n	Without Selection			Symmetrical Selection			
			#Modules	Time(ms)	acc.(%)	#Modules	Bound	Time(ms)	acc.(%)
90	243	202	49086	154023	90.81	229.56	444	1389	90.81
103	212	176	37312	130945	90.73	199.20	387	1279	90.73
121	181	150	27150	107417	90.99	169.43	330	1096	90.99
145	151	125	18875	88564	90.68	141.66	275	883	90.68
181	121	100	12100	66653	90.61	112.55	220	754	90.61
242	90	75	6750	52709	90.80	84.01	164	679	90.80
363	60	50	3000	32309	90.75	55.87	109	645	90.75

under the theoretical bound. However, the ratio of numbers of checked modules and the ratio of practical response time between two cases are not exactly the same, which may mostly attribute to different number of support vectors obtained by each base classifier.

6 Conclusions

An efficient module selection policy has been presented for improving the response performance of min-max modular classifier. We show that the quadratic



(a)

(b)

(c)

Fig. 1. Comparison of theoretical bound and experimental result on Internet Ads data set(a), Adult data set(b), and Banana data set(c)

complexity of the original min-max modular classifier can be reduced onto linear complexity of the number of base-classifiers for each input sample. The experimental results indicate that an outstanding improvement on the response performance for such modular classifier is obtained.

References

1. Lu, B.L., Ito, M.: Task Decomposition Based on Class Relations: a Modular Neural Network Architecture for Pattern Classification. Lecture Notes in Computer Science. Vol. 1240. Springer (1997) 330-339
2. 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
3. Blake, C.L., Merz, C.J: UCI Repository of Machine Learning Databases [http://www.ics.uci.edu/ mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science (1998)
4. Lu, B.L., Wang, K. A., Utiyama, M., Isahara, H.: A Part-versus-part Method for Massively Parallel Training of Support Vector Machines. Proc. of IEEE/INNS Int. Joint Conf. on Neural Networks (IJCNN2004), Budabest, Hungary (2004) 25-29

Principle for Outputs of Hidden Neurons in CC4 Network*

Zhenya Zhang[1,3], Shuguang Zhang[2], Xufa Wang[3],
Shuangping Chen[1,3], and Hongmei Cheng[4]

[1] Electronical Engineering & Information Science Department, University of Science and Technology Of China (USTC), Hefei 230027, China
zhenyazhang@ustc.edu
http://mail.ustc.edu.cn/~zzychm
[2] Statistics&Finance Department of USTC, Hefei 230027, China
sgzhang@ustc.edu.cn
[3] Computer Science Department of USTC 230027, Hefei, China
{zzychm,chamos}@mail.ustc.edu.cn, xfwang@ustc.edu.cn
[4] Mathematics Department of Anhui Normal University,Wuhu 241000,China
zzychm@mail.ahwhptt.net.cn

Abstract. Corner classification (CC) is a kind of algorithms for instantly classification. The feed forward neural network trained by CC algorithm can be used validly by information retrieval, especially online information retrieval. CC4 is the fourth version of CC. In this paper, the generalized distance is defined according to the construction of weight matrix of hidden layer of CC4. With the definition of generalized distance, the principle for outputs of hidden neurons in CC4 network is given.

1 Introduction

With the explosive growth of information sources available on the WWW, many search engines have emerged to provide convenience for the user [1]. Once a search engine accepts a query from a user, it will retrieve and recommend web pages that match the query according to certain criteria. But the user may be interested in only a small portion of those recommendations. Providing accurate search results to users with information classification approach can help users mine the Web more efficiently. Anvish [1], [2], [3], a metasearch engine, has provided an efficient neural network-based classification algorithm, CC4 neural network. The CC4 neural network requires each training sample to be presented only once. Compared to back propagation, the corner classification approach has been proved to have comparable generalization capability for pattern recognition and prediction problems and is much faster in speed [5], [6], [7].

Although some features (with generalization radius as representative), of CC4 are discussed by lots of papers [3], [4], [5], [6], [7], there is no any interpretation on con-

* This paper was granted financial support from "21st Century Education Revitalization Project" and "China Postdoctoral Science Foundation (2004036463)".

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 45–50, 2005.

structions of weight matrix of hidden and output layer. This paper gives an interpretation for outputs of neurons in hidden layer of CC4. This paper is organized as following. CC4 is introduced at session2. The definition of generalized distance in weight matrix of hidden layer of CC4 is presented at session3 while the principle for outputs of hidden neurons of CC4 is given at same session. Conclusion is given at sesction4.

2 The Introduction of CC4

The CC4 network can map an input binary vector X to a binary vector Y. The topology of a CC4 is given at fig1. The transfer function *f* of each neuron is given by equation (1). Suppose V=(v_1, v_2…v_k), *f*(V) =(*f*(v_1), *f*(v_2)…*f*(v_k)).

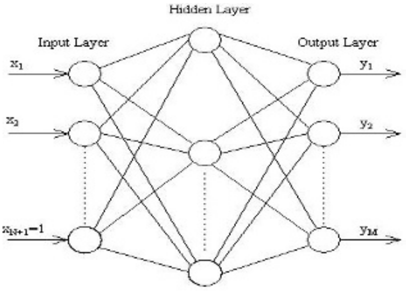


Fig. 1. The topology of CC4 network

$$y = f(x) = \begin{cases} 1 & x > 0 \\ 0 & x \le 0 \end{cases} \tag{1}$$

Here *x* is the input of a binary neuron.

Suppose the dimension of the binary vector presented to a CC4 as input is N, there are N+1 input neurons in the CC4. The input for the (N+1)*th* input neuron is 1 and the input for each other N input neuron is one component of the presented vector. The number of hidden neurons is equal to the number of training samples. The number of output neurons is equal to the dimension of the target binary vector.

Let w_{ij} be the weight of the connection from input neuron *i* to hidden neuron *j* and let X_{ij} be the input for the i*th* input neuron when the j*th* training sample is presented, here *i*=1,2…N+1 and *j*=1,2…H. w_{ij} can be valued according to (2).

$$w_{ij} = \begin{cases} 1 & \text{if } X_{ij} = 1, i < N+1 \\ -1 & \text{if } X_{ij} = 0, i < N+1 \\ r - s + 1 & \text{if i = N+1} \end{cases} \tag{2}$$

Here *r* is generalization radius and *s* is the number of 1 in the sample vector.

Let u_{jk} be the weight of connection from the j*th* hidden neuron to the k*th* output neuron and let Y_{jk} be the output of the k*th* output neuron for the j*th* training sample, here *j*=1,2…H and *k*=1,2…M. The value of u_{jk} is determined by equation (3).

$$u_{jk} = \begin{cases} 1 & \text{if } \mathrm{Y}_{\mathrm{jk}} = 1 \\ -1 & \text{if } \mathrm{Y}_{\mathrm{jk}} = 0 \end{cases} \tag{3}$$

Let a probe vector X=(x_1,x_2…x_n) is presented to the CC4 and the output vector of the CC4 is Y. Y is computed according to equation (4).

$$\mathrm{Y}=f(f((\mathrm{X},\ 1)\times \mathrm{W})\times \mathrm{U})=f(f((x_1,x_2\ldots x_N,1)\ \times \mathrm{W})\times \mathrm{U}) \tag{4}$$

Here W is the weight matrix of hidden layer and U is the weight matrix of output layer.

3 The Principle for Outputs of Hidden Neurons in CC4 Network

Definition 1. Let X=(x_1, x_2…x_n) be an n-dimension binary vector. X'=(x_1', x_2'…x_n') is the derived vector of X. Here X' is constructed according to equation (5).

$$x_i' = \begin{cases} 1 & \text{if } \mathrm{x}_{\mathrm{i}} = 1 \\ -1 & \text{if } \mathrm{x}_{\mathrm{i}} = 0 \end{cases} \tag{5}$$

Definition 2. Let X=(x_1, x_2…x_n), Y=(y_1, y_2…y_N) be two n-dimension binary vectors and $\mathrm{s}_\mathrm{X}=\sum_{i=1}^{n} x_i$. Function g(X,Y) is defined by equation (6).

$$(\mathrm{X},\mathrm{Y})= \mathrm{s}_\mathrm{X} - \mathrm{X}'\cdot \mathrm{Y} \tag{6}$$

Here X' is derived vector of X and X'·Y is the inner product of vector X' and Y.

Theorem 1. Let W_1, W_2…W_H be all sample binary vectors for the learning of a CC4 network and W be the weight matrix of hidden layer of the CC4. Suppose that W_i' be the derived vector of W_i, i=1,2…H. W=$\begin{pmatrix} (W_1')^T & (W_2')^T & \cdots & (W_H')^T \\ r-s_{W_1}+1 & r-s_{W_2}+1 & \cdots & r-s_{W_H}+1 \end{pmatrix}$.

Lemma 1. Let X=(x_1, x_2…x_n), Y=(y_1, y_2…y_N) be two n-dimension binary vectors. g(X,Y)$\geq$0 and g(X,Y)=0 iff X=Y.

Proof.

(a) if X$\neq$Y

$$\because g(X,Y) = s_X - X'\cdot Y$$

$$\therefore g(X,Y) = s_X - \sum_{i=1,x_i=1}^{n} y_i + \sum_{i=1,x_i=0}^{n} y_i$$

$$\because s_X - \sum_{i=1,x_i=1}^{n} y_i = \sum_{i=1,x_i=1}^{n} (1 - y_i) \geq 0,\ \sum_{i=1,x_i=0}^{n} y_i \geq 0, y_i \geq 0, 1 - y_i \geq 0$$

$$\because X \neq Y$$

$\therefore s_X - \sum_{i=1,x_i=1}^{n} y_i \ge 0$ and $\sum_{i=1,x_i=0}^{n} y_i \ge 0$ could not be satisfied at same time.

$\therefore g(X,Y) > 0$

(b) if X=Y,

$$\because \mathrm{g(X, Y)} = \mathrm{g(X, X)} = s_X - \sum_{i=1,x_i=1}^{n} x_i + \sum_{i=1,x_i=0}^{n} x_i = s_X - s_X$$

$\therefore g(X,Y) = 0$

(c) if g(X,Y)=0

$\because g(X,Y) = s_x - X^{'} \cdot Y$

$$\therefore g(X,Y) = s_X - \sum_{i=1,x_i=1}^{n} y_i + \sum_{i=1,x_i=0}^{n} y_i$$

$$\because s_X - \sum_{i=1,x_i=1}^{n} y_i = \sum_{i=1,x_i=1}^{n} (1-y_i) \ge 0, \sum_{i=1,x_i=0}^{n} y_i \ge 0, y_i \ge 0, 1-y_i \ge 0 \text{ ,g(X,Y)=0}$$

$$\therefore y_i = \begin{cases} 1 & \text{if } \mathrm{x_i} = 1 \\ 0 & \text{if } \mathrm{x_i} = 0 \end{cases} \text{,i=1,2...n}$$

According to (a), (b) and (c), the conclusion is true. #

Lemma 2. Let X=(x_1, x_2...x_n), Y=(y_1, y_2...y_N) be two n-dimension binary vectors. g(X,Y)=g(Y,X).

Proof.

$$\because g(X,Y) - g(Y,X) = s_X - X^{'} \cdot Y - s_Y + Y^{'} \cdot X$$

$$\therefore g(X,Y) - g(Y,X) = s_x - \sum_{i=1}^{n} x_i^{'} y_i - s_Y + \sum_{i=1}^{n} x_i y_i^{'}$$

$$\therefore g(X,Y) - g(Y,X) = \sum_{i=1,x_i=1}^{n} (1-y_i) + \sum_{i=1,x_i=0}^{n} y_i - \sum_{i=1,y_i=1}^{n} (1-x_i) - \sum_{i=1,y_i=0}^{n} x_i$$

$$\therefore g(X,Y) - g(Y,X) = \sum_{i=1,x_i=1,y_i=0}^{n} 1 - \sum_{i=1,x_i=0,y_i=1}^{n} 1 + \sum_{i=1,x_i=0,y_i=1}^{n} 1 - \sum_{i=1,x_i=1,y_i=0}^{n} 1 = 0$$

$\therefore g(X,Y) = g(Y,X)$ #

Lemma 3. Let X=(x_1, x_2...x_n), Y=(y_1, y_2...y_N), Z=(z_1, z_2...z_n) be two n-dimension binary vectors. g(X,Y)+g(Y,Z)≥g(X,Z)

Proof.

$$\because g(X,Y) + g(Y,Z) - g(X,Z) = s_X - X^{'} \cdot Y + s_Y - Y^{'} \cdot Z - s_X + X^{'} \cdot Z$$

$$\therefore g(X,Y) + g(Y,Z) - g(X,Z) = s_Y + \sum_{i=1}^{n} (-x_i^{'} y_i - y_i^{'} z_i + x_i^{'} z_i)$$

$$\therefore g(X,Y)+g(Y,Z)-g(X,Z)=\sum_{i=1,x_i=1,y_i=1}^{n}0+\sum_{i=1,x_i=0,y_i=1}^{n}(2-2z_i)+\sum_{i=1,y_i=0}^{n}(z_i+x_i^{'}z_i)\geq 0$$

$\therefore \mathrm{g(X,Y)}+\mathrm{g(Y,Z)}\geq \mathrm{g(X,Z)}$ #

According to lemma1~lemma3, g(X,Y) is a generalized distance function for n-dimension binary vector.

Lemma 4. Suppose W_1, W_2...W_H be all sample binary vectors for the learning of a CC4 and X is a probe binary vector. If $g(W_i,X)\leq r$, the output of i*th* hidden neuron of the CC4 is 1. Here, *r* is the generalization radius of the CC4 and $1\leq i\leq H$.

Proof.

According to assumptions of this lemma and theorem1, W, the weight matrix of hidden layer of the CC4 can be denoted as $\begin{pmatrix} (W_1^{'})^T & (W_2^{'})^T & \dots & (W_H^{'})^T \\ r-s_{W_1}+1 & r-s_{W_2}+1 & \dots & r-s_{W_H}+1 \end{pmatrix}$.

$\because$ X is a probe vector.

$\therefore$ The input of the CC4 is (X,1) according to the topology of CC4 network.

$\therefore$The input of the i*th* hidden neuron of the CC4 is $X(W_i^{'})^T+r-s_{w_i}+1$.

$\because$*f* is the transfer function of neurons in the CC4.

$\therefore$The output of the i*th* hidden neuron of the CC4 is $f(X(W_i^{'})^T+r-s_{w_i}+1)$.

$\because$ $g(W_i,X)\leq r$

$\therefore s_{W_i}-W_i^{'}\cdot X<r+1$

$\therefore X(W_i^{'})^T+r-s_{w_i}+1>0$

$\therefore f(X(W_i^{'})^T+r-s_{w_i}+1)=1.$ #

Lemma 5. Suppose W_1, W_2...W_H be all sample binary vectors for the learning of a CC4 and X is a probe binary vector. If the output of i*th* hidden neuron of the CC4 is 1, $g(W_i,X)\leq r$. Here, *r* is the generalization radius of the CC4 and $1\leq i\leq H$.

Proof.

According to assumptions of this lemma and theorem1, W, the weight matrix of hidden layer of the CC4 can be denoted as $\begin{pmatrix} (W_1^{'})^T & (W_2^{'})^T & \dots & (W_H^{'})^T \\ r-s_{W_1}+1 & r-s_{W_2}+1 & \dots & r-s_{W_H}+1 \end{pmatrix}$.

$\because$ X is a probe vector.

$\therefore$ The input of the CC4 is (X,1) according to the topology of CC4 network.

$\therefore$The input of the i*th* hidden neuron of the CC4 is $X(W_i^{'})^T+r-s_{w_i}+1$.

$\because$*f* is the transfer function of neurons in the CC4 network.

$\therefore$The output of the i*th* hidden neuron of the CC4 is $f(X(W_i^{'})^T+r-s_{w_i}+1)$.

$\therefore$ $f(X(W_i^{'})^T+r-s_{w_i}+1)=1$

$\therefore X(W_i^{'})^T+r-s_{w_i}+1>0$

$\because s_{W_i} - X(W_i^{'})^T$ is an integer.

$\therefore s_{W_i} - W_i^{'} \cdot X \le r$

$\therefore$ g(W_i,X) $\le r$. #

Theorem 2. Suppose W_1, $W_2 \ldots W_H$ be all sample binary vectors for the learning of a CC4 and X is a probe binary vector. The output of *ith* hidden neuron of the CC4 is 1 iff g(W_i,X) $\le r$. Here, *r* is the generalization radius of the CC4 and $1 \le \mathrm{i} \le \mathrm{H}$.

Proof.

According to lemma4 and lemma5, the conclusion is obvious. #

4 Discussion

Let W_1, $W_2 \ldots W_H$ be all sample binary vectors for the training of a CC4 and W be the weight matrix of hidden layer. W should be $\begin{pmatrix} (W_1^{'})^T & (W_2^{'})^T & \cdots & (W_H^{'})^T \\ r - s_{W_1} + 1 & r - s_{W_2} + 1 & \cdots & r - s_{W_H} + 1 \end{pmatrix}$. Here *r* is generalization radius of the CC4. According to the process of lemma5 proved, W can be valued as $\begin{pmatrix} (W_1^{'})^T & (W_2^{'})^T & \cdots & (W_H^{'})^T \\ r - s_{W_1} + \varepsilon & r - s_{W_2} + \varepsilon & \cdots & r - s_{W_H} + \varepsilon \end{pmatrix}$, here $\varepsilon \in (0,1]$. Because binary vector is the input and output of the CC4, $W = \begin{pmatrix} (W_1^{'})^T & (W_2^{'})^T & \cdots & (W_H^{'})^T \\ r - s_{W_1} + 1 & r - s_{W_2} + 1 & \cdots & r - s_{W_H} + 1 \end{pmatrix}$ is better than $W = \begin{pmatrix} (W_1^{'})^T & (W_2^{'})^T & \cdots & (W_H^{'})^T \\ r - s_{W_1} + \varepsilon & r - s_{W_2} + \varepsilon & \cdots & r - s_{W_H} + \varepsilon \end{pmatrix}$.

References

1. Shu, B., Subhash Kak, S.: A Neural Network-based Intelligent Metasearch Engine. Information Sciences, **120** (1999),: 1-11
2. Venkat, N., Gudivada, V.V., Raghavan, Grosky, W.I., Kasanagott, R.: Information Retrieval on the World Wide Web. IEEE Internet Computing, **1** (1997) 59-68
3. Chen, E., Zhang, Z., Wang, X.: An Extended Corner Classification Neural Network Based Document Classification Approach. Journal of Software, (2002) 871-878
4. Tang, K.W., Kak, S.C.: A New Corner Classification Approach to Neural Network Training. Circuits Systems Signal Processing, **17** (1998) 459-469
5. Raina. P.: Comparison of Learning and Generalization Capabilities of the Kak and the Back Propagation Algorithms. Information Sciences, **81** (1994) 261-274
6. Kak, S.: New Algorithms for Training Feed Forward Neural Networks, Pattern Recognition Letters, **15** (1994) 295-298
7. SKak, S.: On Generalization by Neural Networks. Information Sciences, **111** (1998) 293-302

Chunk Incremental LDA Computing on Data Streams

Shaoning Pang[1], Seiichi Ozawa[2], and Nikola Kasabov[1]

[1] Knowledge Engineering & Discover Research Institute
Auckland University of Technology, Private Bag 92006, Auckland 1020, New Zealand
{spang,nkasabov}@aut.ac.nz
[2] Graduate School of Science and Technology, Kobe University
Kobe 657-8501, Japan
ozawa@eedept.kobe-u.ac.jp

Abstract. This paper presents a constructive method for deriving an updated discriminant eigenspace for classification, when bursts of new classes of data is being added to an initial discriminant eigenspace in the form of random chunks. The proposed Chunk incremental linear discriminant analysis (I-LDA) can effectively evolve a discriminant eigenspace over a fast and large data stream, and extract features with superior discriminability in classification, when compared with other methods.

1 Introduction

Linear Discriminant Analysis(LDA), also known as Fisher Discriminant Analysis-FDA, seeks directions for efficient discrimination, while another technique, principle component analysis (PCA)[3], seeks directions efficient for representation. The typical implementation of these two techniques assumes that a complete dataset for training is given in advance, and learning is carried out in one batch. However, when we conduct LDA/PCA learning over datasets in real-world application, we often confront difficult situation where a complete set of training samples is not given in advance. Actually in most cases, data is divided into many chunks and is being presented as a data stream.

One-pass incremental learning gives a solution to the above problem. In this learning scheme, a system must acquire knowledge with a single presentation of the training data, and retaining the knowledge acquired in the past without keeping a large number of training samples. To achieve this, Peter M Hall etc. [1] proposed Incremental PCA (IPCA) based on the updating of covariance matrix through a residue estimating procedure. Later, he improved his method by proposing a method of merging and splitting eigenspace models[2] that allows a chunk of new samples to be learned in a single step.

Motivated by the IPCA, we propose here an incremental linear discriminant analysis (I-LDA) for classification. A difficulty for incremental LDA modelling, compared with previous IPCA modelling, is that all class data of a complete training dataset may not be presented at every incremental learning stage. The number of classes presented at each learning stage might be very random in

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 51–56, 2005.

real life. In this paper, we derive a solution of I-LDA updating discriminant eigenspace while bursts of new class data is coming in at different times.

2 Chunk Incremental LDA

Let us assume that N training samples$\boldsymbol{X} = \{\boldsymbol{x}_i\}, (i = 1, ..., N)$ in M classes have been presented so far. According to definition [3], a discriminant eigenspace, also called fisherspace, model can be represented by a discriminant eigenspace model as, $\Omega = (\boldsymbol{Sw}, \boldsymbol{Sb}, \bar{\boldsymbol{x}}, N)$. $\boldsymbol{U}$ is a $n \times n$ matrix whose columns correspond to the discriminant eigenvectors. $\boldsymbol{Sw}$ is within class scatter matrix, and $\boldsymbol{Sb}$ is between scatter matrix. The traditional LDA works in a batch way assuming that the whole dataset is given in advance, and is trained in one batch. We called it as Batch LDA in this paper. However, in a streaming environment, new samples are being presented continuously in the form of random chunks. The addition of these new samples will lead to the changes of the original mean vector $\bar{\boldsymbol{x}}$, $\boldsymbol{Sw}$, as well as $\boldsymbol{Sw}$, therefore the whole discriminant eigenspace model Ω should be updated.

Suppose that the new samples are acquired in a chunk way, $Y_0, Y_1, ...$, to infinity. As shown in Fig 1, at each time point t, a chunk of samples instead of just one sample $Y = \{\boldsymbol{y}_1, .., \boldsymbol{y}_L\}$ are acquired, where L is the number of samples in one chunk, it is a random positive integer and $L \geq 1$.

In this sense, we need to derive a solution to the following problem. Let X and Y be two sets observations, where X is the presented observation set, and Y is a set of new observations. Let their discriminant eigenspace models be $\Omega = (\boldsymbol{Sw}, \boldsymbol{Sb}, \bar{\boldsymbol{x}}, N)$ and $\Psi = (\boldsymbol{Sw}_y, \boldsymbol{Sb}_y, \bar{\boldsymbol{y}}, L)$, respectively. As in IPCA[2], the above updating problem here is to compute the new fisherspace model $\Phi = (\boldsymbol{Sw}', \boldsymbol{Sb}', \bar{\boldsymbol{x}}', N + L)$ using fisherspace model Ω and Ψ.

Without loss of generality, we can assume l_c of L new samples belong to class c, thus $n'_c = n_c + l_c$, $N + L = \sum_{c=1}^{M} n'_c = \sum_{c=1}^{M} (n_c + l_c)$ and $\bar{\boldsymbol{x}}'_c = \frac{1}{n_c + l_c}(n_c \bar{\boldsymbol{x}}_c + l_c \bar{\boldsymbol{y}}_c)$, where $\bar{\boldsymbol{y}}_c$ is the mean of new samples in class c.

The updated mean is: $\bar{\boldsymbol{x}}' = (N\bar{\boldsymbol{x}} + L\bar{\boldsymbol{y}})/(N + L)$, where $\bar{\boldsymbol{y}} = \frac{1}{L}\sum_{j=1}^{L} \boldsymbol{y}_j$. The updated $\boldsymbol{Sb}$ matrix is:

$$\boldsymbol{Sb}' = \sum_{c=1}^{M} n'_c (\bar{\boldsymbol{x}}'_c - \bar{\boldsymbol{x}}')(\bar{\boldsymbol{x}}'_c - \bar{\boldsymbol{x}}')^T. \tag{1}$$

The updated $\boldsymbol{Sw}$ matrix can be proofed as:

$$\boldsymbol{Sw}' = \sum_{c=1}^{M} \boldsymbol{\Sigma}'_c \tag{2}$$

$$\boldsymbol{\Sigma}'_c = \Sigma_c + \frac{n_c l_c^2}{(n_c + l_c)^2}(\boldsymbol{D}_c) + \frac{n_c^2}{(n_c + l_c)^2}(\boldsymbol{E}_c) + \frac{l_c(l_c + 2n_c)}{(n_c + l_c)^2}(\boldsymbol{F}_c) \tag{3}$$

where the second term $\boldsymbol{D_c}$ is the scatter matrix of the new sample class mean vector $\overline{\mathbf{y}}_i$ around the mean vector $\bar{\boldsymbol{x}}_c$ of class c,

$$\boldsymbol{D_c} = (\bar{\boldsymbol{y}}_c - \bar{\boldsymbol{x}}_c)(\bar{\boldsymbol{y}}_c - \bar{\boldsymbol{x}}_c)^T. \tag{4}$$

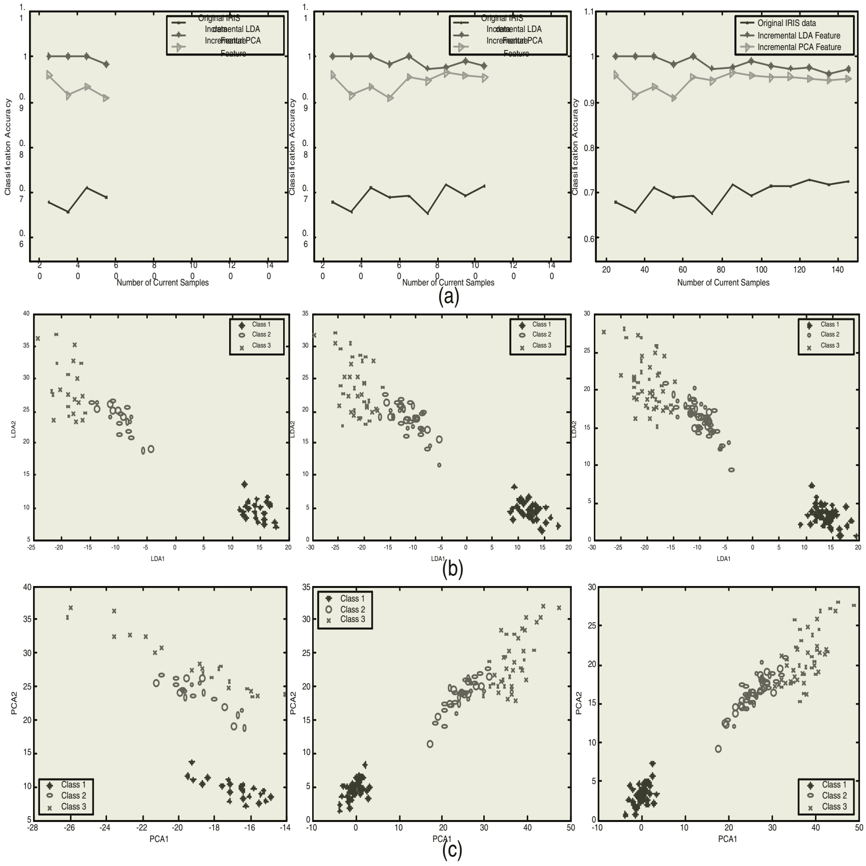


Fig. 1. The variation of I-LDA versus IPCA on (a) classification accuracy, (b) IPCA projection of Iris data, and (c) I-LDA projection of Iris

The third term $\boldsymbol{E_c}$ is the scatter matrix of the new sample mean vector $\overline{\mathbf{y}}_i$ around the mean vector $\bar{\boldsymbol{x}}_c$ of class c,

$$\boldsymbol{E_c} = \sum_{j=1}^{l_c} (\boldsymbol{y}_{cj} - \bar{\boldsymbol{x}}_c)(\boldsymbol{y}_{cj} - \bar{\boldsymbol{x}}_c)^T. \tag{5}$$

The fourth term $\boldsymbol{F_c}$ is a within-class scatter matrix of the new samples,

$$\boldsymbol{F_c} = \sum_{j=1}^{l_c} (\bar{\boldsymbol{y}}_{cj} - \bar{\boldsymbol{y}}_c)(\bar{\boldsymbol{y}}_{cj} - \bar{\boldsymbol{y}}_c)^T. \tag{6}$$

In addition, we also can assume l_{M+1} of L new samples belong to class $M+1$ without loss of generality. In this case, the updated between-class matrix Eq.(5) can be rewritten as

Table 1. The comparison results of I-LDA, Batch LDA, IPCA and Batch IPCA on the classification at the final incremental learning stage for 8 UCL datasets

	No. Eig.	Incremental LDA	Batch LDA	Incremental PCA	Batch PCA
Name		Acc. %	Acc. %	Acc. %	Acc. %
Iris	2	98.0	98.0	93.3	93.3
Liver-disorder	3	62.6	62.6	58.8	58.8
Vehicle	9	75.4	75.4	57.8	57.8
Glass	6	67.7	67.3	51.8	51.8
Wine	7	96.6	96.6	87.6	87.64
Segmentation	6	83.9	80.5	81.4	80.0
Vowel	10	60.9	59.8	57.9	60.4
Sonar	6	81.2	81.2	73.5	73.5

$$\begin{aligned} \boldsymbol{Sb}' &= \sum_{c=1}^{M} n'_c(\bar{\boldsymbol{x}}_c - \bar{\boldsymbol{x}}')(\bar{\boldsymbol{x}}_c - \bar{\boldsymbol{x}}')^T + l_{M+1}(\bar{\boldsymbol{y}} - \bar{\boldsymbol{x}}')(\bar{\boldsymbol{y}} - \bar{\boldsymbol{x}}')^T \\ &= \sum_{c=1}^{M+1} n'_c(\bar{\boldsymbol{x}}_c - \bar{\boldsymbol{x}}')(\bar{\boldsymbol{x}}_c - \bar{\boldsymbol{x}}')^T \end{aligned} \tag{7}$$

where n'_c is the number of samples in class c after Y is presented. When $c = M+1$, $n'_c = l_{M+1}$; else $n'_c = n_c + l_c$. Also, the updated within-class matrix can be written as,

$$\boldsymbol{Sw}' = \sum_{c=1}^{M} \boldsymbol{\Sigma}_c + \boldsymbol{\Sigma}_{M+1} = \sum_{c=1}^{M+1} \boldsymbol{\Sigma}_c{}' \tag{8}$$

where $\boldsymbol{\Sigma}_{M+1} = \sum_{y \in \{y_c\}} (\boldsymbol{y} - \bar{\boldsymbol{y}}_c)(\boldsymbol{y} - \bar{\boldsymbol{y}}_c)^T$.

3 Results and Discussions

We have examined the efficiency and accuracy of our incremental LDA methods for the classification of datastreams. Particularly, we are interested in evaluating the discriminability of I-LDA with comparison to batch LDA and incremental PCA (IPCA) [1, 2].

For every test, we construct an initial feature space (eigenspace) using 10% of total samples, in which at least two classes data are ensured to be included according to the definition of Eq. (2). These training samples are used for calculating eigenvectors and eigenvalues through conventional LDA/PCA. The remaining training data is enumerated into a number of chunks without considering the chunk size and number of classes in each chunk. Since the events of data arriving in the above incremental learning may not happen at regular time intervals, we use the term *learning stage* instead of the usual time scale. Here, the number of learning stages is equivalent to the number of samples that has been learned by incremental models. In addition, for the convenience of illustration, we also define the percentage of samples presented so far at current stage to measure the *progress* of incremental learning.

The database we used consists of 8 standard datasets selected from UCI Machine Learning Repository[4], where each dataset has its features 100% percent of continuous/integer values, and no missing value. Over 8 UCI datasets, we carried out the incremental learning described above to test the discriminablity of I-LDA for classification.

As an example, Fig. 1 shows a time course of the above incremental learning over Iris data with 3 learning stages, whose *progress* of incremental learning are at 30%, 65% and 100% respectively, in which (a) gives the variation of classification accuracy; (b) and (c) are IPCA projection and I-LDA projection of Iris data onto the first 2 eigenvectors, respectively. Here, I-LDA is compared with IPCA on the variation of classification accuracy and feature distribution with the original data as references. As can be seen in Fig. 1(a), I-LDA is leading the classification accuracy at most stages of incremental learning, particularly it achieves a better final classification accuracy than IPCA. This superiority on classification can be also clearly reflected from the discrimination differences between corresponding I-LDA projections in Fig. 1(b) and IPCA projections in Fig. 1(c).

Table 1 presents the comparison results of I-LDA, Batch LDA, IPCA and batch IPCA on the classification at the final incremental learning stage for all 8 UCL datasets, where the number of Eigenvector (denoted as No. Eig.) specifies the dimension of LDA and PCA eigenfeatures used in classification. As can be seen, when using the same number of eigenvectors, the classification accuracies due to I-LDA and LDA are very similar. In most cases, they are exactly the same. So for IPCA and PCA, but their accuracies are obviously lower than those of I-LDA and LDA. It suggests that discriminant ability of I-LDA is equivalent to that of LDA, and is better than IPCA.

To check the similarity of the two discriminant eigenspaces constructed by I-LDA and Batch LDA, the inner products between discriminant eigenvectors obtained by I-LDA and LDA are evaluated. For I-LDA, the discriminant eigenvectors I-LDA are being updated over the learning stages. For Batch LDA, the discriminant eigenvectors are calculated from the whole dataset in a batch mode. Fig. 2 shows typical time courses of inner products $d_1 \sim d_4$ between discriminant eigenvectors obtained by I-LDA and batch I-LDA learning over Iris data. Here, we assume that each eigenvector has an unit length, and the total 4 discriminant eigenvectors of I-LDA and LDA are compared correspondingly in Fig. 4. As seen from Fig. 2, d_1 and d_2 have large values from the beginning of learning stages. Although d_3 and d_4 have some fluctuations during the incremental learning, the values converge to the maximum value at the final stage. Hence, we can observe that I-LDA is updating the discriminant eigenspace gradually as the progress of incremental learning grows, and it can finally construct exactly the same discriminant eigenspace as that of Batch LDA.

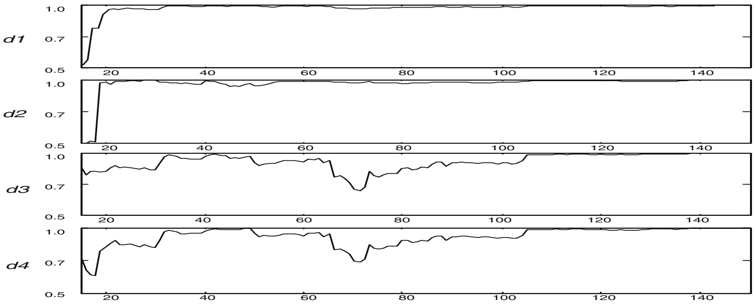


Fig. 2. Typical time courses of inner products $d_1 \sim d_4$ between discriminant eigenvectors obtained by I-LDA and batch LDA as the incremental learning stages proceed

4 Conclusions

Like IPCA, the essence of Incremental linear discriminant analysis (ILDA)should be the incremental updating of the eigen-decomposition. As an alternative solution of ILDA, this paper proposed the method of incremental LDA computing (I-LDA) to derive discriminant eigen-space in a streaming environment without updating the eigen-decomposition. This method is in terms of the incremental updating of the between-class and within-class scatter matrices, thus are able dealing with volumes of data addition.

I-LDA has an equivalent power to batch LDA in terms of discriminability and is very effective for handling bursts of new classes coming in at different times. One limitation of the proposed I-LDA is, when the chunk-size is too large, the memory cost of I-LDA increases. But generally, it is very distinct that the above optimal properties have determined I-LDA as an useful method, when we conduct classification on fast and large data streams.

References

1. Gall, P., Martin, R.: Incremental Eigenanalysis for Classification. Proc. British Machine Vision Conference, **1** 286-295
2. Hall, P., Marshall, D., Martin, R.: Merging and Splitting Eigenspace Models. IEEE Trans. Pattern Analysis and Machine Intelligence, **22** (2000) 1042–1049
3. Aleix M. Martinez, Avinash C. Kak: PCA versus LDA. IEEE Trans. Pattern Analysis and Machine Intelligence, **23** (2001) 228-233
4. http://www.ics.uci.edu/mlearn/MLRepository.html

A Novel Clustering Method Based on SVM*

Jie Li, Xinbo Gao, and Licheng Jiao

School of Electronic Engineering, Xidian Univ., Xi'an 710071, China

Abstract. For the problem of cluster analysis, the objective function based algorithms are popular and widely used methods. However, the performance of these algorithms depends upon the priori information about cluster number and cluster prototypes. Moreover, it is only effective for analyzing data set with the same type of cluster prototypes. For this end, this paper presents a novel algorithm based on support vector machine (SVM) for realizing fully unsupervised clustering. The experimental results with various test data sets illustrate the effectiveness of the proposed novel clustering algorithm based on SVM.

1 Introduction

Cluster analysis is one of multivariant statistical analysis methods, which is also an important branch of unsupervised classification in statistical pattern recognition [1]. In traditional cluster analysis methods, the objective function based clustering algorithms convert the clustering problem into an optimization problem. Due to having profound functional foundation, this type of clustering algorithms become the main research topic of cluster analysis, in which *c*-means algorithm is one of representative algorithms [2]. However, *c*-means algorithm cannot detect clusters in nonlinear sub-spaces, since it assumes the clustering prototypes as points in feature space, called clustering centers. To this end, the clustering prototypes are generalized from points to lines, planes, and shells, and some *c*-means type algorithms are proposed with various prototypes, such as *c*-lines, *c*-planes and *c*-shells algorithms [3–5]. Therefore, these *c*-means type algorithms can perform clustering analysis on data sets with different prototypes.

Although the above *c*-means type algorithms extend the application range of objective function based clustering algorithms, it costs at the increasing of requirements of clustering priori information. Meanwhile, this type of algorithms require that all the prototypes should be with the same form, of course with different parameters. Such requirement limits their practical application. In many fields, the handled data set to be analyzed often contains unknown number of subsets with different prototypes. The number of clusters is difficult to automatically determine, especially in high-dimensional feature space.

In recent years, SVM attracts more and more attentions in the field of machine learning. Based on the statistical learning theory, the SVM has some good

* This project was supported by NFSC (No.60202004) and the key project of Chinese Ministry of Education (No.104173).

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 57–62, 2005.

properties, such as concise mathematical formulation, intuitive geometric explanation and good generalization, which makes it successively applied in the fields of classification and regression. In 1999, Schölkopf[4], Tax and Duin[5] proposed to describe the high dimensional distributed data set with SVM. These algorithms can obtain a set of support vectors on the boundary of the data set. Later, based on these algorithms, Asa presented a SVM clustering algorithm. By taking the obtained support vectors as the boundaries among subsets, the partition of the given data set can be achieved. Unfortunately, Asa's algorithm only outputs the partition of the data set and does not provide the distribution or prototype of each subset. While, in the most practical applications of cluster analysis in data mining, the prototype information is more important than data partition. For this purpose, this paper presents a novel SVM-based clustering algorithm, which can obtain not only the partition information but also the analytic expression of the cluster prototypes.

2 The Searching Algorithm for Support Vectors

Let $X = \{x_1, x_2, \cdots, x_n\}$ denote a set of objects, where $x_i = [x_{i1}, x_{i2}, \cdots, x_{iN}]^T$ denotes the N features of the i-th samples. Using a nonlinear transformation Φ from X to some high dimensional feature-space, we look for the smallest enclosing sphere of radius R.

$$||\Phi(x_j) - \mathbf{a}||^2 \leq R^2 \qquad \forall j \tag{1}$$

where $||\cdot||$ is the Euclidean norm and $\mathbf{a}$ is the center of the sphere. Soft constraints are incorporated by adding slack variables ξ_j.

$$||\Phi(x_j) - \mathbf{a}||^2 \leq R^2 + \xi_j \qquad \xi_j \geq 0 \qquad \forall j \tag{2}$$

To solve this problem we introduce the Lagrangian

$$L = R^2 - \sum_j (R^2 + \xi_j - ||\Phi(x_j) - \mathbf{a}||^2)\beta_j - \sum \xi_j \mu_j + C \sum \xi_j \tag{3}$$

where $\xi_j \geq 0$ and $\mu_j \geq 0$ are Lagrange multipliers, C is a constant. Setting to zero the derivative of L with respect to R, $\mathbf{a}$ and ξ_j, respectively, leads to

$$\sum_j \beta_j = 1 \tag{4}$$

$$\mathbf{a} = \sum_j \beta_j \Phi(x_j) \tag{5}$$

$$\beta_j = C - \mu_j \tag{6}$$

The KKT complementarity conditions of Fletcher[6] result in

$$\xi_j \mu_j = 0 \tag{7}$$

$$(R^2 + \xi_j - ||\Phi(x_j) - \mathbf{a}||^2)\beta_j = 1 \tag{8}$$

It follows from Eq.(8) that the image of a point x_i with $\xi_j \geq 0$ and $\beta_j \geq 0$ lies out-side the feature-space sphere. Eq.(7) states that such a point has $\mu_j = 0$, hence we conclude from Eq.(6) that $\beta_i = C$. This will be called a *bounded support vector* (BSV). A point x_i with $\xi_i = 0$ is mapped to the inside or to the surface of the feature space sphere. If its $0 < \beta_i < C$ then Eq.(8) implies that its image $\Phi(x_i)$ lies on the surface of the feature space sphere. Such a point will be referred to as a *support vector* (SV). SVs lie on cluster boundaries, BSVs lie outside the boundaries, and all other points lie inside them.

Using these relations we may eliminate the variables R, $\mathbf{a}$ and μ_j, turning the Lagrangian into the Wolfe dual form that is a function of the variables β_j:

$$W = \sum_j \Phi(x_j)^2 \beta_j - \sum_{i,j} \beta_i \beta_j \Phi(x_i) \cdot \Phi(x_j) \tag{9}$$

We follow the SV method and represent the dot products $\Phi(x_i) \cdot \Phi(x_j)$ by an appropriate Mercer kernel $K(x_i, x_j)$. We use the Gaussian kernel function.

$$K(x_i, x_j) = e^{-q||x_i - x_j||^2} \tag{10}$$

At each point x we define the distance of its image in feature space from the center of the sphere:

$$R^2(x) = ||\Phi(x) - \mathbf{a}||^2 \tag{11}$$

In view of Eq.(4) and the definition of the kernel we have

$$R^2(x) = K(x, x) - 2 \sum_j K(x_j, x) \beta_j + \sum_{i,j} \beta_i \beta_j K(x_i, x_j) \tag{12}$$

The radius of the sphere is

$$R = \{R(x_i) | x_i \text{ is a support vector}\} \tag{13}$$

3 Class Labelling Algorithm

For partitioning the given data set, the obtained support vectors (SVs) should be labelled with different marks. The classification procedure follows the below principle. If a pair of SVs belongs to different clusters, any path that connects them must exist from the sphere in feature space. Therefore, such a path contains a segment of point y such that $R(y) > R$. On the contrary, if a pair of SVs belongs to the same class, any path that connects them must exist within the sphere in feature space. This leads to the definition of the label l as follows.

$$\begin{cases} l(SV_i) = l(SV_j) & \text{if, for all } y \text{ on the line segment connecting} \\ & SV_i \text{ and } SV_j, R(y) \leq R \\ l(SV_i) \neq l(SV_j) & \text{otherwise} \end{cases} \tag{14}$$

After labelling all the SVs, Let $P_i = [SV_{i1}, SV_{i2}, \cdots, SV_{ii_n}]$, $i = 1, 2, \cdots, c$ be the SVs belonging to the i-th cluster and i_n be the total number of support vectors in this class. The dissimilarity between x_j and the i-th cluster can be defined as follows.

$$d^2(x_j, P_i) = \min\{(x_j - SV_{ig})^T \cdot (x_j - SV_{ig}), \quad g = 1, 2, \cdots i_n\} \tag{15}$$

Therefore, when the sample x_j belongs to the i-th class, we have

$$d^2(x_j, P_i) = \min\{d^2(x_j, P_l), \quad l = 1, 2, \cdots c\} \tag{16}$$

4 The Prototype Analysis Algorithm

In this section, we present a prototype and its parameters estimation algorithm. First, the mean and covariance matrix are computed by statistically analyzing the obtained support vector set of each cluster, P_i.

$$m_i = \frac{1}{i_n}\sum_{j=1}^{i_n} SV_{ij} \qquad C_i = \frac{1}{i_n}\sum_{j=1}^{i_n} SV_{ij}SV_{ij}^T - m_i m_i^T \tag{17}$$

Since C_i is a $N \times N$ symmetry matrix, one can always find N orthogonal eigenvectors of C_i. Let $\lambda_{ik}(k = 1, 2, \cdots, N)$ be the corresponding eigen values, which is monotonously ranked in ascending order. Define a variable *ratio* as

$$ratio_i = \frac{\lambda_{i1}}{\lambda_{iN}} \tag{18}$$

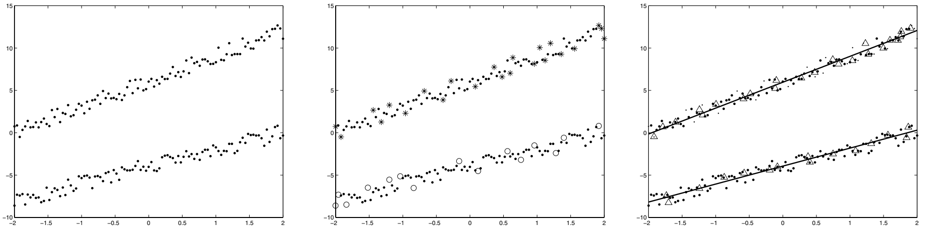
In the case of $ratio \approx 1$, it implies that the long axis of the minimal hype-ellipsoid containing all the i-th class samples is approximately equal to its short axis. That is to say, the subset is distributed with spherical or spherical-shell pattern. The center of the cluster can be obtained by averaging the SVs of this class. To distinguish the spherical and spherical-shelled patterns, one can count the number of samples within a ball with the center at the cluster centroid and the radii of σ. If the counted number is greater than a preset threshold, it implies that the cluster is distributed in spherical pattern.

In the case of $ratio << 1$, it means that this cluster is distributed in a line-shaped pattern. For such cluster, the fuzzy c-means algorithm is employed to partition the samples into c groups, where c corresponds to the number of SVs in this class. The obtained clustering centers will be taken as typical patterns for further polynomial fitting.

5 The Experimental Results

5.1 Experiment on Data Set with Same Type of Prototypes

For the sake of visualization, we generate a set of data with 300 samples in 2D plane, which is composed of 2 sets of linear separable subsets as shown in Fig.1(a). In Fig.1(b), "$*$" and "o" denote the obtained SVs for the two subsets by the SVM respectively. According to Eq.(18), the *retio*s for two subsets equal to 0.0022 and 0.0044 respectively, which implies that both the two subsets are distributed in line-shaped pattern. For each subset, we employ the fuzzy c-means (FCM) algorithm to extract a set of representative vectors with the same number of SVs, which is shown in Fig.1(c) with the mark ”$\triangle$”. The fitting result with the first-order polynomial is presented in Fig.1(c).



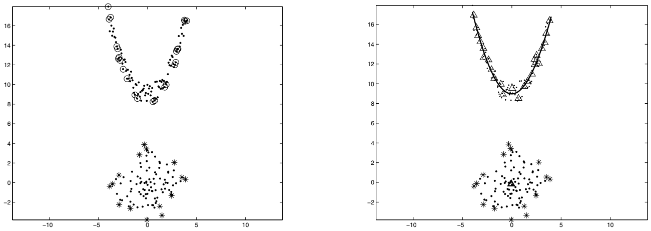
(a) (b) (c)

Fig. 1. The clustering result of the data set with two line-shaped clusters

5.2 Experiment on Data Set with Different Types of Prototypes

In some practical applications in data mining, data set containing different types of prototypes is often encountered. In this experiments, we use 2 types data sets with different types of prototypes as test bed.

Fig.2(a) presents a mixed data set with a spherical cluster and a parabola-shaped cluster. By analysis, we get the *ratio*s of the two obtained clusters by the SVM are 0.9155 and 0.8757 respectively. The further analysis indicates that the cluster with is a spherical subset, while the cluster with $ratio = 0.8757$ is a shell-shaped subset. By using polynomial fitting procedure, we get the analytical expression of $y = 0.5152x^2 - 0.039x + 9.023$, which is obvious a parabola.

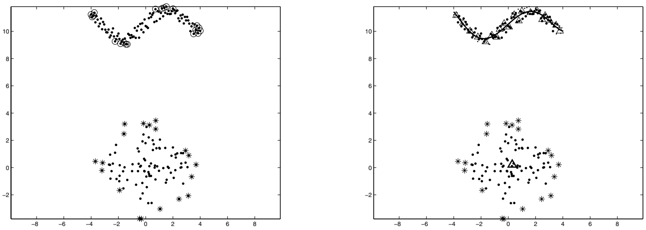


(a) (b)

Fig. 2. The clustering result of the mixed data set with a spherical cluster and a parabola-shaped cluster

Fig.3(a) gives a data set with a spherical cluster and an arbitrary shaped cluster. With the proposed SVM clustering, we obtain two clusters with $ratio =$

0.8993 and $ratio = 0.1175$. The further analysis shows that the former is spherical cluster and the later is an arbitrary shaped cluster with analytical expression as follows, $y = 0.0045x^5 + 0.0038x^4 - 0.1456x^3 - 0.0529x^2 + 1.0262x + 10.6061$.



(a) (b)

Fig. 3. The clustering result of the mixed data set with spherical cluster and an arbitrary shaped cluster

6 Conclusions

This paper presents a novel SVM-based clustering algorithm. The experimental results demonstrate that it can detect the clustering structures effectively. Moreover, it does not depend on the priori information about the cluster number and cluster prototypes. Especially, the proposed clustering algorithm can output the analytical expression of the cluster prototype automatically. That is to say, compared with the available clustering algorithms, the proposed one is a more unsupervised learning algorithm, which will facilitate the practical applications.

References

1. He, Qing: Advance of the Theory and Application of Fuzzy Clustering Analysis. Fuzzy System and Fuzzy Mathematics, **12** (1998) 89–94
2. Gao, Xinbo: Studies of Optimization and Applications of Fuzzy Clustering Algorithm, Doctoral Dissertation, Xidian University, Xi'an, China (1999)
3. Li, Jie, Gao, Xinbo, Jiao, Licheng: A Novel Clustering Algorithm Based on Immune Network with Limited Resource. Proc. of AI 2004, **LNAI(3339)** (2004) 319–331
4. Scholkopf, R.C., Williamson, B., A.J. Smola, J.: Support Vector Method for Novelty Detection. in Advances in Neural Information Processing Systems 12: Proc. of the 1999 Conf., Sara A. Solla, Todd K. Leen and Klaus-Robert Muller eds., (2000)
5. Tax, D.M.J., Duin, R.P.W.: Support Vector Domain Description. Pattern Recognition Letters, **(20)** (1999) 1991–1999
6. Fletcher, R.: Practical Methods of Optimization. Wiley-Interscience, Chichester, (1987)

Clustering High-Dimensional Data Using Growing SOM*

Junlin Zhou and Yan Fu

School of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, Sichuan 610054, China
{jlzhou,fuyan}@uestc.edu.cn

Abstract. The self-organizing map (SOM) is a very popular unsupervised neural-network model for analyzing of high-dimensional input data as in scientific data mining applications. However, to use the SOM, the network structure must be predetermined, this often leads constrains on potential applications. When the network is unfit to the data model, the resulting map will be of poor quality. In this paper, an intuitive and effective SOM is proposed for mapping high-dimensional data onto the two-dimensional SOM structure with a growing self-organizing map. In the training phase, an improved growing node structure is used. In the procedure of adaptive growing, the probability distribution of sample data is also a criterion to distinguish where the new nodes should to be added or deleted besides the maximal quantization error (mqe) of a unit. The improved method is demonstrated on a data set with promising results and a significantly reduced network size.

1 Introduction

The Self-Organizing Map (SOM in short), originated by Kohonen[1], is an unsupervised, competitive learning algorithm that maps high dimensional data onto a discrete network structure of lower dimensions. Since the mapping of data from a high-dimensional space to a two or three-dimensional grid makes the interrelations among the data points perceptible, it provides a better insight into the data structure and clustering tendency. This mapping retains the relationship between input data as faithfully as possible, thus describing a topology-preserving representation of input similarities in terms of distances in the output space. It is then possible to visually identify clusters on the map. This feature capability has made the SOM an important tool in a wide range of applications such as data mining, and or more generally, pattern recognition and knowledge acquisition.

However, some difficulties in SOM utilization remained largely untouched, even though a large number of research papers on applications of the SOM were presented over the years[2]. First, the SOM uses a fixed network architecture

* This work was supported by National Science Foundation of China under Grant 10476006.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 63–68, 2005.

in terms of number and arrangement of neural processing elements, which has to be defined prior to training. In most cases, the user does not have much knowledge of the inherent data structure. So it is difficult to predefine a proper layout of the network. This often leads to a significant limitation on potential applications[3]. It is very likely that a predetermined size of the network is either too small or too big. In either case, the resulting map will be of poor quality. Thus, it certainly is worth considering neural-network models that determine the number and arrangement of units during their unsupervised training process. We refer to[4] for recently proposed models that are based on the SOM, yet allow for adaptation of the network architecture during training.

Another disadvantage of the fixed grid is that though data vectors are mapped to corresponding best-matching neurons, it is usually difficult to provide much information about the global distribution of the data by observing the raw map. In this paper, a growing SOM based on probability distribution is proposed to help overcome the constraints imposed by the traditional SOM.

2 Related Work

The principal goal of the SOM is to transform an incoming signal pattern of arbitrary dimension into a one or two-dimensional discrete map, and perform this transformation adaptively in topologically ordered fashion. There are three essential processes involved in the formation of the self-organizing map as below.

Competition. For each input vector, the neurons in the map compute their respective values of a discriminant function. The particular neuron with the largest value of discriminant function is declared winner of the competition. Let m denote the dimension of the input (data) space. Let an input pattern selected at random form the input space be denoted by $\overline{x} = [x_1, x_2, ..., x_m]$. Let the synaptic weight vector of neuron j be denoted by $\overline{w} = [w_1, w_2, ..., w_m]^T, j = 1, 2, ..., l$ where l is the total number of neurons in the map. To find the best match of the input vector $\overline{x}$ with the synaptic weight vectors $\overline{w_j}$, use the best matching criterion, based on maximizing the inner product $\overline{w_j}^T \cdot \overline{x}$, is mathematically equivalent to minimizing the Euclidean distance between vectors $\overline{x}$ and $\overline{w_j}$. If we use the index $i(\overline{x})$ to identify the neuron that best matches the input vector $\overline{x}$, we may then determine $i(\overline{x})$ by applying the condition $i(\overline{x}) = \arg\min_j \|\overline{x} - \overline{w_j}\|, j = 1, 2, ..., l$. The particular neuron i that satisfies this condition is called the winning neuron for the input vector $\overline{x}$.

Cooperation. The winning neuron determines the spatial location of a topological neighborhood of excited neurons, thereby providing the basis for cooperation among such neighboring neurons. Let h_{ij} denote a typical topological neighborhood centered on the winning neuron i, and j denote a typical neuron of a set of excited neurons around winning neuron i. Let d_{ij} denote the lateral distance between winning neuron i and excited neuron j. A typical choice of h_{ij} that satisfies these two requirements is the Gaussian function

$$h_{j,i(\overline{x})}(n) = e^{(\frac{d_{j,i}^2}{2\sigma(n)^2})}, n = 0, 1, 2, ... \quad (1)$$

The lateral distance d_{ij} between winning neuron i and the excited neuron j is defined as $d_{j,i}^2 = \|\overline{r_j} - \overline{r_i}\|^2$. Where the discrete vector $\overline{r_j}$ and $\overline{r_i}$ defines the position of excited neuron j and i. A popular choice for $\sigma(n)$ which depends on discrete time n is the exponential decay described as below

$$\sigma(n) = \sigma_0 e^{\frac{n}{\tau_1}}, n = 0, 1, 2, ... \tag{2}$$

where σ_0 is the value of σ at the initiation of the SOM algorithm and τ_1 is a time constant through the whole learning process.

Synaptic adaptation. It enables the excited neurons (weight vectors) to increase their individual values of the discriminant function in relation to the input pattern through suitable adjustments applied to their synaptic weights. By using discrete-time formalism, given the synaptic weight $\overline{w_j}(n)$ of neuron j at time n, the updated weight $\overline{w_j}(n+1)$ at time $n+1$ is then defined by

$$\overline{w_j}(n+1) = \overline{w_j}(n) + \alpha(n) h_{j,i(\overline{x})}(n)(\overline{x} - \overline{w_j}(n)) \tag{3}$$

The $\alpha(n)$ is the learning-rate parameter, it should be time varying as indicated in Equation above for stochastic approximation. In particular, it should start at an initial value α_0, and then decrease gradually with increasing n time. It is shown by

$$\alpha(n) = \alpha_0 e^{-\frac{n}{\tau_2}}, n = 0, 1, 2, ... \tag{4}$$

where τ_2 is another time constant of the SOM algorithm.

3 The Proposed GSOM

The GSOM is based on 2-dimensional Self-Organizing Maps, but the nodes are connected in a triangular way. These two-dimensional structures will be used to describe clusters of patterns in the n-dimensional vector space of real numbers R^n. For this reason every nodes c has an n-dimensional model vector $pos(c)$ indicating its virtual position in R^n. Now let $p(x)$ be an unknown probability distribution on R^n. Although $p(x)$ is not known explicitly it shall be possible to generate arbitrarily many sample vectors according to $p(x)$. By evaluating those sample vectors clusters can be detected as follows:

First, for every input vector x the nearest nodes is determined. This is the winner node satisfying the relation

$$\|pos(winner) - x\| \leq \|pos(c) - x\|, (c \in S) \tag{5}$$

Each time when a node is winner, we add the distance between this node and the input vector to an "error" variable attached to the node. Then we can determine the amount of error caused by every node over the time. A high error value indicates that near the position of the corresponding node in R^n the density of nodes is comparatively low. Therefore after the redistribution we search for the node bs("black sheep") with maximum error value. We then determine the farthest direct neighbor of bs, for example a node f, and split the edge which connects both. A new node c_{new} is then inserted halfway between bs and f in R^n. Its model vector is initialized as

$$pos(c_{new}) = \frac{1}{2}(pos(bs + pos(f))) \tag{6}$$

Two different cases can occur depending on whether the split edge lies inside the structure S or on the rim of S. The basic growing process is illustrated in Fig.1.

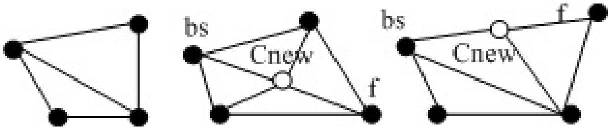


Fig. 1. Insertion of a new node between bs and f

The error variable of c_{new} is initialized as

$$err(c_{new}) = \frac{1}{d+1}\sum_{i=1}^{d} err(neighbor_i) \tag{7}$$

Thereby d is the number of direct neighbors of c_{new} and $neighbor_i$ is the $i - th$ of these nodes. It is desirable to keep the total error of the involved nodes constant. Therefore the error values of the direct neighbors of c_{new} are divided by $(d+1)/d$. Since the insertion of a new node diminishes the error of the direct neighbor nodes they will less likely become the "black sheep" the next time. In the long run this will lead to a node density which approximates the probability density of $p(x)$. Since in the extended structure each node is directly or indirectly connected to others. To detect clusters it is necessary to introduce the removal of nodes. Clusters of similar vectors are separated by regions with probability density. Consequently an obvious criterion for a node to be removed would be, that it has a position in an area of the R^n, where the probability density is very little or equal with zero. The general idea is that the longer a node was not winner, the more likely it lies in an area of R^n where the probability density of $p(x)$ is zero.

Let n now be the current number of nodes in the structure. Then assume, that $p(x)$ is a uniform distribution in a sub area B of R^n and the nodes are uniformly distributed over B. Let c be an arbitrary node with a position somewhere in B. The event "c is winner for the current input vector" has the probability $p_{c\_winner} = 1/n$. This event does not occur with a probability $p_{\overline{c\_winner}} = 1 - 1/n$. We want to remove nodes, which were not winner for a "sufficiently" large number k_{remove} of successive input vectors. Considering that after every $n_{distribution}$ input vectors a new node is created, the probability that a node lying in B has not been winner for k_{remove} input vector is

$$p_1 = (1 - \frac{k_{remove}}{n_{distribution} \cdot n}) \tag{8}$$

Then the probability that each of the currently n nodes lying in B was at least one time winner during the last k_{remove} input vectors is $p_2 = (1 - p_1)^n$. To make it improbable that a node in B is removed we have to choose k_{remove} sufficiently large. If we want to be certain to a degree p_{sure} that no node in B is removed, then p_2 must exceed p_{sure}. The following relation must be satisfied: $p_2 > p_{sure}$. With the above definition finally leads to

$$k_{remove} > n_{distribution} \cdot n(1 - (1 - p^{\frac{1}{n}})^{\frac{1}{n_{distribution}+1}}) \tag{9}$$

Then k_{remove} is periodically computed for the current number n of nodes. If the difference between the current input vector number and the number stored with some node exceeds k_{remove} this node and all adjacent edges are removed.

4 Experiments

The work was conducted using the Growing Cell Structure Visualization toolbox [5]. To test the proposed method, the animal data set used was introduced by Ritter and Kohonen.

Table 1. The animals data set

	large				leg		has										likes to			
Name	1	2	3	4	2	4	hair	hoof	mane	fea	sca	str	tail	swi	horn	tusk	hunt	run	fly	dig
boar	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	1	0	1	0	1
dog	0	0	1	0	0	1	1	0	0	0	0	0	1	0	0	0	0	1	0	1
duck	0	1	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0
eagle	0	0	1	0	1	0	0	0	0	1	0	0	0	1	0	0	1	0	1	0
fox	0	0	1	0	0	1	1	0	0	0	0	0	1	0	0	0	1	0	0	0
hen	0	1	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
mouse	1	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	1	0	1
pig	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
snake	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
wolf	0	0	1	0	0	1	1	0	1	0	0	0	1	0	0	0	1	1	0	0

The data set comprises 24 records of different kinds of animals with 20 attributes. Table 1 shows part of them. Each of record can be seen as a high-dimensional vector, after training, the map was partitioned into four regions, corresponding to birds, carnivores, herbivores and creeper. Using the proposed new method, the network structure starts from three nodes connected in a triangular way. Then the nodes are added to adapt the data distributions. The left of Fig.2 shows the final network structure used for the animal data set. The upper right triangle is the triangle the network starts with. The right of Fig.2 is the map generated by mapping the animals to the corresponding nodes on the map. The names of the animals are labelled manually.

The result shows meaningful clustering of the animals. Birds are mapped to the middle of the map; herbivores are mapped to the upper right of the map;

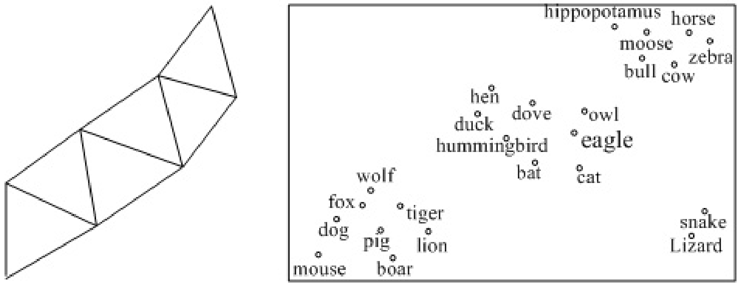


Fig. 2. The GSOM network structure and the Map of animals data set

creepers on the right of the map; the carnivores are gathered lower left corner. We can see clearly that within the cluster of *hummingbird*, *hawk*, *owl* is a subgroup, and *dove*, *duck* and *hen* is another subgroup. In the same cluster there are many similar phenomenon can be observed. *Lion* and *tiger* or *snake* and *lizard* are all have significant same features.

5 Conclusions

We present a growing structure according to an unknown probability distribution to cluster data which has the similar patterns. The clustering is done unsupervised only by sample vectors according to the distribution. The testing data set are 20-dimensional input vectors, from the result we can see it is very suitable to analyze high-dimensional data.

References

1. Kohonen, T.: Self-Organizing Maps. Springer-Verlag Berlin and Heidelberg, Germany (1995)
2. Kaski, S., Kangas, J., Kohonen, T.: Bibliography of Self-Organizing Map (SOM) Papers: 1981-1997. Neural Compute Surveys, **1** (1998) 1-176
3. Alahakoon, D., Halgarmuge, S.K., Srinivasan, B.: Dynamic Self-Organizing Maps with Controlled Growth for Knowledge Discovery. IEEE Transactions on Neural Networks, **11** (2000) 601-614
4. Fritzke, B.: Growing Grid A Self-Organizing Network with Constant Neighborhood Range and Adaption Strength. Neural Processing Letters, **2** (1995) 9-13
5. Davison, ML.: Mulidimensional Scaling. John Wiley and Sons, New York (1983)
6. Suganthan, P.N.: Hierarchical Overlapped SOM's for Pattern Classification. IEEE Transactions on Neural Networks, **10** (1999) 193-196
7. Rauber, A., Merkl, D., Dittenach, M.: The Growing Hierarchical Self-organizing Map: Exploratory Analysis of High-dimensional Data. IEEE Transactions on Neural Networks, **13** (2002) 1331-1341

A Novel Clustering Algorithm Based upon a SOFM Neural Network Family

Junhao Wen[1,2], Kaiwen Meng[2], Hongyan Wu[3], and Zhongfu Wu[3]

[1] College of Software Engineering, Chongqing University, Chongqing 400044, China
jhwen@cqu.edu.cn
[2] College of Science, Chongqing University, Chongqing 40044, China
[3] College of Computer Science, Chongqing University, Chongqing 400044, China

Abstract. A novel clustering algorithm based upon a SOFM neural network family is proposed in this paper. The algorithm takes full advantage of the characteristics of SOFM Neural Network family and defines a novel similarity measure, topological similarity, which help the clustering algorithm to handle the clusters with arbitrary shapes and avoid suffering from the limitations of the conventional clustering algorithms. The paper suggests another novel thought to tackle the clustering problem.

1 Introduction

Clustering may be defined as a process of partitioning a multidimensional input data space into groups of similar objects. The similarity of the objects is determined by a cardinal similarity measurement over the object attributes. The clustering arranges the objects so that objects within a cluster are more similar to each other than to objects in another cluster; the intracluster similarity is higher than the intercluster similarity [1]. Due to its unsupervised learning ability, clustering has been widely used in numerous applications, such as pattern recognition, image processing and so on. In spite of many various clustering algorithms, clustering is still a challenging field of research, where its potential applications pose their own special requirements [2].

A novel clustering algorithm based upon a family of self-organizing feature map, called SOFMF, is proposed in this paper. SOFMF takes advantage of a SOFM family, not a single SOFM, which really performs well.

2 Self-organizing Feature Map

Self-organizing feature maps (SOFM) learn to classify input vectors according to how they are grouped in the input space. As the neighboring neurons in the self-organizing output map learn to recognize neighboring sections of the input space, SOFM learn both the distribution and topology of the input vectors they are trained on, which is the main reason that our algorithm is based upon SOFM.

2.1 SOFM Algorithm

The self-organizing feature map learns the topological mapping $f : D \subset R^s \to G \subset R^m$ by means of self-organization driven by samples X in D,

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 69–74, 2005.

where G is an output map containing a set of neurons, each representing an element in the *m*-dimensional Euclidean space [3]. Denote $r_i \in G$ as the position of the i^{th} neuron in the output map. Let $X = [x_1, x_2, ..., x_s]^T \in D$ be the input vector. It is assumed that the input vector is connected in parallel to every neuron in the output map. The weight vector of the neuron i is denoted by $W_i = [w_{i1}, w_{i2}, ..., w_{is}]^T \in R^s$. The learning rule is given by [4]. $W_i(t+1) = W_i(t) + \alpha(t)\lambda(i, i^*)[X(t) - W_i(t)]$, where $t = 1, 2, 3, ...$ is the discrete time coordinate, $\alpha(t)$ is the learning rate factor, and $\lambda(i, i^*)$ is the neighborhood function. The winner neuron i^* is defined to be the neuron whose weight vector has the smallest Euclidean distance from the input $X(t)$:

$$\|W_{i^*}(t) - X(t)\| \le \|W_i(t) - X(t)\| \qquad \forall r_i \in G$$

The following is a typical neighborhood function proposed by Kohonen [5]: $\lambda(i, i^*) = \begin{cases} 1 & \text{for } \|r_i - r_{i^*}\| \le N_{i^*}(t) \\ 0 & \text{otherwise} \end{cases}$, where $N_{i^*}(t)$ is some decreasing function of time.

2.2 Definition of the SOFM Family

Now that the proposed algorithm is based upon a family of SOFM, we should firstly give out here the definition of the SOFM family. A SOFM family can be defined as follows:

Definition 1: A SOFM family is a collection of SOFM neural networks with all the network parameters the same except for the sizes of the output maps.

Let the input data set *P* be the set of n samples $X(t) \in D$, t=1,2,...,n . Giving each member of the SOFM family a member index, e.g. k , we can then denote the corresponding member as $SOFM_k$, and denote G_k as the output map of $SOFM_k$. If we train $SOFM_k$ several epochs regardless of the convergence using SOFM algorithm in section 2.1, then by simulating $SOFM_k$ with P , we obtain a partitioning of the input data set, as follows:

$$\begin{cases} P = \bigcup\limits_{\forall i \in G_k} N_k(i) \\ N_k(i) \cap N_k(j) = \varnothing \quad , \forall i, j \in G_k, i \ne j \end{cases} \tag{1}$$

In formula (1), $N_k(i)$denotes the winning region of the i^{th} neuron in G_k. If the i^{th} neuron wins none, $N_k(i)$is equal to $\varnothing$. Let C_k denotes the number of neurons whose winning regions are non-empty. The number C_k can be called the partition number in a sense of the hard C_k -partition of the input data set *P*. As SOFM keeps the distribution and topology of the input vectors, unlike conventional hard partition methods, the C_k -partition of P here keeps the distribution and topology of *P*. It is apparently that C_k can range from 1 to n provided that G_k changes accordingly and $SOFM_k$ is trained sufficiently. We conclude that larger G_k, hence lager C_k represents relatively local topology of P whereas smaller G_k, hence smaller C_k represents relatively global topology of P. The ratio of partition number C_k to the size of G_k gives us the impression of the usage factor of $SOFM_k$ in the way that the small ratio means too many neurons in G_k win none, then we can conclude that the output map G_k is too large and the usage factor is small. The definition of usage factor of $SOFM_k$ is as follows:

Definition 2: The usage factor $UsageFactor_k$ of $SOFM_k$ is the ratio of partition number c_k to the size of the output map G_k or the total number of the neurons in G_k, denoted as$\| G_k \|$, i.e. $UsageFactor_k = \frac{c_k}{\|G_k\|}$.

3 The Proposed Algorithm

In general, the idea of the proposed algorithm FSOFM comes from the idea of teamwork. We know that each member in a team has his own ability of doing work. They do different part of work and have different contributions. And the effectiveness and efficiency of teamwork is always high because all members work together, help each other and learn from each other. FSOFM takes advantage of a family of SOFM neural network. We expect that members of the family of SOFM neural network correct each other and work together to produce a satisfactory clustering result.

3.1 Construction of A SOFM Family

We follow the definition 1 to construct a SOFM family. A SOFM family with two-dimensional output maps is shown in table 1. Note that the neurons in the output map of a SOFM do not have to be arranged in a two-dimensional manner. You can use a one-dimensional arrangement, or even three or more dimensions.

Table 1. A SOFM family with two-dimensional output maps

Member index	1	2	3	4	5	6	7	8	9	…	21
Dimension 1	2	2	3	3	4	4	5	5	6	…	12
Dimension 2	2	3	3	4	4	5	5	6	6	…	12

3.2 The Topological Similarity Matrix

Assume that we have $SOFM_k$ trained several epochs with the input data set P, and then we get a c_k -partition of P by simulating the trained $SOFM_k$ with P, we may find that objects in the same winning region have more similarity in sense of Euclidean distance than those in different winning regions. As the c_k -partition of P is a topological-keeping partition, we use the term 'topological similarity' to define a novel similarity measure for all pairs of n objects, as follows:

Definition 3: Let $X(p), X(q) \in P$, then the topological similarity between $X(p)$ and $X(q)$ with respect to $SOFM_k$ is

$$TS_k(X(p),X(q))=\begin{cases} 1 & , X(p)\in N_k(i) \text{ and } X(q)\in N_k(i) \text{ for some } i\in G_k \\ 0 & , X(p)\in N_k(i) \text{ and } X(q)\in N_k(j) \text{ , } i\neq j, i,j\in G_k \end{cases} \tag{2}$$

From the definition above, the topological similarity between object and itself is always 1. If we store topological similarities of all pairs of n objects in a lower triangular matrix, then we have the **topological similarity** matrix, denoted as TSM_k with respect to $SOFM_k$, as follows:

$$TSM_k = \begin{bmatrix} 1 & & & & \\ TS_k(X(2),X(1)) & 1 & & & \\ TS_k(X(3),X(1)) & TS_k(X(3),X(2)) & 1 & & \\ & & & & \\ TS_k(X(n),X(1)) & TS_k(X(n),X(2)) & \ldots & \ldots & 1 \end{bmatrix} \tag{3}$$

When $c_k = 1$, TSM_k is the n-by-n unit upper triangular matrix, and when $c_k = n$, TSM_k is n-by-n unit matrix. They are two proper cases of the topological similarity matrix. The concept of topological similarity matrix introduced here is very important for our proposed algorithm as we will see in next section.

3.3 The Proposed Algorithm

As many clustering algorithms operate on dissimilarity or similarity matrix, our proposed algorithm also operates a topological similarity matrix defined in section 3.2. Figs.1-3 shows the sketched procedure of the proposed algorithm and the sketch of the details procedure.

All the topological similarity matrixes obtained from members of the SOFM family will work together and learn from other's strong points to counteract one's weak points, and thus correct each other. We define a topological similarity matrix *TSM* for the SOFM family, and then update it with TSM_k by matrix addition. The updating procedure stops when $UsageFactor_k$ is smaller than an acceptable usage factor, *AcceptUsageFactor*. Otherwise, all the members of the SOFM family will be involved in the updating procedure. See Fig. 2 for details about the updating procedure. In this paper, we use $AcceptUsageFactor$=0.2.

After the updating procedure is terminated, we can use *TSM* to build a hierarchical clustering. It is unsurprised that *TSM* contains the most interesting and valuable information underlying the input data set since *TSM* is the corporate result of the SFOM family. One can easily imagine that larger elements in *TSM* may reflect more topological similarity between pairs of objects they represents whereas smaller elements reflect less topological similarity. Denote *Minlevel* and *Maxlevel* as the minimal and maximal topological similarity respectively. If we construct an undirected complete graph using *TSM* as its adjacency matrix with weights, denoting it as *TSG*, and then cut off the edges whose weights are smaller than a number between *Minlevel* and *Maxlevel*, e.g. the current hierarchical level CHL, then by searing connected sub-graphs using *depth-first search* (*DFS*) technique, we can build the clusters in the current hierarchical level. Apparently, hierarchical level can range from *Minlevel* to *Maxlevel*. We describe in detail the procedure of building hierarchical clusters in Fig.3.

```
1: Load data set $P$
2: initialize $TSM$ with n-by-n unit lower matrix
3: initialize sizes of output maps for the SOFM family with table 1 .
4: update $TSM$ with $TSM_k$ obtained from $SOFM_k$
5: build hierarchical clusters
```

Fig. 1. The sketched procedure of the proposed algorithm (SOFMF)

```
1: Initialize current member index $k$ with 0
2: REPEAT
3:     $k=k+1$
4:     construct $SOFM_k$, with size of the output map equal to the $k^{th}$ member of the SOFM family, train it a fixed epochs, simulate $SOFM_k$ with input data set $P$ to obtain partition number $c_k$ and $TSM_k$ from (1) ,(2) and (3).
5:     $TSM = TSM + TSM_k$
6:     IF {$UsageFactor_k$ is smaller than $AcceptUsageFactor$ OR. $k$ reaches its maximum}
7:         BREAK;
8: END
```

Fig. 2. The sketch of the procedure of updating *TSM* with TSM_k

```
1: Calculate the maximal topological similarity $Maxlevel$ and the minimal topological similarity $Minlevel$ using $TSM_k$
2: initialize current hierarchical level $CHL$ with $Minlevel$
3: construct an undirected complete graph $TSG$ using $TSM$ as its adjacency matrix with weights.
4: Initialize current graph $CurTSG$ with $TSG$.
5: REPEAT
6:     cut off the edges of $CurTSG$ whose weights are smaller that $CHL$, producing a new graph $NewTSG$.
7:     search connected sub-graphs of $NewTSG$ using depth-first search (DFS) technique.
8:     build sub-clusters in the current hierarchical level with connected sub-graphs.
9:     Let the current graph $CurTSG$ be the new graph $NewTSG$
10:    $CHL= CHL+1$
11:    IF ( $CHL$ is larger than $Maxlevel$)
12:        BREAK
13: END
```

Fig. 3. The sketch of the procedure of building hierarchical clusters using *TSM*

4 Results

In this section, the performance of the proposed algorithm (SOFMF) is examined by data set **A**, which includes 1076 points with 2 outliers forming 14 clusters with arbitrary shapes, see Fig. 5.As the clustering results by SOFMF are organized in a hierarchical structure, and there are usually many hierarchical levels for a data set, we will not display clusters in all hierarchical levels. Each connected graph represents a cluster. The number of clusters is consistent with the number of the connected graphs algorithms, the algorithm is more efficient for its satisfactory clustering results and freedom of limitations of old algorithms.

5 Conclusions

Due to its great learning ability, Clustering algorithm has been an important research topic of diverse fields for many years. The paper introduces the concept, topological similarity, and takes advantage of the characteristics of SOFM Neural Network family to handle the clusters with arbitrary shapes. Compared with many conventional clustering although there maybe are some week points, it is still a novel algorithm worthy of researching further.

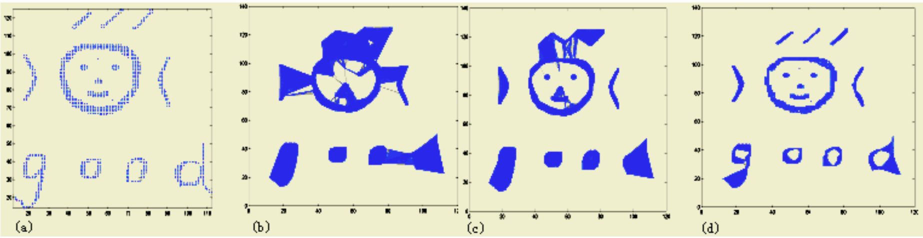


Fig. 4. (a) represents data set **A**; (b) The proposed algorithm finds out 4 clusters in level 6 .(c) 7 clusters are found in level 9; (d) All clusters and outliers are found in level 20.

Acknowledgements

This research was supported by Natural Science Foundation of Chongqing under agreement number:2004BB2182.

References

1. Hodge, V.J., Austin, J.: Hierarchical Growing Cell Structure: TreeGCS. IEEE transactions on knowledge and data engineering, **13** (2001)
2. Han, J., Kamber, M.: Data Mining Concepts and Techniques. Morgan Kaufmann Publisher, San Francisco (2001)
3. Lin, S., Si, J., Weight: Convergence and Weight Density of the Multi-dimensional SOFM Algorithm. Proceedings of the American Control Conference, Albuquerque, New Mexico (1997)
4. Kohonen, T.: Self-organizing Map. Heidelberg: Springer-Verlag (1995)
5. Kohonen, T.: The Self-organizing Map. Proceedings of the IEEE. **78** (1990) 1464-1480

Advanced Visualization Techniques for Self-organizing Maps with Graph-Based Methods

Georg Pölzlbauer[1], Andreas Rauber[1], and Michael Dittenbach[2]

[1] Department of Software Technology
Vienna University of Technology
Favoritenstr. 11-13, Vienna, Austria
{poelzlbauer,rauber}@ifs.tuwien.ac.at
[2] eCommerce Competence Center – ec3
Donau-City-Str. 1, Vienna, Austria
michael.dittenbach@ec3.at

Abstract. The Self-Organizing Map is a popular neural network model for data analysis, for which a wide variety of visualization techniques exists. We present a novel technique that takes the density of the data into account. Our method defines graphs resulting from nearest neighbor- and radius-based distance calculations in data space and shows projections of these graph structures on the map. It can then be observed how relations between the data are preserved by the projection, yielding interesting insights into the topology of the mapping, and helping to identify outliers as well as dense regions.

1 Introduction

The Self-Organizing Map [2] is a very popular artificial neural network algorithm based on unsupervised learning. It has been extended in several ways [1, 3]. It provides several beneficial properties, like vector quantization and topology preserving mapping from a high-dimensional input space to a usually two-dimensional map space. This projection can be visualized in numerous ways in order to reveal the characteristics of the input data or to analyze the quality of the obtained mapping. In this paper, we present a novel graph-based visualization technique, which provides an overview of the cluster structure and uncovers topology violations of the mapping. We propose two methods for defining the graph in input space. The first one computes a graph structure based on nearest neighbor calculations, and is especially useful for large SOMs, where map units outnumber data samples. The second method creates a graph structure based on pairwise distances between data points in input space, and its advantages are the easy identification of outliers and insight into the density of a region on the map. We provide experimental results to illustrate our methods on SOMs trained on the Ionosphere data set [7].

The remainder of this paper is organized as follows. In Section 2 a brief introduction to related visualization techniques is given. Section 3 details our

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 75–80, 2005.

proposed method, followed by detailed description of its properties and experimental results with this method provided in Section 4. Finally, some conclusions are drawn in Section 5.

2 Related Work

In this section, we briefly describe visualization concepts related to our method. The most common ones are component planes and the U-Matrix. For an in-depth discussion, see [9]. The emphasis of our paper lies on visualization techniques that take the distribution of the data set in input space and its density into account. Most commonly, this is visualized as hit histograms, which display the number of data points projected to each map node. A more advanced method is the P-Matrix [8] that visualizes how densely populated each unit is by counting the number of data points within the sphere of a certain radius around the model vector in question. Another recently proposed technique that aims at depicting both density and cluster structures is the Smoothed Data Histogram [4], which relies on a parameter that determines how blurred the visualization will be. There are also techniques that depict the contribution of the individual variable dimensions to the clustering structure, like LabelSOM [5]. Other techniques providing insight into the distribution of the data manifold are projection methods like PCA and Sammon's Mapping.

3 A Graph Projection Method

Our method investigates the proximity of the data vectors in input space and the preservation of pairwise distances after projection. First, we introduce a notation for both the SOM and the required concepts from graph theory. The input data set X contains N sample vectors x_i of dimension D^{input}. The SOM consists of M model vectors m_i of the same dimension as the input data, which are arranged on a two-dimensional map lattice, usually either in a rectangular or hexagonal fashion. Since the SOM is a vector projection technique, all data samples can be assigned a position on the map lattice. This is performed by finding the best-matching unit (BMU), formally

$$\phi(x_i) = \arg\min_j d(x_i, m_j) \tag{1}$$

where d is a suitable distance metric, like Euclidean Distance. The BMU is the prototype vector which is closest to the data sample x_i. The position of the model vector m_j on the map in the form of its two-dimensional position is also the projection of data vector x_i.

Next, we require some definitions from graph theory. A graph is a set of vertices and edges, formally $G = \{V, E\}$. The edges are usually represented by a square adjacency matrix (e_{ij}). In case the graph is undirected the adjacency matrix is symmetric. We require that there are no connections from vertices to themselves, so the diagonal elements are all zero.

Then, we compute a graph G^{input} that captures the characteristics of the data set. The vertices of this graph are the data vectors. Our goal is to obtain a set of edges that connect those data samples which satisfy a certain condition of proximity. We then aim at depicting the projection of G^{input} on the map lattice that visually link the corresponding map nodes with lines, indicating whether the original distances are preserved. In the following, we describe two methods of how to define the edges of G^{input}. The first one requires a parameter r and defines the data sample x_i to be adjacent to x_j if it lies within a sphere of radius r and center x_i. The entries of the $N \times N$ adjacency matrix are computed as

$$e_{ij}^{rad} = \begin{cases} 1 \text{ if } i \neq j \wedge d(x_i, x_j) \leq r \\ 0 \text{ otherwise} \end{cases} \tag{2}$$

The resulting graph is necessarily undirected due to the symmetry condition of distance metric d. The radius r serves as a threshold value. The number of edges increases monotonically with increasing r.

The second way to define the graph structure involves nearest neighbor calculations. It requires the integer parameter k which indicates how many neighbors to include. A sample x_j is connected to x_i if it is among its set of k nearest neighbors $N_k(x_i)$, formally

$$x_j \in N_k(x_i) \Longleftrightarrow \text{Card}\{x_l \in X : l \neq i, j \wedge d(x_l, x_i) < d(x_j, x_i)\} < k \tag{3}$$

where Card denotes the number of elements of a set. In case of a tie in the ranking of the distances, this formula does not lead to a set of exactly k members, a policy to handle this exception has to be applied. Other than the radius method, the nearest neighbor relationship is not necessarily symmetric, so the definition of the elements of the adjacency matrix are defined as

$$e_{ij}^{kNN} = \begin{cases} 1 \text{ if } i \neq j \wedge \big(x_i \in N_k(x_j) \vee x_j \in N_k(x_i)\big) \\ 0 \text{ otherwise} \end{cases} \tag{4}$$

Here, edges are defined if x_i is k nearest neighbor to x_j or vice versa. As with the radius method, increasing the value of k leads to more edges in the graph.

Once the graph is computed, it is ready for projection. A second graph G^{SOM} is defined for the output space. The vertices v_i^{SOM} are the prototype vectors m_i. The edges are preserved from the original structure such that a pair of model vectors are connected if two connected data samples are mapped onto them. The elements of the $M \times M$ adjacency matrix (e_{ij}^{SOM}) are defined as

$$e_{ij}^{SOM} = \begin{cases} 1 \text{ if } i \neq j \wedge \exists x_k, x_l : e_{kl}^{input} = 1 \wedge m_i = \phi(x_k) \wedge m_j = \phi(x_l) \\ 0 \text{ otherwise} \end{cases} \tag{5}$$

where e^{input} is the graph for the connectivity of the input space, either e^{rad} or e^{kNN}. While the number of vertices of the projection can be either greater or lesser than the number of vertices in input space, depending on whether the prototype vectors outnumber the sample vectors or vice versa, the number of

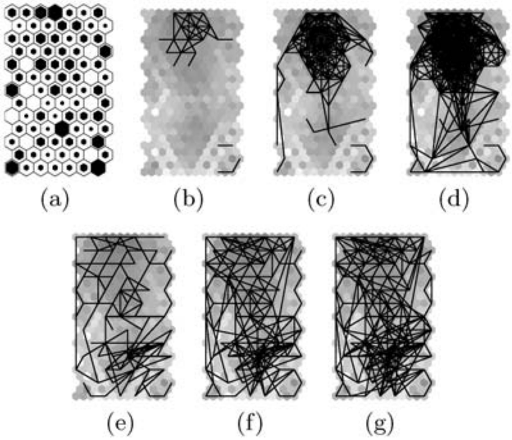


Fig. 1. Ionosphere 7 × 13 SOM: (a) hit histogram, Radius method: (b) $r = 1.0$, (c) $r = 2.0$, (d) $r = 3.0$, Nearest neighbors method: (e) $k = 1$, (f) $k = 2$, (g) $k = 3$

projected edges are at most equal to the number of edges before the mapping is applied. Connected data points mapped to the same map unit are not counted as edges, and this is generally an indication of good projection quality, since samples close in input space are close on the SOM as well in this case.

Finally, we can visualize the results on the two-dimensional map. This is performed by drawing lines connecting those map units for which edges in graph G^{SOM} exist. The resulting image reveals which areas of the SOM are densely populated, where interpolating units and regions lie, and where outliers are located. The interpretation of this visualization depends on the size of the map and the choice of parameter r or k, respectively.

4 Experimental Results and Properties of Our Method

In this section, we will see some experimental results with maps trained on the Ionosphere data set, which consists of 351 sample vectors of 34 nominal and numerical dimensions. There is a 35^{th} variable that serves as a class label and is omitted in SOM training, because of the SOM's unsupervised nature. The input space is densely populated in one region, while very sparsely in others. We will see that this property can be illustrated by our technique. Before training, the data is normalized to unit variance. The SOMs we investigate are trained with a two-dimensional lattice with hexagonal map units, one with a grid consisting of 7 × 13 nodes, and a larger one with 40 × 60 nodes, where data vectors are outnumbered by map units.

In Figure 1, the smaller version of the SOMs is visualized. Figure 1(a) shows the hit histogram that depicts the data vectors projected onto the map lattice. The U-Matrix is shown in the background of the other plots, indicating cluster borders with bright colors. Figures 1(b)–(d) depict the radius induced method at different levels of $r = 1, 2, 3$. It can be clearly seen that the density of the data

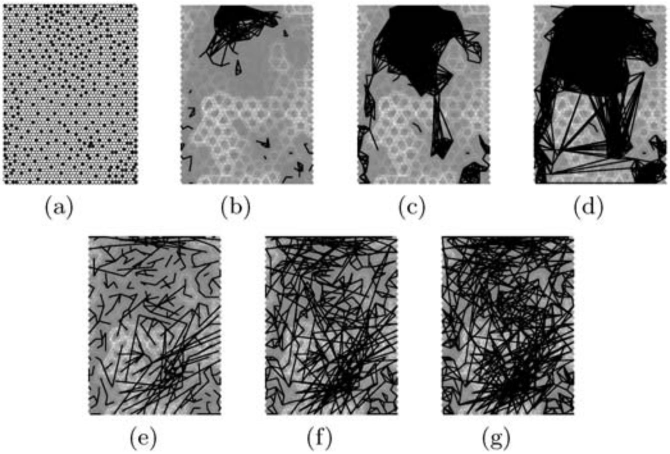


Fig. 2. Ionosphere 40 × 60 SOM: (a) hit histogram, Radius method: (b) $r = 1.0$, (c) $r = 2.0$, (d) $r = 3.0$, Nearest neighbors method: (e) $k = 1$, (f) $k = 2$, (g) $k = 3$

points is higher in the upper half of the map. This is not so obvious in either the U-Matrix and hit histogram visualizations. The nearest neighbors method is shown for $k = 1, 2, 3$ in Figures 1(e)–(g). Obviously, more lines are plotted. The emphasis here lies not on the identification of dense areas, but rather to single out regions where the mapping is distorted, as in the center of the bottom part of the map. Here, many lines point to distant areas of the map, which is an indication that the input space cannot be as easily clustered and projected as the model vectors in the upper half.

The larger version of the map is depicted in Figure 2 with the same parameter values as before. The graphs in input space is of course the same, only the mapping is different. It can be seen that the dense regions identified by the radius method is very similar to the smaller version. Outliers can be detected as those areas that do not show connections for high values of r, like the upper left corner. Due to the higher resolution, the lines can be distinguished more easily. The nearest neighbors method, depicted in Figures 2(e)–(g), again shows an evenly distributed picture of the connections. The region in the center of the bottom part seems distorted with lines running diagonally through it, although the radius method shows that it is not sparsely populated. Thus, topology violations due to the loss of dimensionality during the mapping are likely to have occurred.

Another interesting property is that the radius method tends to form more closed geometric figures like triangles, while these forms are star-shaped in the nearest neighbors method. This is due to the different relation, which is symmetric in the radius case, and while not transitive in a mathematical sense, tends to group points together. The radius method is related to single linkage clustering [6]. When single linkage is performed, nodes are joined within a certain distance. Our radius method works similarly, hence, the graph structure with radius r reflects the clustering at level r in single linkage.

5 Conclusion

In this paper, we have seen a novel method for visualization of Self-Organizing Maps that is based on the set of data samples. This technique can easily be implemented for 2-dimensional map lattices. Two different definitions of proximity have been introduced, one that defines connectivity as a nearest neighbor relationship, while the second employs a density-based approach. Our experiments have shown that they are best applied in combination with other SOM visualization methods, like U-Matrix and hit histograms. We have found the nearest neighbor approach to be especially useful for maps with a large number of units compared to the number of data points. The radius method is more reliable with respect to outliers.

Acknowledgements

Part of this work was supported by the European Union in the IST 6. Framework Program, MUSCLE NoE on Multimedia Understanding through Semantics, Computation and Learning, contract 507752.

References

1. Oja, E., Pakkanen, J., Iivarinen, J.: The Evolving Tree–a Novel Self-Organizing Network for Data Analysis. Neural Processing Letters, **20** (2004) 199–211
2. Kohonen, T.: Self-Organizing Maps, 3rd edition. Springer (2001)
3. Merkl, D., Dittenbach, M., Rauber, A.: Uncovering Hierarchical Structure in Data Using the Growing Hierarchical Self-Organizing Map. Neurocomputing, **48** (2002) 199–216
4. Pampalk, E., Rauber, A., Merkl, D.: Using Smoothed Data Histograms for Cluster Visualization in Self-Organizing Maps. In Proc. Intl. Conf. on Artifical Neural Networks (ICANN'02), Madrid, Spain, Springer (2002)
5. Rauber, A., Merkl, D.: Automatic Labeling of Self-Organizing Maps: Making a Treasure-map Reveal Its Secrets. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'99), Bejing, China, Springer (1999)
6. Rauber, A., Pampalk, E., Paralic, J.: Empirical Evaluation of Clustering Algorithms. Journal of Information and Organizational Sciences (JIOS), **24** (2000) 195–209
7. Sigillito, V., Wing, S., Hutton, L., Baker, K.: Classification of Radar Returns from the Ionosphere Using Neural Networks. Johns Hopkins APL Technical Digest, **10** (1989) 262–266
8. Ultsch, A.: Maps for the Visualization of High-dimensional Data Spaces. In Proc. Workshop on Self organizing Maps, Kyushu, Japan (2003)
9. Vesanto, J.: Data Exploration Process Based on the Self-Organizing Map. Ph.D. thesis, Helsinki University of Technology (2002)

Selection of Optimal Features for Iris Recognition

Hongying Gu, Zhiwen Gao, and Fei Wu

Institute of Artificial Intelligence, Zhejiang University 310027
Hangzhou, China
guhy@cs.zju.edu.cn

Abstract. Iris recognition is a prospering biometric method, but some technical difficulties still exist. This paper proposes an iris recognition method based on selected optimal features and statistical learning. To better represent the variation details in irises, we extract features from both spatial and frequency domain. Multi-objective genetic algorithm is then employed to optimize the features. Next step is doing classification of the optimal feature sequence. SVM has recently generated a great interest in the community of machine learning due to its excellent generalization performance in a wide variety of learning problems. We modified traditional SVM as non-symmetrical support vector machine to satisfy the different security requirements in iris recognition applications. Experimental data shows that the selected feature sequence represents the variation details of the iris patterns properly.

1 Introduction

As a kind of pattern recognition, iris recognition is to classify the iris correctly by comparing the similarity between irises. It includes two major works: Feature extraction and classifier design. Usually there is only one kind of features considered in former iris recognition methods. Wavelet transform and Gabor transform are most often chosen to get the features in the former iris recognition algorithms [1], [2]. While iris textures are so complex that it is inadequate to represent iris by only one transform in some cases. Steerable pyramid is designed to enhance wavelet transform by translation, dilation, and rotation invariance [3]. We extract the features based on steerable pyramid and fractal geometry to represent the oriented self-similar textures.

With the variation fractal dimension and the steerable pyramid features working together, the features can represent irises more completely. However, as a feature sequence, it is not true that the more is the better. There might be some redundancy existing among the features which increases the algorithm cost. Selecting the right features is an important issue for accurate classification of irises. To get more precise fusion features, we do the feature selection by multi-object evolution algorithm.

Finally, we adopt non-symmetrical support vector machine (NSVM) as the classifier which satisfies variant security requirements in real world's biometrics applications.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 81–86, 2005.

2 Features Extraction and Selection

The variation details of iris texture are studied in iris patterns and the self-similarity is found in variation details. Based on this, the variation fractal dimension [4] is chosen to represent irises. It is a new fractal dimension derived from traditional fractal geometry. It works well even if some regions of iris are obscured or occluded by eyelashes and eyelids.

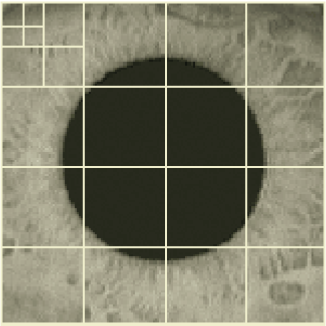


Fig. 1. An iris pattern from CASIA database

Fig. 1. shows a common iris image from CASIA iris database. Besides the distinct aspect of the iris which comes from randomly distributed features, we can see that the pupil has a big influence on features and thus the classification result. It will be processed by multi-objectives genetic algorithm after feature extraction unit.

2.1 Variant Fractal Dimension

Based on the self-similarity we get from the study on the texture in iris patterns, we take it as a kind of typical fractal phenomenon. The mathematical way to measure fractals is by fractal dimension.

[4] proposed a special fractal dimension for texture images like irises:

$$Dim_B F = \lim_{\delta \to 0} \frac{\log N_\delta(IsChange(F_\delta))}{-\log \delta} \tag{1}$$

Where F_δ is the image covered by square boxes of side δ , $IsChange(F_\delta)$ will be evaluated with two thresholds. Basically it is 1 if the gray scale in F_δ changes and 0 otherwise. $N_\delta\left(IsChange(F_\delta)\right)$ is the smallest number if square (or cubic) boxes of side δ necessary to cover F_δ.

How many variation fractal dimension features we should extract depends on the different application conditions. Here 16 features are extracted in each iris. These features are input to multi-objectives genetic algorithm for feature selection.

2.2 Steerable Pyramid

We can see this self-similarity is multi-oriented around the pupil in Fig. 1. The variation fractal dimension faces a problem that it doesn't take the orientation of iris textures into consideration. However, for recognition applications, the orientation information is very important. In order to better represent iris patterns, we extract features based on both spatial and frequency domain information. Variation fractal dimension is used to get self-similarity feature in spatial domain, and steerable pyramid transform is adopted to process frequency information of iris image.

To solve this problem, we introduce steerable pyramid into iris feature extraction. Steerable pyramid transform can analyze the anisotropic textures of irises. This transform decomposes the image into several spatial frequency bands. In addition, it further divides each frequency band into a set of orientation bands. By studying each orientation bands, we can get orientation information of iris textures.

More importantly, the representation is translation invariant and rotation invariant [5], which perfectly suits iris recognition.

A linear image transform represents an image as a weighted sum of basis functions. That is, the image, $I(x, y)$, is represented as a sum over an indexed collection of functions, $g_i(x, y)$

$$I(x, y)=\sum_i y_i g_i(x, y) . \tag{2}$$

Where y_i are the transform coefficients. In iris recognition applications, an iris image is decomposed into a set of subbands (see fig. 2.), and the information within each subband is processed more or less independently of that in the other subbands.

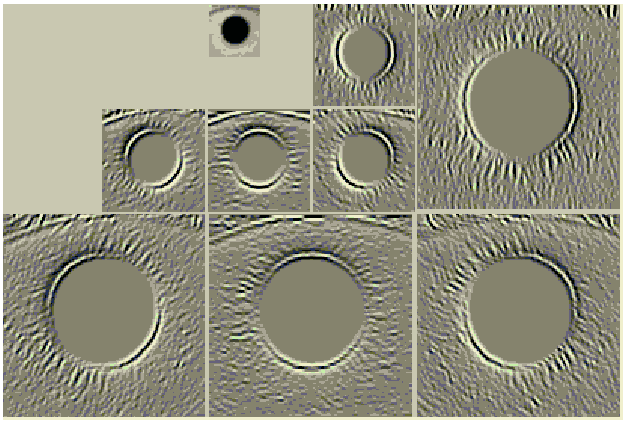


Fig. 2. A set of orientation bands of an iris image

Firstly, we get a set of orientation bands of an Iris image. Then, we calculate the variation fractal dimension upon each orientation bands to present the self-similarity of the different orientation bands.

2.3 Feature Selection by Multi-objectives Genetic Algorithm

Now we get 24 features in total as the feature sequence with a relative high dimension. It is necessary to select the most representative feature sequence. We propose a genetic process to select an appropriate set of features. It has been demonstrated that feature selection through multi-objective genetic algorithm (MOGA) is a very powerful tool to find a set (Pareto-optimal) of good classifiers [6]. Besides, it can overcome problems such as scaling and sensitivity towards the weights. In this light, we propose a new methodology for selecting features based on MOGA. For more information about genetic algorithm, please refer to [7].

In our feature selection process, iris feature sequences are represented in binary stings. In the binary string, a bit is "1" represents the corresponding feature is selected and "0" represents the corresponding feature is abandoned. For example, 111110011001111 represents a binary string of the selected results where 6th, 7th, 10th and 11th features are abandoned.

The objectives of genetic algorithm are defined as:

Cohesion: the distance between iris samples in the same class.

$$J_l = \frac{1}{n_l}\sum_{k=1}^{n_l}(x_k{}^l - m_l)^T(x_k{}^l - m_l)\,. \tag{3}$$

Where $l = 1,2,\cdots,c$; m_l is the mean vector of class l .

Coupling: the distance between iris samples of different classes.

$$J_b = \sum_{l=1}^{c} p_l \times (m_l - m)^T(m_l - m)\,. \tag{4}$$

Where, m is the mean vector of all iris samples.

Usually we want loose coupling between iris classes and high cohesion of classes. So we process genetic algorithm with these two objectives. And the fitness is defined as:

$$\min(Cost(S)) = \min(J_l(S)) + \min(\frac{1}{J_b(S)+1}) \tag{5}$$

.

The arguments of the algorithm are set as following: Population (the scale of iris samples) is 108; Length of chromosome code (number of features) is 24; crossing probability is 0.7; crossing points number is 2; aberrance probability is 0.01; generation is 50.

The experimental result (table 1) shows that the performance of the system has an apparent improvement. It is because redundancy does exist among the 24 extracted features and MOGA works well when performing feature selection.

3 Features Classification

We use support vector machine (SVM) as our features classifier. SVM attracts great attention recently both in theory and in application [8]. For more details about SVM theory, see [9]. The traditional SVM has a problem that it process False Accept and False Reject without any difference, which limits the feasibility. So we modify traditional SVM as non-symmetrical SVM to satisfy the different security requirements by a constant m .

We call m non-symmetrical parameter. It is used to adjust the position of the classification hyperplane H :

$$w \cdot x + b + m = 0 . \tag{6}$$

In (6), if $m > 0$, it means the classification hyperplane H is closer to the center of positive samples. In this way, we can adjust the position of H according to different security requirements in different applications, and H can be different distance away from support vectors. By changing the value of non-symmetrical parameter m , the False Accept Ratio (FAR) can be reduced. Consequently the two kinds of mistakes are punished differently. The decision-making function is:

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{k} \alpha_i y_i K(x, x_i) + b + m \right). \tag{7}$$

4 Experiments and Results

To evaluate the performance of the proposed method, we provide evidence of our analysis using CASIA Iris Image Database [10] from National Laboratory of Pattern Recognition (NLPR), Institute of Automation (IA), Chinese Academy of Sciences (CAS). The database includes 756 iris images from 108 different eyes of 80 subjects. The images are acquired during different sessions and the time interval between two collections is one month, which is a real-world application case simulation.

According to the algorithms discussed above, we implement the iris recognition system and do 5250 times testing. Now we list the experimental result as the widely-used form:

Table 1. Comparison of Accuracy, FAR and FRR

Features used	Accuracy (%)	FAR (%)	FRR (%)
Traditional fractal	87.89	11.75	0.36
Variation fractal	98.21	1.45	0.34
Variation fractal + orientation features	98.5	1.26	0.23
Selected variation fractal + orientation features	99.14	0.63	0.23

We can see the variation fractal gets much higher accuracy than the traditional fractal when used alone. The experiments on four different feature sets have demonstrated the validity and efficiency of the proposed strategy by finding small ensembles, which succeed in improving the recognition rates.

5 Conclusion

In this paper, we have presented a method which regards the texture of the iris as a kind of fractal and uses the steerable pyramid to make the features invariant to translation, scale and rotation. We propose iris feature fusion mechanism. SVM is employed as the classifier. To make the SVM more applicable in biometric systems, we make it non-symmetrical. The experiment results show that the method proposed in this paper is promising. The ideas of non-symmetrical SVM and multi-orientation features can be applied in a wide variety of application fields.

References

1. Daugman, J.G.: High Confidence Visual Recognition of Persons by a Test of Statistical Independence. IEEE Transactions on Pattern Analysis and Machine Intelligence, **15** (1993) 1148-1161
2. Ma, L., Wang, Y., Tan, T.: Iris Recognition Based on Multichannel Gabor filtering. Proc.5th Asian Conf. Computer Vision, **1** (2002) 279-283
3. Simoncelli, E. P.: A Rotation-Invariant Pattern Signature. IEEE International Conference on Image Processing, **3** (1996) 185-188
4. Gu, H., Pan, H., Wu, F., Zhuang, Y., Pan, Y.: The Research of Iris Recognition Based on Self-similarity. Journal of Computer-Aided Design & Computer Graphics, **16** (2004) 973-977
5. Simoncelli, E.P., Freeman, W.T.: The Steerable Pyramid: A Flexible Architecture for Multi-Scale Derivative Computation. 2nd IEEE International Conference on Image Processing, Washington, DC. **3** (1995) 444-447
6. Oliveira, L.S., Sabourin, R.F., Bortolozzi, Suen, C.Y.: Feature Selection Using Multiobjective Genetic Algorithms for Handwritten Digit Recognition. In 16th ICPR (2002) 568–571
7. Tan, K.C., Khor, E.F., Lee, T.H.: Multiobjective Evolutionary Algorithms and Applications. Springer-Verlag, United Kingdom (2004)
8. DeCoste, D., Scholkopf, B.: Training Invariant Support Vector Machines. Machine Learning, **46** (2002) 161-190
9. Vapnik, V.N.: Statistical Learning Theory. Wiley, J. New York (1998)
10. National Laboratory of Pattern Recognition (NLPR), Institute of Automation (IA), Chinese Academy of Sciences (CAS)□CASIA Iris Image Database, http://www.sinobiometrics.com (2003)

Application of Multi-weighted Neuron for Iris Recognition

Wenming Cao[1,2], Jianhui Hu[1], Gang Xiao[1], and Shoujue Wang[2]

[1] The College of Information Engineering, Zhejiang University of Technology
Hangzhou, Zhejiang 310014, China
[2] Lab. of Artificial Neural Networks, Institute of Semiconductors, CAS
Beijing 100083, China
csann@zjut.edu.cn

Abstract. In this paper, from the cognition science point of view, we constructed a neuron of multi-weighted neural network, and proposed a new method for iris recognition based on multi-weighted neuron. In this method, irises are trained as "cognition" one class by one class, and it doesn't influence the original recognition knowledge for samples of the new added class. The results of experiments show the correct rejection rate is 98.9%, the correct cognition rate and the error recognition rate are 95.71% and 3.5% respectively. The experimental results demonstrate that the correct rejection rate of the test samples excluded in the classes of training samples is very high. It proves the proposed method for iris recognition is effective.

1 Introduction

In recent years, with the development of information technology and the increasing need for security, intelligent personal identification has become a very important and urgent problem. The emerging biometric technology can solve the problem, which takes the unique, reliable and stable biometric features (such as fingerprints, iris, face, palm-prints, gait etc.) as identification body. This technology has very high security, reliability and effectiveness. As one of the biometric technology, iris recognition has very high reliability. Comparing with other biometric identification technology, the fault acceptance rate and the fault rejection rate of iris recognition are very low. The technology of iris recognition has many advantages, i.e., stability, non-invasiveness, uniqueness. All there desirable properties make the technology of iris recognition has very high commercial value. Based on the above reasons, many researchers have applied themselves to this field. Daugman used multi-scale quadrature wavelets to extract texture-phase structure information of iris to generate a 2048-bit iriscode and compared the difference between a pair of iris representations by computing their Hamming distance via the XOR operator, Wildes et al. represented the iris texture with a Laplacian pyramid constructed with four different resolution levels and used the normalized correlation to determine whether the input image and the model image are from the same class [1]. Boles et al. calculated zero-crossing representation of 1D wavelet transform at various resolution levels of a virtual circle on an iris image to characterize the texture of the iris. Iris mating was based on two dissimilarity func-

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 87–92, 2005.

tions [6]. In this paper, from the cognition science point of view, we constructed a neuron of multi-weighted neural network[2],[3],[5],[8], and proposed a new method for iris recognition based on multi-weighted neuron. In this method, irises are trained as "cognition" one class by one class, and it doesn't influence the original recognition knowledge for samples of the new added class. The results of experiments show the correct rejection rate is 98.9%, the correct cognition rate and the error recognition rate are 95.71% and 3.5% respectively. The experimental results demonstrate that the correct rejection rate of the test samples excluded in the classes of training samples is very high. It proves the proposed method for iris recognition is effective.

The remainder of this paper is organized as follows. Section 2 describes image preprocessing. Section 3 introduces iris recognition algorithm based on multi-weighted neuron. Experiments results and experimental analysis are given in Section 4 and Section 5 respectively.

2 Image Preprocessing

Iris image preprocessing is mainly composed of iris localization, iris normalization and enhancement.

The original iris image (see Fig.1(a)) has some character of the gray-scale distribution. The iris is darker than the sclera, and the pupil is greatly darker than the iris, as shown in Fig.1. From the histogram (see Fig.1(b)), we can clearly see that the low gray-scale mainly converges at the first peak value. Therefore, we adopt the binary transform to localize the inner boundary. From the image after binary transform (see Fig.1(c)), we find that the areas of zero gray-scale are almost the areas of the pupil and eyelash. Therefore, we reduce the influence of the eyelash by erode and dilation (see Fig.1(d)).

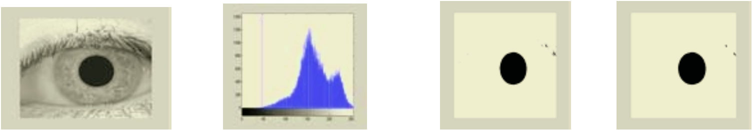


Fig. 1. (a) original image (b) histogram of the iris(c) binary image (d) binary image after erode

From the Fig.1(d), we can find that the length and the midpoint of the longest chord can be taken as the approximate diameter and center of the pupil respectively. Namely, according the geometry knowledge, let the length of the longest chord is $dia\max$, and the coordinates of the first point of the chord are $xbegin$ and $ybegin$, then

$$xpupil = xbegin + \frac{dia\max}{2}, ypupil = ybegin, rpupil = \frac{dia\max}{2} 。$$

Where $xpupil$ and $ypupil$ denote the center coordinates of the pupil, and $rpupil$ denotes the radius of the pupil.

When the quality of the image is reliable, this algorithm can localize the pupil quickly and exactly. Otherwise, we can correct the method as follow:

1. We can reduce the searching area by subtracting the pixels on the edge of the image.

2. We can get k chords, which are less than a certain threshold near the longest chord, and take the average value of center coordinates of k chord as the center of the pupil.

The exact parameters of the outer boundary are obtained by using edge detection (Canny operator in our experiments) and Hough transform. The image after Edge detection includes some useless points. For eliminating the influence, we remove the useless points between the areas of $\left[30^\circ, 150^\circ\right]$ and $\left[225^\circ, 315^\circ\right]$ according to the center of the pupil. Then, Hough transform is adopted to localize the outer boundary.

By the above method, we can localize the inner boundary and the outer boundary of the iris exactly. The localizations results of the iris are showed in Fig.2.

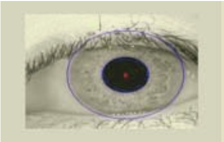


Fig. 2. Localized image

3 Iris Recognition Algorithm Based on Multi-weighted Neuron

Now, the task of geometrical learning is to cover a given sample set by a chain of hyper sausage units with a minimum sum of volumes via determining the end points of each line segment and the radius of each hyper-sphere. As introduced in section 2.2 the main idea is similar to finding the center and radius of the outer hyper-sphere via successive projection from a higher dimensional space to a lower dimensional space in help of descriptive high dimensional geometry.

To simplify implementation, the multi-weighted neuron is approximated by the shape (in solid line) that can be computed be the following characteristic function:

$$f_{MWN}(X) = \operatorname{sgn}\left[2^{\frac{d^2(X.\overline{X_1X_2})}{r^2}} - 0.5\right] \tag{1}$$

which contains a radius parameter r and the distance between X and the line segment $\overline{X_1X_2}$ as follows:

$$d^2(X,\overline{X_1X_2}) = \begin{cases} \|X - X_1\|^2, q(X, X_1, X_2) < 0 \\ \|X - X_1\|^2, q(X, X_1, X_2) > \|X_1 - X_2\| \\ \|X - X_1\|^2 - q^2(X, X_1, X_2), otherwise \end{cases} \tag{2}$$

Where $q(X,X_1,X_2)=(X-X_1)\cdot\frac{(X_1-X_2)}{\|X_1-X_2\|}$,Given a ordered set of samples $P=\{x_i\}_{i=j}^{n}$. The set is sampled in a certain order, which obeys the rule that the mid sample is more like the anterior sample than the latter one. This assures that the set of the samples is a continuously mutative chain. We select a parameter D ,the distance between the two contiguous selected samples in S . This parameter determines the total of the MWN neurons. From P we choose a set $S\{s_i\,|\,d(s_{i+1},s_i)\approx D,1\leq i<m\}$of n_j sample support points as the sausage parameters $\{X_{j1},X_{j2}\}_{j=i}^{n}$defined by (1) such that all the MWN units become overlapped, in help of the following algorithm:

Let S denote the filtered set that contains the samples which determine the network and X denote the original set that contains all the samples sampled in the order.

Begin

1.Put the first sample into the result set S and let it be the fiducial sample s_b , and the distance between the others and it will be compared. Set S ={ s_b }. $s_{\max}=s_b$ and $d_{\max}=0$

2. If no sample in the original set X ,stop filtering. Otherwise , check the next sample in X , then compute its distance to s_b ,i.e., $d=\|s-s_b\|$.

3. If $d>d_{\max}$, goto step 6. Otherwise continue to step 4.

4.If $d<\varepsilon$,set $s_{\max}=s$, $d_{\max}=d$, goto step 2. Otherwise continue to step 5.

5.Put s into the result set: $S=S\cup\{s\}$,and let $s_b=s$, $s_{\max}=s$, and $d_{\max}=d$. Then go to step 2.

6.If $d_{\max}-d>\varepsilon_2$, go to step 2. Otherwise put $s_{\max}$ into the result set: $S=S\cup\{s_{\max}\}$,and let $s_b=s_{\max}$, $d_{\max}=\|s-s_{\max}\|$ go to step2.

4 Experimental Results

Images of CASIA (Institute of Automation, Chinese Academy of Sciences) iris image database are used in this paper. The database includes 742 iris images from 106 different eyes (hence 106 different classes) of 80 subjects. For each iris class, images are captured in two different sessions and the interval between two sessions is one month. The experiment processes and experiment results are presented as follow:

(1) In our experiment, 3 random samples from each class in the frontal 80 classes (hence, 240 samples) are chosen for training, and a MWN neuron of multi-weighted

neural network is constructed for the 3 samples. Five samples from the training set are shown in Fig.3. Then, the entire iris database is taken as test sample set. In which, 182 (26×7) samples, which don't belong to the classes of training samples, are referred to the first sample set. The remainder of total 560 (80×7) samples is referred to the second sample set. Fig.4 shows five samples from the second test set and Fig.5 shows five samples from the first test set.

(2) The correct rejection rate=the number of samples which are rejected correctly in the first sample set/the total number of the first sample set. The correct cognition rate=the number of samples which are recognized corrected in the second sample set / the total number of the second sample set. The error recognition rate= (the number of samples which are recognized mistakenly in the first sample set +the number of samples which are recognized mistakenly in the second sample set) / the total number of the second sample set.

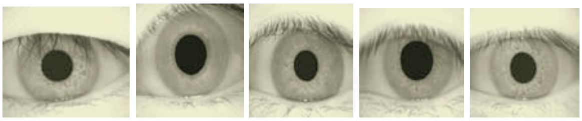


Fig. 4. Iris samples from the training set

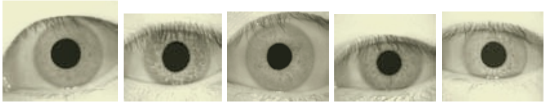


Fig. 5. Iris samples from the second test set

Fig. 6. Iris samples from the first test set

(3) For total 742 test samples, 180 samples are rejected correctly and the other 2 samples are recognized mistakenly in the first test sample; and 536 samples are recognized correctly and the rest 24 samples are recognized mistakenly in the second test sample. Therefore, the correct rejection rate is 98.9%(180/182), the correct cognition rate and the error recognition rate are 95.71%(536/560) and 3.5%((2+24)/742) respectively.

5 Experimental Analysis

We can conclude from the above experimental results that:

(1) Irises are trained as "cognition" one class by one class in our method, and it doesn't influence the original recognition knowledge for samples of the new added class.

(2) Although the correct cognition rate is not very well, the result of the correct rejection rate is wonderful. In our experiment, the correct rejection rate is 98.9%, namely, the iris classes that don't belong to the training test can be rejected successfully.

(3) The iris recognition algorithm based on neuron of multi-weighted neural network is applied in the experiment and the total samples of every class construct the shape of 1D distribution. Namely, it is the network connection of different neuron.

(4) In above experiment, if the image preprocessing is more perfectly, the experimental results maybe better.

To sum up, it proves the proposed iris recognition algorithm based on multi-weighted neuron is effective.

References

1. Boles, W., Boashash, B.: A Human Identification Technique Using Image of the Iris and Wavelet Transform, IEEE Trans. Signal Process, **46** (1998) 1185-1188
2. Wang, S., Li, Z., Chen, X., Wang, B.: Discussion on the Basic Mata-hematical Models of Neurons in General Purpose Neurocomputer, Acta Electronica Sinica, 29 (2001) 577-580
3. Wang, S.: A New Development on ANN in China - Biomimetic Pattern Recognition and Multi weight Vector Neurons. Lecture Notes In Artificial Intelligence, **2639** (2003) 35-43
4. Yang, W., Yu, L., et al.: A Fast Iris Location Algorithm. Computer Engineering and Applications , **10** (2004)
5. Wang, Y., Zhu, Y., Tan, T.: Biometrics Personal Identification Based on Iris Pattern. Acta Automatica Sinica, **28** (2002) 1-10
6. Li, M., et al.: Local Intensity Variation Analysis for Iris Recognition. Pattern Recognition , **37** (2004) 1287-1298.
7. Han, F., Chen, Y., Lu, H.: An Effective Iris Location Algorithm. Journal of Shanghai University (Natural Science) , **7** (2001) 1- 3
8. Cao, W., Feng, H., Wang, S.: The application of DBF Neural Networks for Object Recognition. Inf. Sci. **160** (2004) 153-160

Robust Precise Eye Location by Adaboost and SVM Techniques

Xusheng Tang, Zongying Ou, Tieming Su, Haibo Sun, and Pengfei Zhao

Key Laboratory for Precision and Non-traditional Machining Technology
of Ministry of Education, Dalian University of Technology
Dalian, Liaoning 116000, China
tribology@163.com, {ouzyg,tiemings}@dlut.edu.cn

Abstract. This paper presents a novel approach for eye detection using a hierarchy cascade classifier based on Adaboost statistical learning method combined with SVM (Support Vector Machines) post classifier. On the first stage a face detector is used to locate the face in the whole image. After finding the face, an eye detector is used to detect the possible eye candidates within the face areas. Finally, the precise eye positions are decided by the eye-pair SVM classifiers which using geometrical and relative position information of eye-pair and the face. Experimental results show that this method can effectively cope with various image conditions and achieve better location performance on diverse test sets than some newly proposed methods.

1 Introduction

Robust eye detection is a crucial step towards face recognition and man-machine interaction because these applications need to normalize faces or extract features according to eye positions [1]. Many eye detection algorithms have been proposed in recent years such as grayscale projection [2],[3] template matching [4],[5] and deformable template matching [6],[7] etc. They can achieve good result under some constraints; however, they all have some limitations. Grayscale projection depends on the lighting conditions. The drawback of template matching method is the initial position of the eye template has to be set approximately to the position of the eyes and their computations are complicated. In practice, eye patterns become quite complex due to various factors, such as variation of size, pose, and illumination conditions etc. Robustly and precisely locating eyes center is still a challenging task.

In this paper, a novel approach for precisely and fast locating eyes is devised. The proposed approach locates face areas using Adaboost face detector [8] prior to detecting eye location. After obtaining the position of face in the image, we can gain good initial constraints on the location of eyes. Then a followed Adaboost eye detector extracts eye candidates. Finally, precise positions of eye are decided by a SVM post classifier which is constructed using intrinsic geometrical relation information of eye-pairs and face. Due to the non-accidental properties of geometrical relation information, the eye-like eye candidates such as eyebrow can be excluded easily. The experimental results demonstrate that this approach can robustly cope with different light condition and achieve high detection rate on diverse test sets.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 93–98, 2005.

2 Proposed Method

We assume the people appear in the image whose head rotation 10° both in and out of image plane. The background is complex. Our system firstly detects the face in the image and then extracts the eye candidates within the upper part of face region using the method of [8]. Next, using a procedure proposed in this paper, the system extracts geometrical features of eye-pair candidates. Finally, precise eyes positions are decided by SVM post classifier using geometrical features. The training and detection process scheme for our system can be summarized as shown in Fig. 1.

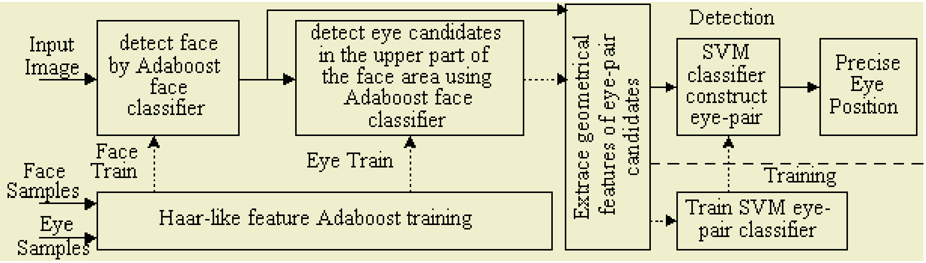

Input Image
detect face by Adaboost face classifier
detect eye candidates in the upper part of the face area using Adaboost face classifier
Extrace geometrical features of eye-pair candidates
Detection
SVM classifier construct eye-pair
Precise Eye Position
Face Train
Eye Train
Training
Face Samples
Eye Samples
Haar-like feature Adaboost training
Train SVM eye-pair classifier

Fig. 1. Training and detection of Hierarchical Adaboost detector with SVM as post classifier

2.1 Face Region Location

The problem of eye location is difficult since there are almost infinite numbers of specious objects with eye-like features in an input image. On the other hand, face detection is easier than eye detection as face pattern have more features than eye pattern. If we can localize the face in the input image the problem can be simplified due to the background being restricted to the face. It can also save searching time.

Viola's method [8] is used to detect face. The face classifier is obtained through supervised AdaBoost learning. Given a sample Set {xi,yi}, the AdaBoost algorithm selects a set of weak classifier {hj(x)} from a set of Haar-like rectangle features and combine them into a strong classifier. The strong classifier H(x) is defined as follow:

$$H(x)=\begin{cases}1 & \sum_j^T \alpha_j h_j(x) \geq \theta \\ 0 & otherwise\end{cases} . \tag{1}$$

The threshold θ is adjusted to meet the detection rate goal. The Haar-like rectangle features can be computed very quickly through a new image representation called "Integral Image". The "cascade" method can quickly filter out non-face image areas. More details can be found in [8].

Our face training set is drawn from FERET, ARData and ORL database. The 7098 face images are processed with gray scale normalization and size normalization to 20*20 pixels. In order to select the effective negative examples for targets detection, the bootstrapping method is used for selecting negative samples, that is, the false accept samples are later served as negative samples during the training process.

In detection process, face and non-face image regions can be discriminate according to Eq.(1).

2.2 Detect Eye Candidates

We extend the above approach to build an eye candidates detector. There are total 7000 eye samples with the eye center being the center of the image and resize to 16*8 pixels. The negative samples are non-eye images, which are cropped from the face image due to eye searching area being restricted to the face.

After the face region is located, we only search the upper area of the face region according to the prior knowledge about face model. The reduction of searching area can save the detection time and dramatically reduce the possible false eye candidates.

Since the eye detector based on Haar-like Feature only encode local gray intensive information of the eye, it is difficult to train an appropriate threshold to discriminate eye and eye-like images in the face regions such as eyebrows, thick frames of glasses, etc. So in this step we set low threshold to avoid false reject according to Eq. (1), that is, let eye candidates detector be with a relative high false accept rate and low false reject rate.

2.3 Precise Eyes Location Based on Eye-Pair SVM Post Classifier

Just as mentioned above, there are still some false eyes which cannot be discarded by hierarchical Adaboost classifier. Thus we propose a method that use SVMs technology to construct the eye-pair classifier based on geometrical model of face. Firstly, eye candidates are grouped into eye-pair candidates. Secondly, the true eye-pair can be selected from the eye-pair candidates by using the SVMs classifier. Finally, according to classifying results we can obtain the position of eye center. Due to the geometrical relation information encoded by SVMs, the disadvantage of eye detector based on Haar-like features can be overcome.

2.3.1 Geometrical Features and Training Data

The results of prior detection process are a set of eye candidates' rectangles and associated face rectangles. We assume the detected face rectangle is $W_f \times H_f$ pixels and randomly selecting two detected eye candidate rectangles are $W_i \times H_i$ pixels and $W_j \times H_j$ pixels respectively. We define W_{ij}, H_{ij} as the horizontal and vertical distance of center of two eye candidates respectively. These distances can be easily calculated based on the prior results of detection. The geometrical relations are shown in Fig.2

As we know, the sizes of two eyes are similar in the front-view face. The positions of two eyes in the face are symmetry. The direction and distance of the line joining the centers of both eyes are also very useful cue to select the true eye-pair. According to these prior information about geometrical face model, we choose L_i/H_f, L_j/H_f, W_{ij}/W_f, H_{ij}/H_f, H_{ij}/W_{ij}, W_i/W_f, H_i/H_f, W_j/W_f, H_j/H_f, L_{if}/W_f, L_{jf}/W_f, $|W_i-W_j|/W_f$, $|H_i-H_j|/H_f$ total 13 measurements for eye-pair classifying features.

For each image in the training set, we apply the following steps to get eye-pair SVM training data:

1. Applying Adaboost face and eye finder to the image, obtained a face rectangle R_f and a set of rectangles of eye candidates R_{ec}

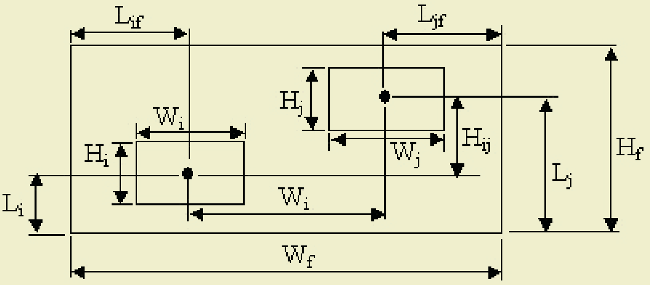


Fig. 2. Geometrical relation of Eye-pair Candidates

2. Selecting two eye candidates as eye-pair candidate $F_{\text{eye-pair}\,,\,j}$
3. For each eye-pair candidates and associate R_f, calculate 13 measurements of eye-pair classifying features and store them into a feature vector $x_{\text{eye-pair, j}}$. Manually labeling each $x_{\text{eye-pair, j}}$, obtaining labeled samples ($x_{\text{eye-pair, j}}$, y_j)

A training set of 400 images has been selected from the above-mentioned database. The 400 position samples and 2672 negative samples are obtained using this method.

2.3.2 Support Vector Machines

Let labeled samples (x_i,y_i) ,where $x_i \in R^{13}$, $y_i \in \{\pm 1\}$ and the kernel function K. SMO[9] is used as training algorithm. The Support Vectors s_j, the weight α_j and threshold b can be obtained by training. Then the decision function can be written as:

$$f(x) = sign(\sum \alpha_j y_j K(s_j, x) + b) \ . \tag{2}$$

where x is the input object to be classified.

As the numbers of negative samples are much more than positive samples during the training process, SVMs showed a tendency towards false rejection (FR) rather than false acceptance (FA). In order to compensate for that, we modify Eq.(2) with the introduction of a threshold θ. Then the decision function is written as follows:

$$f(x_i) = \begin{cases} 1 & (\sum \alpha_j y_j K(s_j, x) + b) \geq \theta \\ -1 & otherwise \end{cases} \ . \tag{3}$$

By varying θ we can adjust the performance of the SVMs. In this paper, we set θ to obtain equal error rate (EER) that is FA=FR.

2.3.3 Precise Eyes Location by SVM Eye-Pair Post Classifier

In practical detection, after eye candidates have been extracted by face and eye finder, any two eye candidates are grouped as eye-pair candidates. Then eye-pair geometrical features vectors can be obtained by the method described in above section. According to Eq. (3), the position of the eye-pair' eye-center classified as 1 is considered as the position of the eye center of the face. If more eye-pair candidates are classified as 1, then the average position of theirs is considered as the eye center.

3 Experimental Results

Two open face databases, BioID [12] and JAFFE [13], are used in our experiments. The BioID database consists of 1521 images. The JAFFE database consists of 213 images.

A relative error measure [10] is used to evaluate the precision of eye localization. Let d_l, d_r be Euclidean distance between the detection left(right) eye position and the marked left(right) eye position respectively, d_{lr} be Euclidean distance between the marked left and right eye. Then the relative error is defined as follows:

$$err = \max(d_l, d_r) / d_{lr} \ . \tag{4}$$

Our system is tested on AthlonXP 2500+ PC with 512MDDR memory. We adopt two criterion measures, err<0.1 and err<0.25, to evaluate the detection precision. The results are given in Table1. By saying, *err<0.1* means the deviation values is about smaller than 7 pixels on both databases. Based on the work by [10], if *err<0.25*, the eyes were deemed to found.

Some examples of eyes location are shown in Fig.3. From Fig.3 we can see that the system appears to be robust to the presence of unbalanced illumination, eyeglasses, partial occlusion and even significant pose variation. This generalization ability is consequence of the combination of Adaboost classifier using local features and SVM post classifier using geometrical features.

The average time is less than 56ms on both databases. The quick performance of our system is due to the efficiency of Adaboost classifier and the hierarchical structure which allows the non-eyes of image areas to be quickly discarded while spending more computation on promising eye-like regions.

We also compared our system with other newly published systems. In Ref. [3], their correction rate was 97.18% and 94.18% on JAFFE and BioID respectively while err<0.25. In Ref.[11], the detection rate was 98.6% on JAFFE when err<0.1. But their method needs to train 4 Adaboost classifiers. The cost of computation is higher than ours. Experiment results show our system outperforms these previous published systems on both test sets.

Table 1. The results of eye detection on different test database

Data-base	Correction rate (err<0.1)	Correction rate (err<0.25)	Average time	Image size	Face detection rate
JAFFE	99.53%	100%	41.4ms	256*256	100%
BioID	91.782%	98.093%	55.2ms	384*286	99.605%

4 Conclusion

This paper has introduced a framework for precisely locating eyes in image with complex background. The proposed method performs the eye location by a hierarchy Adaboost classifiers and a SVM post eye-pair classifier. The hierarchy Adaboost classifiers can rapidly and effectively search the face and the eye candidates. The eye-

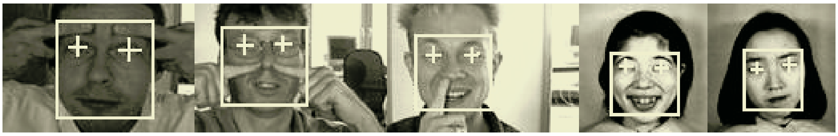


Fig. 3. Some eye locating results from test sets

pair SVM post classifier based on geometrical information can effectively locate the true eye position finally. Experiment results show that our method is fast, robust and reliable and can cope with the problems caused by diversity of illumination, pose variation, rotation and small occlusion. Though demonstrated on eye, the approach is also applicable to other facial organs detection such as mouth and nose.

Reference

1. Riopka, T.P. and Boult, T.: The Eyes Have it. In Proceedings of ACM SIGMM Multimedia Biometrics Methods and Applications Workshop. Berkeley, CA (2003) 9-16
2. Baskan, S., Bulut, M.M., Atalay, V.: Projection Based Method for Segmentation of Human Face and Its Evaluation. Pattern Recognition Letters, **23** (2002) 1623-1629
3. Zhou,Zhi Hua, Geng,Xin.: Projection Functions for Eye Detection. Pattern Recognition, **37**, (2004) 1049-1056
4. Huang, W.J., Yin, B.C., Jiang, C.Y., Miao, J.: A New Approach for Eye Feature Extraction Using 3D Eye Template. Proc. of ISIMP (2001) 340 – 343
5. Kawaguchi, T., Hikada, D., Rizon, M.: Detection of the Eyes From Human Faces by Hough Transform and Separability Filter. Proc. of ICIP (2000) 49-52
6. Lam, K.M., Yan, H.: Locating and Extracting the Eye in Human Face Images. Pattern Recognition, **29** (1996) 711–779
7. Huang, W.M., Mariani, R.: Face Detection and Precise Eyes Location. Proceedings of the 15th International Conference on Pattern Recognition, **4** (2000) 722–727
8. Viola, P., Jones, M.: Rapid Object Detection Using a Boosted Cascade of Simple Features. In Computer Vision and Pattern Recognition Conference 2001, **1** (2001) 511-518
9. Platt., J.: Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Technical Report MSR-TR-98-14, Microsoft Research (1998).
10. 10..Jesorsky, O. et. al.: Robust Face Detection Using the Hausdorff Distance. In: J. Bigun, F. Smeraldi (Eds.),Lecture Notes in Computer Science, 2091,Springer,Berlin (2001) 90-95
11. Yong Ma et. al.: Robust Precise Eye Location Under Probabilistic Framework. FGR(2004) 339-344
12. http://www.bioid.com/downloads/facedb/facedatabase.html
13. http://www.mis.atr.co.jp/~mlyons.jaffe.html

Classification-Based Face Detection Using Compound Features

Linlin Huang, Akinobu Shimizu, and Hidefumi Kobatake

Graduate School of BASE, Tokyo University of Agriculture and Technology
Nakacho 2-24-16, Koganeishi, Tokyo 184-8588, Japan
llhuang@cc.tuat.ac.jp

Abstract. In this paper, we propose a classification-based face detection method using compound features. Four kinds of features, namely, intensity, Gabor filter feature, decomposed gradient feature, and Harr wavelet feature are combined to construct a compound feature vector. The projection of the feature vector on a reduced feature subspace learned by principal component analysis (PCA) is used as the input of the underlying classifier, which is a polynomial neural network (PNN). The experimental results on testing a large number of images demonstrate the effectiveness of the proposed method.

1 Introduction

Face detection has been emerging as a very active research topic due to its numerous applications. The methods can be roughly divided into two groups: classification-based methods and feature-based approaches. Classification-based methods can benefit from learning from examples and have been demonstrated to be more robust than feature-based methods according to the reported results [1].

In classification-based methods, classifier design and feature extraction are the key problems of classification-based approaches. Many classification models have been proposed for face detection, such as Bayesian classifier [2], neural networks [3],[4], support vector machines [5], etc. As for the feature extraction, most classification-based methods have used the intensity values of the search window as the input features of the classifier. In order to further improve the detection performance, more effective features should be exploited.

In this paper, we propose a classification-based face detection method using compound features. Four kinds of features, namely, intensity, Gabor filter feature, decomposed gradient feature, and Harr wavelet feature are combined to construct a compound feature vector. The projection of the feature vector on a reduced feature subspace learned by principal component analysis (PCA) is used as the input of the underlying classifier, which is a polynomial neural network (PNN). The experimental results on testing a large number of images demonstrate the effectiveness of the proposed method.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 99–104, 2005.

2 Feature Representation Schemes

The performance of pattern classification largely depends on the feature representation scheme of input patterns. The combination of different features can yield improved performance. For face detection, we extracted four kinds of features from the local images: intensity representation [6], gradient feature representation [7], Gabor filter feature [8] and Haar wavelet decomposition [9], and construct a compound feature vector using the four kinds of features.

2.1 Intensity Representation

A simple representation scheme is to take the pixel intensity values of windowed local images directly as the inputs of classification. In our experiments, the size of local images is 20x20 pixels. The local image undergoes pre-processing (linear illumination correction, histogram equalization) to reduce the variance of feature values. After removing some corner pixels that do not compose face organs, 368 intensity values are stored in a feature vector [6].

2.2 Gradient Feature

In a face image, the direction of edges is a prominent and stable feature. In [7], we extracted gradient direction feature from windowed local images and have achieved superior detection performance. After pre-processing the local image, the gradient vector $u(x,y)=[u_x,u_y]^T$ is computed for each pixel using the Sobel operator. We then decompose the gradient vector into the components in eight chaincode directions Fig.2. The magnitude of $u_d(x,y)$ in *d* chain-code direction *d*=1, ..., 8 are assigned to the *d* directional plane (sub-image).

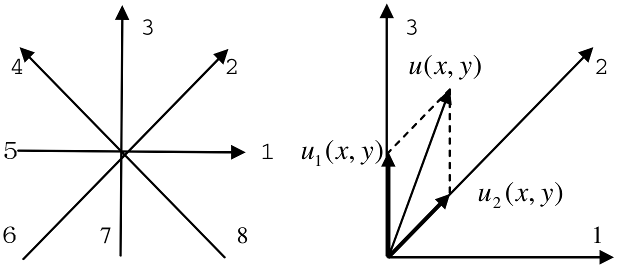


Fig. 1. Ecomposition of gradient vector

To generate a feature vector of moderate dimensionality, each pair of planes of opposite directions are merged into a single orientation plane and each plane is compressed to 10x10 pixels by block averaging. We select three stable orientation planes (the orientation plane from directions 1 and 5 is abandoned) and the compressed intensity image to construct a feature vector. After removing some corner pixels in each compressed plane/image, we obtain 368 feature measurements.

2.3 Harr Decompositions

Using 2D Haar wavelet transform coefficients as features, we have obtained promising results in face detection [9]. Motivated by the recent work of [2], we then combine the intensity values and horizontal/vertical projections with the wavelet transform coefficients to construct an enhanced feature vector.

Two types of 2D Haar basis functions that capture the changes of intensity along horizontal and vertical directions are employed. The basis function corresponding to diagonal changes is not used because it does not benefit the detection performance.

The basis functions are convoluted with the 20x20 intensity image to compute the transform coefficients. The basis functions have the size of 4x4 pixels and shift by 2 pixels. Consequently, each type of basis function gives 81 transform coefficients. Finally, the resulting 302D feature vector contains 162 transform coefficients, 20 horizontal projection values, 20 vertical projection values, and 10x10 compressed intensity values.

2.4 Gabor Filter Features

Gabor filters have been applied to various image recognition problems for feature extraction. The general functional form of a 2D Gabor filter is specified in space and spatial frequency domain as:

$$g(x,y) = \exp(-(\frac{x^2+y^2}{2\sigma_{xy}{}^2}))\cos(2\pi(u_0x+v_0y)) \cdot \tag{1}$$

$$G(u,v) = A\cdot\{\exp(-\frac{1}{2}(\frac{(u-u_0)^2}{\sigma_{uv}{}^2}+\frac{(v-v_0)^2}{\sigma_{uv}{}^2}))+\exp(-\frac{1}{2}(\frac{(u+u_0)^2}{\sigma_{uv}{}^2}+\frac{(v+v_0)^2}{\sigma_{uv}{}^2}))\}. \tag{2}$$

where σ_{xy} is the standard deviation of the Gaussian envelope, (u_0,v_0) defines the spatial frequency of a sinusoidal plane wave, with amplitude $r_0{}^2 = u_0{}^2 + v_0{}^2$, and orientation $\theta = \tan^{-1}(v_0/u_0)$. We use four Gabor filters with orientation $\theta_k = \pi(k-1)/4, k = 1,2,3,4$. Their frequency responses are depicted in Fig.3. The frequency band of each filter is a Gaussian centered at (u_0,v_0).

The Gabor features are extracted from the Gabor representations obtained by convolving the local image with the four Gabor filters. To encompass the orientation selectivities, we concatenate all the four Gabor representations and derive a feature vector. Before the concatenation, we first down-sample the Gabor representation by a factor of 2, then mask each down-sampled Gabor representation so as to remove the corner pixels since they do not compose of face organs.

In our work, the size of search window is 20x20 pixels. After down-sampled into 10x10, 8 corner pixels are excluded, finally we derive a 368D Gabor feature vector, which will be used as the input of the classifier.

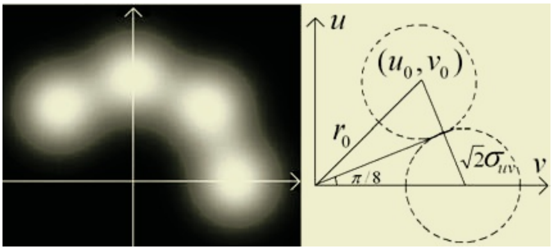


Fig. 2. our Gabor filters in frequency domain

2.5 Compound Features

After the four features are extracted, we then construct a compound feature vector by combining the four kinds of features. The dimensionality of the compound feature vector is 938D.

3 Classification Method

On each feature representation scheme, we use a polynomial neural network (PNN) to assign a face likelihood score to each windowed local image [6]. The PNN has one output unit with the linear and binomial terms of feature values as inputs. To reduce the complexity, the dimensionality of input feature vector is reduced by principal component analysis (PCA). The subspace parameters are estimated on a sample set of face images. Denoting the principal components (projections on subspace) of input feature vector **x** by $z_j, j = 1,...m$. The outputs of PNN is computed by

$$y(\mathbf{x}) = s(\sum_{i=1}^{m} w_i z_i + \sum_{i=1}^{m}\sum_{j=i}^{m} w_{ij} z_i z_j + w^D D_f + w_0). \tag{3}$$

where $s(\cdot)$ is the sigmoid function. D_f is the residual of subspace projection, also called distance from feature subspace (DFFS).

$$D_f = \|\mathbf{x} - \boldsymbol{\mu}\|^2 - \sum_{j=1}^{m} z_j^2. \tag{4}$$

In training PNN on face and non-face examples, the connecting weights are updated by gradient descent to minimize the mean square error. Since the PNN is a single layer network, the training process is very fast and the influence induced by the random initialization of weights is very less.

4 Experimental Results

For training the classifiers, 29,900 face samples were generated from 2,990 real face images by varying the size, aspect ratio and reflection. Non-face samples were collected using the bootstrap strategy in three stages. The samples collected in the first stage were the non-face windowed local images that have small Euclidean distance from the mean vector of face samples. In the second and third stages, the non-face samples were collected by the PNN trained with the face samples and the already collected non-face samples. Using the PNN, the non-face local images that have face scores greater than a threshold are collected. The images for collecting non-face samples are totally different from the test images.

We test the proposed method on two sets of images, which are totally different with those used in training procedure. Test Set1 consists of 109 images with cluttered images, containing a total of 487 faces. The 23 images collected by Sung and Poggio [4] are called Test Set2, which containing 149 faces.

We denote the four kinds of features as Inten, Gradient, Gabor, and Harr respectively. In our previous works, we have used the four kinds of features individually, and in this paper we want to investigate the effectiveness of the compound features, which is denoted as Com. Com_200 means the dimensionality of feature subspace learned by PCA is 200 and likewise Com_250, Com_300.

The 109 images are tested and the ROC curves are shown in Fig. 3. From the ROC curves we can see that the compound feature vector performs significantly better than the individual feature vectors. The ROC curves also show that the performance of Com_250 is superior to that of Com_200 and Com_300.

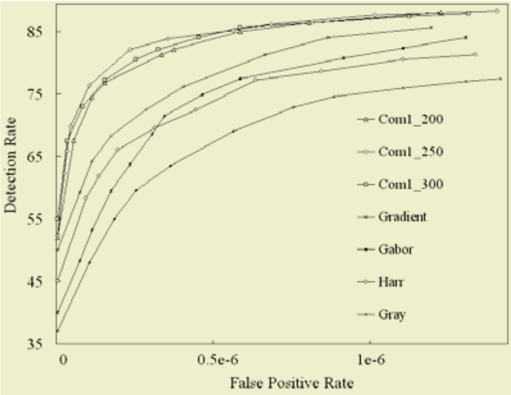


Fig. 3. Curves

The detection rates and false positive rates on Test Set1 and Set2 are listed in Table1 and Table 2. From Table 2, we can see that the proposed method performs better than the approach [4].

Table 1. Detection results on Test Set1.

	Detection rate	False positives	False positive rate
Com_200	87.6%	78	1.0×10^{-6}
Com_250	87.6%	71	0.9×10^{-6}
Com_300	87.6%	76	1.0×10^{-6}

Table 2. Detection results on Test Set2.

	True positives	Detection rate	False positives
Com_250	126	84.6%	10
[4]	126	84.6%	13

5 Conclusions

In this paper, we propose a classification-based face detection method which uses compound features as the input of the classifier. The classifier is a polynomial neural network in reduced feature subspace. The proposed method has been tested on a large number of images and the results demonstrated the effectiveness.

References

1. Hjelmas, R. and Low, B.K.: Face Detection: A Survey. Computer Vision and Image Understanding (2001) 236-274
2. Liu, C.: A Bayesian Discriminating Features Method for Face Detection. IEEE Trans. Pattern Anal. Mach. Intell (2003) 725-740
3. Rowley, H.A., Baluja, S. and Kanade, T.: Neural Network-based Face Detection. IEEE Trans. Pattern Anal. Mach. Intell. (1998) 23-28
4. Sung, K.K. and Poggio, T.: Example-based Learning for View-based Human Face Detection. IEEE Trans. Pattern Anal. Mach. Intell. (1998) 39-50
5. Osuna, E., Freund, R. and Girosi, F.: Training Support Vector Machines: An Application to Face Detection. Proc. IEEE Conf. CVPR (1997) 130-136
6. Huang, L.L., Shimizu, A., Hagihara, Y. and Kobatake, H.: Face Detection from Cluttered Images Using a Polynomial Neural Network, Neurocomputing (2003) 197-211
7. Huang, L.L., Shimizu, A., Hagihara, Y. and Kobatake, H.: Gradient Feature Extraction for Classification-based Face Detection Pattern Recognition (2003) 2502-2511
8. Huang, L.L., Shimizu, A. and Kobatake, H.: Classification-based Face Detection Using Gabor Filter Based Features. Proc. IEEE Conf. On Face and Gesture Recognition (2004) 397-402
9. Tokunaga, H., Huang, L.L., Shimizu, A., Hagihara, Y. and Kobatake, H.: Facial Characteristics Extraction Using Wavelet Transform and Its Application to Face Detection. Proc. Forum on Information Technology, Japan, **31** (2002)

Face Recognition Using RBF Neural Networks and Wavelet Transform

Bicheng Li[1,2] and Hujun Yin[1,*]

[1] School of Electrical and Electronic Engineering, University of Manchester
Manchester, M60 1QD, UK
[2] Information Engineering Institute, Information Engineering University
Zhengzhou, Henan 450002, China
lbclm@263.net, h.yin@manchester.ac.uk

Abstract. Recently, wavelet transform and image fusion mechanism have been used in face recognition to improve the performance. In this paper, we propose a new face recognition method based on wavelet transform and radial basis function (RBF) fusion network. Firstly, an image is decomposed with wavelet transform (WT) to three levels. Secondly, the Fisherface method is applied to three low-frequency sub-images respectively. Then, the individual classifiers are fused using the RBF neural network. Experimental results show that the proposed method outperforms both individual classifiers and the direct Fisherface method.

1 Introduction

Within the last decade, face recognition (FR) has become one of the most challenging areas in computer vision and pattern recognition. The popular appearance-based approaches are the eigenface method [1] and Fisherface method [2]. The eigenface method is built on principle component analysis (PCA). Fisher linear discriminant analysis (FLD or LDA) maximises the ratio of the determinant of between-class and within-class scatter matrices. The Fisherface method is used in conjunction with PCA and LDA.

Recently, wavelet transform (WT) has been widely used in face recognition [3-5]. Lai [4] and Chien [5] exploited the approximation of image in WT. Kwak [3] applied the Fisherface method to four subimages of the same resolution respectively, and then fused the individual classifiers with fuzzy integral [6]. In fact, high-frequency subimages provide little valuable information for recognition and themselves alone give poor performances, while low-frequency subimages can achieve good performance. Furthermore, different resolution images contain auxiliary information. In this paper, we propose a method based on WT and radial basis function (RBF) networks [7-8]. Firstly, an image is decomposed with WT to three levels. Then, the Fisherface method is applied to three low-frequency subimages respectively. Finally, the individual classifiers are fused by a RBF network, which is versatile and computationally simple. Efficient algorithms exist for training the RBF networks [7].

[*] Corresponding author

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 105–111, 2005.

To test the algorithm we used the FERET face database [9]. Experimental results show that the proposed method outperforms both the individual classifiers and the Fisherface method. Section 2, 3 and 4 give brief overviews on the RBF networks, wavelet transform and the Fisherface method. The proposed face recognition method is proposed in Section 5, followed by the results and conclusions in Section 6.

2 Radial Basis Function Neural Networks

In the RBF networks [7-8], the hidden layer plays an important role as an interface between the networks and their environment. They are built of locally tuned respective fields called radial basis functions. Generally, a RBF network is a feedforward architecture consisting of three layers as shown in Fig. 1. The hidden layer is composed of the respective fields, where nonlinear effect of neurocomputing lies, while the output layer is a collection of some linear processing units.

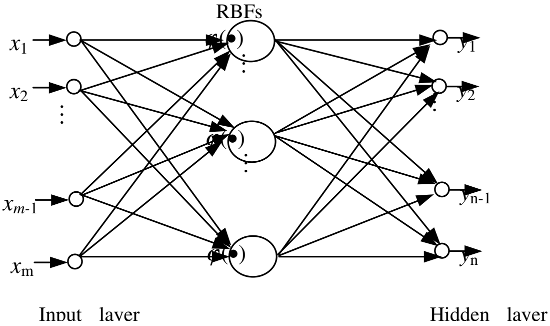


Fig. 1. RBF Neural Network

The output of the hidden layer is given by

$$z_i = \varphi\left(\frac{\|X - U_i\|}{\sigma_i^2}\right), \qquad i = 1,2,\cdots,k . \tag{1}$$

where k is the number of RBFs. φ is a positive radically symmetric function centered at U_i and distributed according to the spread defined by σ_i. X is an input vector i.e. $X = (x_1, x_2, \cdots, x_m)^T$. Both X and U_i are defined in R^m. The centers U_i are also considered to be the prototypes of the respective fields.

The output layer is a linear combination of the activation levels of the hidden layer, i.e.

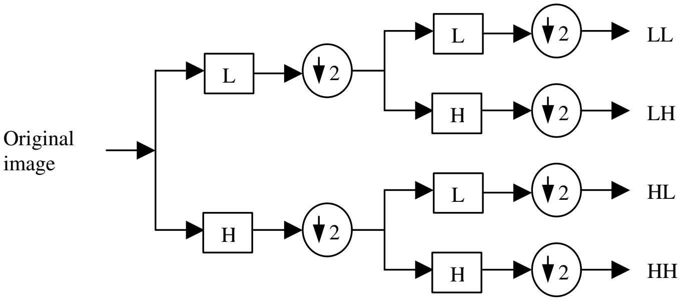
$$y_j = \sum_{i=1}^{m} w_{ji} z_i + b_j \qquad j = 1,2,\cdots,n . \tag{2}$$

where n is the number of output nodes. b_j is the bias of jth output node.

Gaussian is widely used as the kernel functions in the RBF network. The hidden layer can be trained separately using for example a clustering algorithm to choose the centres and widths for the hidden kernels. More optimised algorithms such as the orthogonal least square (OLS) [10] can be used to optimally selecting centres from the data sample. A heuristic search method has been used in optimise the width for each of the candidate centres [11].

3 Wavelet Decomposition

Wavelet transform (WT) [12-14] is a multiresolution analysis tool and it decomposes a signal into different frequency bands. We use the two-dimensional (2-D) WT to efficiently decompose images, generating (at level 1) an approximate subimage and three high frequency subimages via the high-pass and low-pass filtering realised with respect to the column vectors and the row vectors of array pixels. Fig. 2 shows the structures of the filter bank that performs the decomposition at level 2. Here, H and L represent the high-pass and low-pass filter respectively. In this way we can obtain four subimages (LL1, LH1, HL1, and HH1). In a similar manner, we can repeatedly decompose the approximation subimages obtained, such as two-level (LL2, LH2, HL2, and HH2), three-level (LL3, LH3, HL3, and HH3), and so forth.



4 Fisher Method

The Fisherface method [2] is a combination of PCA with FLD, and it consists of two stages. First, PCA is used to project face pattern from a high-dimensional image space into a lower-dimensional space. Then, FLD is applied to the feature-vectors.

Suppose a face image is a 2-D $p \times q$ array. An image Z_i is considered as a column vector of dimension $M = p \times q$. A training set consisting of N face images is denoted by $Z = \{Z_1, Z_2, \cdots, Z_N\}$. The covariance matrix is defined as: $R = \frac{1}{N}\sum_{i=1}^{N}(Z_i - \overline{Z})(Z_i - \overline{Z})^T$, where $\overline{Z} = \frac{1}{N}\sum_{i=1}^{N} Z_i$. Let $E = (E_1, E_2, \cdots, E_r)$ consists of the r eigenvectors corresponding to the r largest eigenvalues of R. We project original face images into a PCA-transformed space, and obtain their corresponding reduced feature vectors $Y = \{Y_1, Y_2, \cdots, Y_N\}$ as follows:

$$Y_i = E^T Z_i . \tag{3}$$

In the second stage, FLD is applied to feature vectors, $Y = \{Y_1, Y_2, \cdots, Y_N\}$. Suppose the training set of N face images consists of C classes. The between-class scatter matrix is defined by

$$S_B = \sum_{i=1}^{C} N_i (\overline{Y}^{(i)} - \overline{Y})(\overline{Y}^{(i)} - \overline{Y})^T . \tag{4}$$

where N_i is the number of samples in *i*th class C_i, $\overline{Y}$ is the mean of all samples, and $\overline{Y}^{(i)}$ is the mean of *i*th class C_i. The within-class scatter matrix is defined as,

$$S_W = \sum_{i=1}^{C} \sum_{Y_k \in C_i} (Y_k - \overline{Y}^{(i)})(Y_k - \overline{Y}^{(i)})^T . \tag{5}$$

The optimal projection matrix is chosen as a matrix with orthonormal columns that minimizes the determinant of within-class scatter matrix while maximizing that of between-class scatter matrix, i.e.

$$W_{FLD} = \arg\max_W \frac{|W^T S_B W|}{|W^T S_W W|} = [W_1, W_2, \cdots, W_{C-1}] . \tag{6}$$

where $\{W_1, W_2, \cdots, W_{C-1}\}$ is the set of generalised eigenvectors of S_B and S_W corresponding to the $C-1$ largest generalised eigenvalues $\{\eta_1, \eta_2, \cdots, \eta_{C-1}\}$, that is

$$S_B W_i = \eta_i S_W W_i , \ \ i = 1, 2, \cdots, C-1 . \tag{7}$$

Because S_B is the sum of C matrices of rank one or less, the rank of S_B is $C-1$ or less. The feature vectors for any face images $V = (V_1, V_2, \cdots, V_N)$ can be obtained by projecting Y onto FLD-transformed space:

$$V_i = W_{FLD}^T Y_i = W_{FLD}^T E^T Z_i , \qquad i = 1, 2, \cdots, N . \tag{8}$$

For a new face image, the classification is realised in FLD-transformed space; that is, it is transformed into its feature vector using Eq. (8), and then is compared with the feature vectors of the training images.

In the Fisherface method, the upper bound on the number of eigenvectors r selected is $N-C$, which corresponds to the rank of the within-class scatter matrix S_B. To determine r, the eigenvalues of the covariance matrix R, λ_i, $i = 1, 2, \cdots, M$, are ranked in non-increasing order. We choose $r\,(\le N-C)$ as the minimum of m such that

$$\sum_{i=1}^{m} \lambda_i \Big/ \sum_{i=1}^{M} \lambda_i \geq P \,. \tag{9}$$

where P is a given constant.

5 Face Recognition Using Wavelet Transform and RBF Fusion

The proposed face recognition algorithm consists of three stages. Suppose the training set of N face images $Z = \{Z_1, Z_2, \cdots, Z_N\}$ consists of C classes.

Firstly, an image is decomposed with WT to three levels, obtaining 3 approximation images LL1, LL2, and LL3. Hence we obtain three subsets of images. In this paper, we use the Daubechies family [13] wavelet filters db*N*, where *N* stands for the order of the wavelet. The choice of 3 levels in WT is determined experimentally, and it is a compromise between recognition performance and computation complexity. More levels result in only slightly better result, while require more computation.

Secondly, the Fisherface method is applied to the decomposed three subsets respectively, obtaining 3 classifiers: classifier1, classifier2, and classifier3. In PCA, we choose the number of eigenvectors using Eq. (9) with $P = 95\%$, instead of the commonly used $N - C$ [3]. We obtain the feature vectors of the training images and a given test image using Eq. (8) and calculate Euclidean distances between the feature vector produced from a given test image and those from the training image set. These Euclidean distances are transformed into membership grades as follows:

$$\eta_{ij} = \frac{1}{1 + (d_{ij} \big/ \overline{d}_i)} \,,\ i = 1,2,3 \,,\ j = 1,2,\cdots,N \,. \tag{10}$$

where *i* is the index of three classifiers and *j* is the index of the training face image. $\overline{d}_i$ denotes the average distance between all distance values in *i*th classifier, d_{ij} is the Euclidean distance between the feature vector of a given test image and that of *j*th training image in *i*th classifier. Furthermore, the membership in which a given test image belongs to the *k*th class in *i*th classifier is computed by,

$$\mu_{ik} = \frac{1}{N_k} \sum_{X_i \in C_k} \eta_{ij} \ ,\ \ i = 1,2,3 \,,\ k = 1,2,\cdots,C \,. \tag{11}$$

where C_k is the set of samples in *k*th class, N_k is the number of samples in C_k. For *i*th classifier, if the memberships of a given test image satisfy:

$$\mu_{ik_0} = \max_{k=1}^{C} \mu_{ik} \,. \tag{12}$$

then the given test image is recognised to belong to k_0 th class.

Finally, the individual classifiers are fused by an RBF network. For each test image, the membership *C* obtained from each classifier using Eq. (11) constitutes a membership vector. The three membership vectors derived from three classifiers results are concatenated into a feature vector of size 3*C*. The RBF network is designed to use these feature vectors as the inputs and to fuse individual classifiers to generate improved classification results. The input layer has 3*C* nodes and the output has *C* nodes, while the number of nodes in the hidden layer is defined experimentally. The class corresponding to the node with the maximum output in the output layer is declared to the result of the recognition.

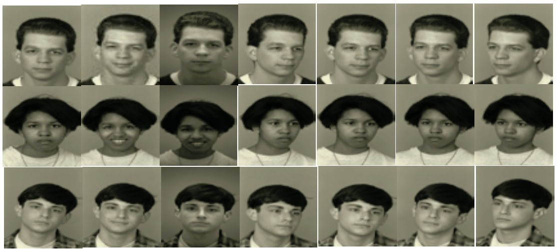


Fig. 3. Samples of three subjects

6 Experimental Results and Conclusions

To test our method, we selected 40 subjects from the "b" series of the FERET database [9], and the indexes of the subjects are from 1141 to1180. For every subject, we select 7 images whose letter codes are ba, bj, bk, bd, be, bf, and bg. They are eight-bit greyscale images of human heads with views ranging from frontal to left and right profiles (pose angle $\leq 25\ ^0$) and different illumination. The original image of size 384×256 is reduced to 128×128. Some samples of 3 subjects are shown in Fig. 3.

For a given n ($n = 4{,}5{,}6$), n images are chosen randomly as the training set and the rest for testing in each run. The recognition is executed for 100 runs, and the mean and standard deviation (Std.) of the performance are calculated.

We used db4 [13] to decompose images into 3 levels with boundary effect being considered. Each approximation subimage of different levels, i.e. LL1, LL2, and LL3, is extracted. The sizes of LL1, LL2, and LL3 are 67×67, 37×37 and 22×22, respectively. The Fisherface method with *P*=95% is applied to the three approximation subimages. A 3-layer RBF network is adopted to fuse the three classifiers. The sizes of the input layer, the hidden layer and the output are 120, 120 and 40, respectively, and the spread is chosen as unit. Table1 lists the results of various methods, i.e. the Fisher method for original database, classifier1, classifier2, classifier3, and fusion with a RBF network.

Experimental results show that the proposed method outperforms individual classifiers and the direct Fisherface method. In addition, more training samples dose not always mean higher accuracy rate. For example, if we choose 4, 5 and 6 images from each subject for training, the accuracy rate of the proposed method are 94.33%, 94.25%, and 94.67%, respectively, while the highest accuracy rate of the three classifiers are 93.71%, 92.05%, and 93.65%, respectively.

Table 1. Results (%) for LL1, LL2, LL3, fusion with RBF, and original iamges. *n* indicates the number of training images used from each subject.

n	Statistics	LL1	LL2	LL3	fusion with RBF	Original images
4	Mean	91.41	92.75	93.71	94.33	88.74
	Std	2.25	1.97	2.12	1.84	2.75
5	Mean	91.00	92.05	91.85	94.25	90.30
	Std	2.77	2.70	2.59	2.42	3.30
6	Mean	92.57	93.50	93.65	94.67	91.83
	Std	3.77	3.34	3.44	3.25	4.06

References

1. Turk, M.A., Pentland, A.P.: Eigenfaces for Recognition. J. Cogn. Neurosci., 3 (1991) 71–86
2. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. IEEE Trans. Pattern Anal. Machine Intell., 19 (1997) 711–720
3. Kwak, K.-C., Pedrycz, W.: Face Recognition Using Fuzzy Integral and Wavelet Decomposition Method. IEEE Trans System, Man, and Cybernetics -B, 34 (2004) 1666–1675
4. Lai, J.H., Yuen, P.C., Feng, G.C.: Face Recognition Using Holistic Fourier Invariant Features. Pattern Recognition, 34 (2002) 95–109
5. Chien, J.T., Wu, C.C.: Discriminant Wavelet Faces and Nearest Feature Classifiers for Face Recognition. IEEE Trans. Pattern Anal. Machine Intell., 24 (2002) 1644–1649
6. Sugeno, M.: Fuzzy Measures and Fuzzy Integrals—A Survey. in Fuzzy Automata and Decision Processes. Gupta, M.M., Saridis, G. N. and Gaines, B.R., Eds. Amsterdam, The Netherlands: North Holland, (1977) 89–102
7. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall International, Inc. New Jerey, 1999
8. Pedrycz, W.: Computational Intelligence: An Introduction. CRC Press, New York (1998)
9. FERET face database, http://www.itl.nist.gov/iad/humanid/feret/
10. Chen, S., Cowan, C.F.N., Grant, P.M.: Orthogonal least squares learning algorithm for radial basis function networks. IEEE Trans. Neural Networks, 2 (1991) 302-309.
11. Moller-Levet, C.S., Yin, H.: Modelling and Clustering of Gene Expressions Using Rbfs and a Shape Similarity Metric. Intelligent Data Engineering and Automated Learning 2004, LNCS-3177, (2004) 1-11.
12. Mallat, S.: A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. IEEE Trans. Pattern Anal. Mach. Intell., 11 (1989) 674–693
13. Daubechies, I.: Orthonormal Bases of Compactly Supported Wavelets. Commun Pure Appl Math, 41 (1988) 909–996.
14. Cohen, A., Daubechies, I., Feauveau, J.: Biorthogonal Bases of Compactly Supported Wavelets. Commun. Pure Appl. Math., XLV (1992) 485–560.

Face Recognition Using Fisher Non-negative Matrix Factorization with Sparseness Constraints*

Xiaorong Pu, Zhang Yi, Ziming Zheng, Wei Zhou, and Mao Ye

Computational Intelligence Laboratory, School of Computer Science & Engineering
University of Electronic Science & Technology of China
Chegndu, Sichuan 610054 China
{puxiaor,zhangyi}@uestc.edu.cn

Abstract. A novel subspace method is proposed for part-based face recognition by using non-negative matrix factorization with sparseness constraints (NMFs) and Fisher's linear discriminant (FLD) hence its abbreviation, FNMFs. A comparative analysis engages PCA+FLD (FPCA) method and FNMFs method for both part-based and holistic-based face recognition. The comparative experiments are completed for the ORL face database and UMIST face database, it shows that FNMFs has better performance than FPCA-based method both for holistic-face and parts-face images recognition.

1 Introduction

Face recognition from images is one of the common methods for identifying an individual. The fundamental problem in face recognition is to find a suitable representation of the facial image. Most face recognition systems learn to represent a face as a linear combination of basis images, which are based on either part or holistic face images [1], [2].

Non-negative matrix factorization with sparseness constraints(NMFs) can find parts-based representations of face images [10], [11] by decomposing a non-negative face image matrix into both sparse and non-negative basis and encoding coefficient. As a classic method for face recognition, Principal component analysis (PCA) is also applied for matrix factorization to 'eigenface' basis, which is arbitrary sign and resembles the distorted versions of whole faces [2], [3]. In fact, there is psychological and physiological evidence for parts-based representations in brain [4],[5]. Therefore it is essential to study the methods of face recognition based on parts of its face images.

The primary objective of this paper is to present an novel and efficient NMFs-FLD aggregate, i.e., FNMFs framework for face recognition, by combining NMFs

* This work was supported by National Science Foundation of China under Grant 60471055, Specialized Research Fund for the Doctoral Program of Higher Education under Grant 20040614017, and the UESTC Youth Fund under Grant L08010601JX04030.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 112–117, 2005.

method and the Fisher's Linear Discriminant (FLD) which is applied to find an optimal subspace for classification in which the ratio of the between-class scatter and the within-class scatter is maximized [6], [7]. In this paper, the comparative performance of the proposed approach and the method of the combination of PCA and FLD [8], [9] on the face recognition problem is studied. The experimental data come from the ORL face database [12] and UMIST face database [13].

This paper is organized as follows. In Section 2, we provide a brief review of the PCA+FLD method in the context of finding the 'eigenface' basis images and classifying facial images of each individual in a training set. Section 3 describes the NMFs approach. Section 4 describes our new approach FNMFs for face recognition. Experiments are carried out in Section 5. Finally, conclusions are given in Section 6.

2 Face Recognition Using FPCA

In this section, we will present brief description of the FPCA-based face recognition method, in which the Principal Component Analysis (PCA) is firstly used for dimensionality reduction in face compression and the Fisher's Linear Discriminant (FLD) is then used for classifying a specific linear projection.

At the beginning, we elaborate on the problem of matrix factorization for dimensionality reduction as addressed by the PCA method, by which calculates the eigenvectors and eigenvalues of the covariance matrix of the original face images. Let a face image database be a non-negative matrix $V_{n\times m}$, each column of which contains n pixel values (≥ 0) of one of the m facial images. Their corresponding eigenface-based features B can be obtained by projecting V into the eigenface subspace as

$$B = U^T \varGamma \tag{1}$$

where $\varGamma$ is the set of zero-mean image in the original face image set V. $U = (U_1, U_2, \ldots, U_r)(r < m)$ is the set of the "r" orthonormal eigenvectors corresponding to the "r" largest orthogonal eigenvalues $\{\lambda_i | i = 1, 2, \ldots, r\}$. For further details, refer to [3].

The Fisher's Linear Discriminant (FLD) is applied to the projection of the set of original face images in the eigenface space $B = (B_1, B_2, \ldots, B_m)$ to find an optimal subspace for classification in which the ratio of the between-class scatter and the within-class scatter is maximized [8],[9]

$$D_{FLD} = arg \max_{D} \frac{|D^T S_B D|}{|D^T S_W D|} = [d1 \; d2 \; \ldots \; d_\mu]^T \tag{2}$$

where the size of D_{FLD} is $\mu \times r(\mu \leq r)$ and $\{d_i | i = 1, 2, \ldots, \mu\}$ is the set of discriminant vectors of S_B and S_W corresponding to the c-1 largest generalized eigenvalues $\{\lambda_i | i = 1, 2, \ldots, c-1\}$,i.e., $S_B d_i = \lambda_i S_W d_i (i = 1, \ldots, c-1)$, where $S_B = \sum_{i=1}^{c} N_i (b_i - \overline{b})(b_i - \overline{b})^T$ is the between-class scatter matrix, N_i is the number of samples in the ith class C_i, and c is the number of classes, $\overline{b} =$

$\frac{1}{m}\sum_{j=1}^{m} b_j$ is the mean image of the ensemble, and $b_i = \frac{1}{N_i}\sum_{j=1}^{N_i} b_j^i$ is the mean image of the ith class C_i. S_W is the within-class scatter matrix defined as $S_W = \sum_{i=1}^{c}\sum_{b_k \in C_i}(b_k - b_i)(b_k - b_i)^T = \sum_{i=1}^{c} S_{W_i}$, where S_{W_i} is the convariance matrix of class C_i.

Thus, the final feature vectors $X = [X_1\ X_2\ \ldots\ X_N]^T$ for training image set are calculated by

$$X_{train} = B_{train} D_{FLD}^T. \tag{3}$$

The feature vectors X_{test} for testing image set are computed as

$$X_{test} = B_{test} D_{FLD}^T. \tag{4}$$

3 Non-negative Matrix Factorization with Sparseness Constraints

Non-negative matrix factorization is a linear, non-negative approximate data representation[2], [11]. Give a data matrix $V_{n\times m}$, NMF refers to the decomposition of the matrix V into two matrices $W_{n\times r}(r < n)$ and $H_{r\times m}$, such that $V \approx WH$, where each column of V consists of n pixel values of one of the m facial images. The r columns of $W_{n\times r}$ are called basis images, and $H_{r\times m}$ is the matrix of encoding coefficients in one-to-one correspondence with a face in V. Unlike the orthogonality constraint of PCA, NMF does not allow negative entries in the matrix factors W and H.

Patrik [10] showed that NMF with sparseness constraints (NMFs) could lead to representations that were part-based and match the intuitive features of face images. The NMF with sparseness constraints (NMFs) is defined as follows [10]:

$$\min_{W,H} \quad D(V|WH) = \sum_{i,j}(V_{ij} - (WH)_{ij})^2 \tag{5}$$

$$s.t. \quad W, H \geq 0, \quad sparseness(w_i) = S_w, \forall i, \quad sparseness(h_i) = S_h, \forall i \tag{6}$$

The sparseness measure based on the relationship between the L_1 norm and the L_2 norm is defined by $sparseness(x) = \frac{\sqrt{n} - ||x||_1/||x||_2}{\sqrt{n} - 1}$, n is the dimensionality of x. Patrik proposed a projected gradient descent algorithm for NMF with sparseness constraints. For further details, refer to [10].

4 Face Recognition Using FNMFs

Similar to the PCA-based method, NMFs does not provide any information for class discrimination but finding the part-based representation by matrix factorization, thus the FLD is essential to classify the face images of an individual. We perform the FLD based on the encoding coefficient vectors H obtained from the NMFs method in the same manner as done for the FPCA-based method. The subspace of FNMFs $S_{train} = [S_1\ S_2\ \ldots\ S_t]^T (t \leq m)$ transformed by the FLD for the training image set is calculated as follows

$$S_{train} = H_{train}^T D_{FLD}^T. \tag{7}$$

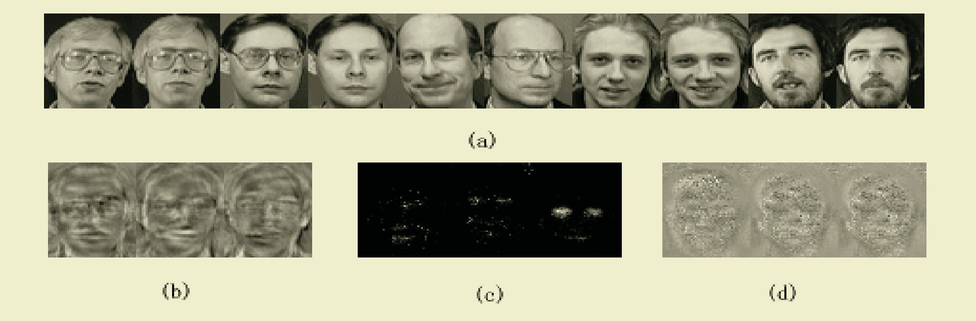


Fig. 1. (a) Some ORL faces (b) FPCA-faces (c)Some NMFs basis (d)FNMFs faces

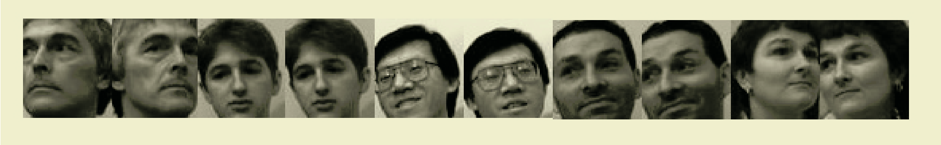


Fig. 2. Some cropped UMIST faces

For a new face image V_i in a test set, the corresponding encoding coefficient is calculated based on the basis W generated from the training set in the form

$$H^i_{test} = W^{\dagger} V_i \tag{8}$$

where $W^{\dagger}$ is the pseudoinverse of the basis matrix W. A $k-$nearest neighbor classifier is used to calculate the Euclidian distance $\epsilon = (\epsilon_1 \; \epsilon_2 \; \ldots \; \epsilon_t)$ between H^i_{test} and S_{train} as $\epsilon = \|S_{train} - H^i_{test}\|$. A test face image is classified as belonging to class q when the minimum distance ϵ_k is below some chosen threshold θ. Otherwise the face is “unknown”.

5 Experiments

Our experiments were carried out using FPCA and FNMFs approaches to compare their performance on the ORL face database and the UMIST face database. Among total ORL database, we specifically considered 240 frontal face images of 40 individuals. There were 6 images per subject including normal,facial expression, and light condition with tolerance for some tilting and rotation of up to 20, in which 5 face images per person in training set and 1 face image per person in testing set were selected randomly. Some of the cropped face images, eigen-fisher and NMFs-fisher faces are shown in Fig. 1. In our experiments for UMIST face database, 57 frontal images of 19 individuals were randomly selected as the training set, in which each person has three images, and another 76 images of the 19 individuals were randomly selected as the testing set, in which each person has four images. All of the images trained and tested were cropped, in which some of them were not the holistic face images Fig. 2. For ORL database, both the

Table 1. Performance comparisons of FPCA and FNMFs

	PCA + FLD		NMFs + FLD	
	Number of features	Recognition rate(%)	Number of basis vectors	Recognition rate(%)
ORL database	10	62.37	10	80.50
	40	95.38	40	95.75
	80	96.00	80	97.88
UMIST database	18	74.87	18	90.13
	22	81.45	22	90.66
	38	88.55	38	88.68

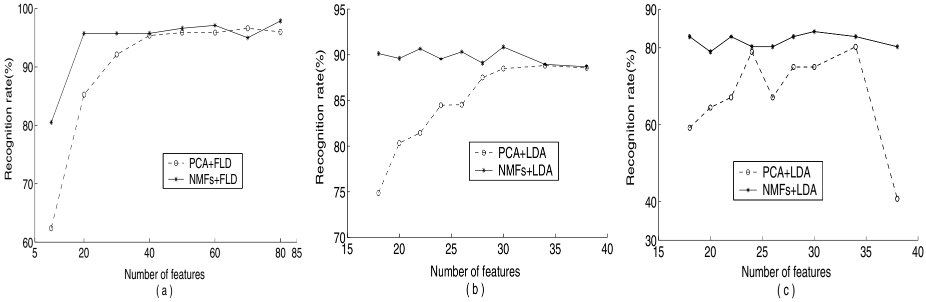


Fig. 3. (a)average recognition rate for ORL database (b)average recognition rate for UMIST database (c)minimum recognition rate for UMIST database

dimensionality '*r*' of FNMFs and the number of principle components to represent the feature vectors were set to: $10, 20, 30, 40, 50, 60, 70, 80$,which were set to: $18, 10, 22, 24, 26, 28, 30, 34, 38$ for UMIST database, respectively. The sparseness of the basis was fixed at 0.75. An average of 800 FNMFs runs were executed for each database. The number of features, basis vectors and the corresponding recognition rates for the two approaches for ORL database and UMIST database have been shown in Table 1. The average recognition rates for FPCA and FNMFs algorithms for the ORL database and UMIST database show in Fig.3(a) and (b), respectively. Fig. 3(c) presents the minimum recognition rate for the two algorithms for the UMIST database, which indicates that the FNMFs algorithm is less sensitive to the selection of training and testing patterns than the FPCA algorithm.

6 Conclusions

The FNMFs algorithm is developed in this paper to recognize the part-based face images. The experiments demonstrate the superiority of the proposed algorithm in human face recognition in comparison with FPCA-based method. The recognition rates of 97.88% for the ORL database and 90.86% for the UMIST database are achieved using the FNMFs algorithm. These recognition rates are clearly higher than that of using FPCA approach. The experiments also show that the FNMFs algorithm is less sensitive than the FPCA algorithm to the selection of training and testing patterns.

Acknowledgment

The authors wish to thank Zeng Yonglin and Jia Jingxi for their useful discussions and comments.

References

1. Jain, A. K., Ross, A., Prabhakar S.: An Introduction to Biometric Recognition. IEEE Transactions on Circuits and Systems for Video Technology, Vol. **14**, No. 1, January 2004.
2. Lee, D. D. and Seung, H. S.: Learning the Parts of Objects by Non-negative Matrix Factorization. Nature, **401** (1999) 788-791
3. Turk, M.A., Pentland, A.: Eigenfaces for Recognition. Journal of Cognitive Neurosience, **3** (1991) 71-86
4. Wachsmuth, E., Oram, M. W., Perrett D. I.: Recognition of Objects and Their Component Parts: Responses of Single Units in the Temporal Cortex of the Macaque, Cereb. Cortex, **4** (1994) 509-522
5. Li, S.Z., Hou, X., Zhang, H., and Cheng, Q.: Learning Spatially Localized Parts-based Representations. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Hawaii, USA **I** (2001) 207-212
6. Belhumeur, P.N., Hespanha, J.P., and Kriegman, D.J.: Eigenfaces versus Fisherfaces: Recognition Using Class Specific Linear Projection. IEEE Trans. Pattern Anal. Machine Intell., **19** (1997) 711-720
7. Swets, D.L., Weng, J.: Using Discriminant Eigenfeatures for Image Retrieval. IEEE Trans. Pattern Anal. Machine Intell., **18** (1996) 831-836
8. Yambor, W.S.: Analysis of PCA-Based and Fisher Discriminant-Based Image Recognition Algorithms. Technical Report CS-00-103, Computer Science Department, Colorado State University, July (2000)
9. Martinez, A.M., Kak, A.C.: PCA versus LDA. IEEE Transactions on Pattern Analysis and Machine Intelligence, **23** (2001) 228-233
10. Patrik, O.H.: Non-negative Matrix Factorization with Sparseness Constraints , Journal of Machine Learning Research **5** (2004) 1457-1469
11. Paatero, P., Tapper, U.: Positive Matrix Factorization- A Non-negative Factor Model with Optimal Utilization of Error Estimates of Data Values. Environmetrics, **5** (1994) 111-126
12. The ORL database is available from http://www.cam-orl.co.uk/facedatabase.html
13. The UMIST database: http://images.ee.umist.ac.uk/danny/database.html

Gabor Features-Based Classification Using SVM for Face Recognition

Yixiong Liang[1], Weiguo Gong[1], Yingjun Pan[1],
Weihong Li[1], and Zhenjiang Hu[2]

[1] MOE Key Lab of Optoelectronic Technology & Systems
Chongqing University, Chongqing 400044, China
{yxliang,wggong,pyj,weihongli}@cqu.edu.cn
[2] Department of Mathematical Informatics, University of Tokyo
Tokyo 113-0033, Japan
hu@ipl.tu-tokyo.ac.jp

Abstract. In this paper we present a novel Gabor-SVM method for face recognition by integrating the Gabor wavelet representation of face images and SVM classifier. Gabor wavelets first derive desirable facial features characterized by spatial frequency, spatial locality and orientation selectivity to deal with the variations due to illumination and facial expression changes. The principal components analysis (PCA) method is then used to reduce the dimensionality of the extracted Gabor features. With the reduced Gabor features, SVM is trained and then employed to do the recognition tasks. The performance of Gabor-SVM method is compared with the standard PCA-NC (*Eigenfaces*) method and PCA-SVM method on a subset of AR face database. The experiment results demonstrate the efficiency and superiority of the proposed Gabor-SVM method.

1 Introduction

Within the last decades, face recognition has been a very hot topic and great progress has been made in this field. As a result,numerous novel algorithms have been proposed for face recognition [1]. Two issues are central to all these algorithms. The first is the representation, namely how to extract the discriminant features for face representation. The other is matching, i.e. how to classify a new face image using the chosen representation. The Gabor wavelets, whose kernels are similar to the response of the two-dimensional receptive field profiles of the mammalian simple cortical cell, exhibit the desirable characteristics of spatial frequency, spatial locality and orientation selectivity [2], [3] and are widely used as the feature extractor for face recognition [4], [5], [6]. The recent research [7] reveals that the Gabor representation of face image give better performance than other techniques such as PCA and LDA. Support vector machine is a relatively new classification system rooted in the statistical learning theory [8]. It is considered as state of the art two-class classifiers because it delivers high performance in real world applications. In nature, a SVM finds an optimal separating hyperplane which minimizes the risk of misclassifying not only the training samples

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 118–123, 2005.

but also the testing samples. Some successful applications of the SVM to face recognition have been reported recently [9], [10], [11].

Prompted by that, in this paper we introduces a novel Gabor-SVM method for face recognition. The Gabor-SVM method first convolves a face image with a set of Gabor wavelets and then applies an elliptical mask to the filtered images to remove the unwanted parts. Whereafter the wrapped images are normalized to zero mean and unit variance, downsampled and concatenated to derive an augmented Gabor feature vector. This augmented feature vector encompasses all the information produced by different Gabor wavelets and so its dimension is very high. For computer efficiency, PCA is applied to the augmented Gabor features for dimensionality reduction. The reduced Gabor features are employed as the input of SVM. Because SVM is originally developed for two-class classification, this basic scheme is extended for multi-class face recognition problems by adopting *one against one* method.

2 Gabor Feature Exaction

This section details the derivation of the desired Gabor features. In the spatial domain, 2D Gabor function is a complex exponential modulated by a Gaussian function, which has been used to model the receptive field of the orientation-selective simple cells [2]:

$$G(x,y) = \frac{1}{2\pi\sigma\beta} e^{-\pi[\frac{(x-x_0)^2}{\sigma^2} + \frac{(y-y_0)^2}{\beta^2}]} e^{i(\xi x + \nu y)}, \tag{1}$$

where (x_0, y_0) is the center of the receptive field in the spatial domain and (ξ, ν) is the optimal spatial frequency of the filter in the frequency domain, σ and β are the standard deviations of the elliptical Gaussian along x and y, which determine the width of the Gaussian window. Mathematically, the above Gabor function achieves the resolution limit in the conjoint space only in its complex form. Based on the Gabor function (1), Lee [3] derives 2D Gabor wavelets with their parameters properly constrained by neuropsychological data on simple cells and by the wavelet theory:

$$\begin{aligned}\Psi(x,y,\omega,\theta) = \tfrac{\omega}{\sqrt{2\pi}k} & e^{-\frac{\omega^2}{8k^2}[4(x\cos\theta + y\sin\theta)^2 + (-x\sin\theta + y\cos\theta)^2]} \\ & \cdot [e^{i\omega(x\cos\theta + y\sin\theta)} - e^{-\frac{k^2}{2}}]\end{aligned}, \tag{2}$$

where ω is the radial center frequency and θ is the wavelet orientation in radians. k is a constant, with $k \approx \pi$ for a frequency bandwidth of one octave and $k \approx 2.5$ for a frequency bandwidth of 1.5 octaves. With a set of ω and θ, one can derive a set of Gabor wavelets (or Gabor filters). Note that those Gabor wavelets are self-similar in that they can be generated by rotation and dilation (affine group) from the following mother wavelet:

$$\Psi(x,y) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{8}[4x^2 + y^2]} [e^{ikx} - e^{-\frac{k^2}{2}}]. \tag{3}$$

The derivation of the Gabor features of the image $I(x, y)$ can be easily performed by convolving the image with a set of Gabor wavelets, namely

$$G_{ij} = \Psi(x, y, \omega_i, \theta_j) * I(x, y), \tag{4}$$

where $*$ is the convolution operator and G_{ij} denotes the convolution result corresponding to the Gabor wavelet at frequency ω_i and orientation θ_j. In our experiments, we select $\omega_i = 2^{-\frac{(i+2)\pi}{2}} (i = 1, 2, 3)$ and $\theta_j = j\frac{\pi}{8} (j = 0, \ldots, 7)$. It should be noted that the regions on the two sides of the chin (i.e. background) are usually not important but the magnitude of them maybe significant. Thus we apply an elliptical mask which is constructed manually to each G_{ij} to remove these regions. A visualized example is shown in Fig.1. Then we can transform the wrapped convolution result G_{ij} to a feature vector GF_{ij} by concatenating all its rows. Before concatenating, each wrapped image is first downsampled by a factor ρ, and then normalized to zero mean and unit variance. To utilize the result that encompasses information of different spatial frequencies, spatial localities and orientations, we concatenate all resultant GF_{ij} into an augmented Gabor face vector GF:

$$GF = (GF_{10}^T, \ldots, GF_{37}^T)^T, \tag{5}$$

where T is the transpose operator. As the augmented Gabor feature GF considers multi-scales and multi-orientations, its dimension is still very high even though the downsampling procedure is applied to reduce the dimension of GF. Our solution to this problem is using PCA, a powerful technique for reducing a large set correlated variables to a smaller number of uncorrelated components, for further dimensionality reduction. The resultant features are then propagated to a set of SVM classifiers to perform the recognition tasks.

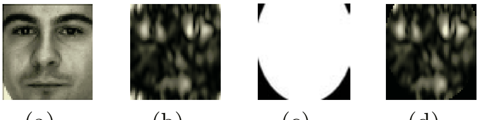


Fig. 1. Example of the Gabor representation of a face image.(a) the original image; (b) G_{10}; (c)Elliptical mask; (d) Wrapped G_{10}

3 Recognition with SVMs

It's well knows that SVM is basically to construct a binary classifier from a set of patterns called training samples, which are available prior to classification. It performs the classification between two classes by mapping the samples into another dot space via kernel tricks and then finding a hyperplane that has maximum distance to the closest points in the training set which are termed as support vectors. Thus the hyperplane is optimal in the sense that it separates the largest fraction of points form each class, while maximizing the distance from either class to the hyperplane. This optimal hyperplane can be computed as a decision surface of the form:

$$f(x) = sgn(\sum_i y_i \alpha_i K(x_i^*, x) + b), \tag{6}$$

where $sgn(\cdot)$ denotes the sign function, x_i^* is the set of support vectors and $K(\cdot, \cdot)$ is the predefined kernel function that satisfies Mercer's condition [8].The coefficients α_i and b can be determined by solving the following large-scale quadratic programming (QP) problem:

$$\min[\sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j)], \tag{7}$$

$$s.t. \sum_i \alpha_i y_i = 0, \tag{8}$$

$$0 < \alpha_i < C, \forall i. \tag{9}$$

The unbound C is the penalty parameter that represents the tradeoff between minimizing the training set error and maximizing the margin. In our experiments, the following Gaussian kernel with parameter $\sigma = 1.0$ is used:

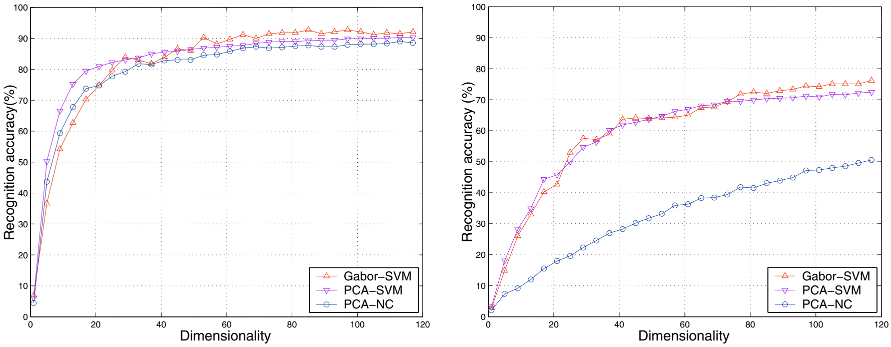
$$K(x, y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}. \tag{10}$$

Since the SVM is a binary classifier, it should be extended for c-class face recognition tasks. There are two basic strategies for solving this problem. The first is *one against all* approach, which needs to train c SVM classifiers and each classifier separates a single class from all remaining classes. The second is *one against one* approach, which is a pairwise approach and need to train $c(c-1)/2$ SVM classifiers. Because the former often leads to ambiguous classification [10], we adopt the latter in the experiments.

4 Experimental Results

In order to establish the performance of Gabor-SVM, we carried out two experiments on a subset of AR face database [12]. This subset contains 1652 frontal images (14 images of 118 individuals) with changes in slight facial expression (*smile* and *angry*) and illumination conditions, and images taken in two sessions two weeks apart. We select two approaches for comparision: (1)the classical PCA-NC (namely $Eigenfaces$) as a benchmark and (2) PCA-SVM. To begin, we manually localize the positions of the two eyes and then convert the RGB images to gray scale ones by simply averaging all three-color channels. Later, each image is scaled, translated, and rotated so that the eyes are located in the fixed positions. We also apply the histogram matching technology to images as photometric normalization. For computation efficiency, we crop all images to a size of 53×56.

In the first experiment, we aim to compare the proposed Gabor-SVM with PCA-NC and PCA-SVM under varying in expression. For each person, two *neutral expression* images which are taken from different session are used for training, the probe set comprises two *angry* images and two *smile* images for each person. Thus we have 236 training samples and 472 testing images. Fig.2(a) shows the recognition accuracy under a varying number of selected features. One can see that when more than 50 features are used for recognition, the proposed Gabor-SVM performs the best and obtains a highest recognition accuracy 92.71%. This curves indicate that the Gabor representation of images carries more discriminant information.



(a) (b)

Fig. 2. Comparative recognition performance of Gabor-SVM, PCA-SVM, and PCA-NC on the subset of AR face image database. (a) Under the condition of variations in facial expression. (b) Under the condition of variations in illuminance

The second experiment is performed to evaluate the performance of Gabor-SVM method under the condition of variations in illuminance. We still use the *neutral expression* images for training but six illuminance varying images per person for testing. In total, we have 708 testing images. The results are shown in Fig.2(b).It's obvious that those methods using SVM classifier (both Gabor-SVM and PCA-SVM) perform better than the method using nearest center classifier (PCA-NC). We postulate that this is due to the strong classification ability of SVM. Again, the proposed Gabor-SVM obtains the best performance when more than 74 features are used.

5 Conclusion

We have developed a novel Gabor-SVM method for face recognition that unifies Gabor representation and SVM classifier. Gabor wavelets first derive desirable facial features which are insensitive to the variance of illumination and facial expression. Then the Gabor features are reduced by PCA and the resultant features are then propagated to a set of SVM classifiers to perform the recognition tasks. The experimental results demonstrate that the performance of the proposed Gabor-SVM method is overall superior to those obtained by PCA-NC or PCA-SVM methods.

Acknowledgements

This work is supported by the scientific technology key project (02057) and Chunhui project (2003589) of Ministry of Education, China.

References

1. Zhao, W., Chellappa, R., Rosenfeld, A., Phillips, P. J.: Face Recognition: A Literature Survey. ACM Computing Survey, **35** (2003) 399–458
2. Daugman, J.G.: Uncertainty Relation for Resolution in Space, Spatial Frequency, and Orientation Optimized by Two-dimensional Visual Cortical Filters. Journal of the Optical Society of America A, **2** (1985) 1160–1169
3. Lee, T.S.: Image Representation Using 2D Gabor Wavelets. IEEE Transactions on Pattern Analysis and Machine Intelligence, **18** (1996) 959–971
4. Lades, M., Vorbruggen, J.C., Buhmann J., Lange, J., Malsburg, C.V.D., Wurtz, R.P., Konen, W.: Distortion Invariant Object Recognition in the Dynamic Link Architecture. IEEE Transactions on Computers, **42** (1993) 300–310
5. Wiskott, L., Fellous, J.M., Kruger, N., Malsburg, C.V.D.: Face Recognition by Elastic Bunch Graph Matching. IEEE Transactions on Pattern Analysis and Machine Intelligence, **19** (1997) 775–779
6. Liu, C.J.: Gabor-based Kernel PCA with Fractional Power Polynomial Models for Face Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, **26** (2004) 572–581
7. Donato, G., Bartlett, M.S., Hager, J.C., Ekman, P., Sejnowski, T.J.: Classifying Facial Actions. IEEE Transactions on Pattern Analysis and Machine Intelligence, **21** (1999) 974–989
8. Vapnik, V.N.: Statistical Learning Theory. John Wiley and Sons, New York (1998)
9. Phillips, P.J.: Support Vector Machines Applied to Face Recognition. In: Mozer, M.C., Jordan, M.I., Petsche, T. (eds.): Advances in Neural Information Processing Systems, Vol. 11, MIT Press (1999) 803–809
10. Guo, G.D., Li, S.Z., Chan, K.L.: Support Vector Machines for Face Recognition. Image and Vision Computing, **19** (2001) 631–638
11. Deniz, O., Castrillon, M., Hernandez, M.: Face Recognition Using Independent Component Analysis and Support Vector Machines. Pattern Recognition Letters, **24** (2003) 2153–2157
12. Martinez, A.M., Benavente, R.: The AR Face Database. Technical Report **24**, CVC (1998)

An Experimental Evaluation of Linear and Kernel-Based Classifiers for Face Recognition

Congde Lu[1], Taiyi Zhang[1], Wei Zhang[2], and Guang Yang[3]

[1] Department of Information & Communication Engineering
Xi'an Jiaotong University, Xi'an, Shaanxi 710059, China
congdelu@mailst.xjtu.edu.cn

[2] School of Electrical Engineering and Computer Science, Oregon State University
Corvallis, OR 97331, USA

[3] Department of Communication, Xi'an Institute of Post and Telecommunication
Xi'an Shaanxi 710059, China

Abstract. This paper presents the results of a comparative study of linear and kernel-based methods for face recognition. We focus mainly on the experimental comparison of classification methods, i.e. Nearest Neighbor, Linear Support Vector Machine, Kernel based Nearest Neighbor and Nonlinear Support Vector Machine. Some interesting conclusions can be obtained after all of these methods are performed on two well-known database, i.e. ORL, YALE Face Database, respectively.

1 Introduction

As a sub-field of pattern recognition, face recognition is widely used to many fields, such as identity certificating, monitor system, machine vision, etc. A perfectly face recognition system mainly includes three parts: preprocessing, feature extraction and face classification. This paper concentrates mainly on the experimental comparison of classification methods.

These classification algorithm presented in this paper is divided into two categories: linear methods, such as Nearest Neighbor (NN) [1], Linear Support Vector Machine (LSVM) [2] and kernel based methods, such as Kernel based Nearest Neighbor (KNN) [3], Nonlinear Support Vector Machine (NSVM) [2] etc.. Some comparative results for face recognition methods had been published [4][5], but they only obtained a experimental comparison between feature extraction algorithms, whereas didn't consider the comparison between classification algorithms. In addition, these comparative results were only performed a single database, such as literature [5] only used UMIST face database, So some conclusions obtained in these paper have not generalization.

The objective of this paper will mainly focus on two aspects: (1) ones compare the performance of linear classification algorithms and one of kernel-based classification algorithms on both the same feature extraction algorithm and the same face database; (2) ones display the performance of the different combination of feature extraction with classification algorithm on different face database. In addition, we also obtain some conclusions on the basis of comparing the performance of linear methods and kernel based classifiers.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 124–130, 2005.

2 Classification Methods

There exist a lot of classification methods so far, such as LSVM, NN etc. These classification algorithms may be generalized to feature space by a nonlinear function, such as NSVM, KNN and so on. We will describe briefly this classification algorithms before they are evaluated by the experiments.

2.1 NN and KNN Algorithms

Assuming there exist c classes $w_1, w_2, \cdots, w_c$ in a pattern recognition problem, where the number of per class is $N_i (i = 1, 2, \cdots, c)$, then we may specify the discriminant function of w_i as

$$g_i(x) = min_k \parallel x - x_i^k \parallel, k = 1, 2, \cdots, N_i, \tag{1}$$

where the subscript i of x_i^k is label of w_i, the superscript k of x_i^k is the kth sample of w_i.

According to the equation (1), the discriminant criteria of the nearest neighbor method represents as follows:

if

$$g_j(x) = min_i g_i(x), i = 1, 2, \cdots, c, \tag{2}$$

then $x \in w_j$.

We have an intuitive interpret: for unlabel sample x, we can take action that x has the same label with the nearest neighbor sample from it on the basis of comparing the distance between x and $N = \sum_{i=1}^{c} N_i$ label samples.

Consider a case of mapping an $n-$dimension feature space to an $m-$dimension feature space:$\phi : x \rightarrow \phi(x), x \in X, \phi(x) \in F$, where X is the original $n-$dimension feature space and F is the new $m-$dimension feature space. x is an arbitrary vector in X, $\phi(x)$ is the corresponding vector in F. ϕ can be an arbitrary nonlinear mapping from the original space X to a possibly high-dimensional space F, and $\phi_i, i = 1, 2, \cdots, m$, are feature mapping functions.

A kernel denotes a function K, such that for all $x, y \in X$

$$K(x, y) =< \phi(x), \phi(y) >, \tag{3}$$

where $< \phi(x), \phi(y) >$ denotes the inner product of $\phi(x)$ and $\phi(y)$, $K(x, y)$ is a function of x and y.

In conventional nearest-neighbor algorithm, a norm distance metric, such as Euclidean distance, is often used. By redefining the distance metric, the 'kernel approach' relies on the fact that we exclusively need to compute inner products between the mapped samples. Since inner products are available in Hilbert space only, norm distance metrics in Hilbert space are concerned here.

The norm distance $d(x, y)$ between any two vectors x and y is denoted by $d(x, y) = \|x - y\|$. Suppose the nearest-neighbor algorithm is used in a high dimensional feature space, norm distance in such a space should be computed.

Thus a feature mapping can be applied. The square of norm distance in the image feature space can be obtained by applying the 'kernel approach'.

It is trivial to prove that the square of norm distance in Hilbert space can be expressed by inner products. By decomposition of $d^2(\phi(x), \phi(y))$ into inner products and substitution of (3) for the inner products, we have

$$d^2(\phi(x), \phi(y)) = K(x, x) - 2K(x, y) + K(y, y). \tag{4}$$

Thus norm distance in the new image feature space can be calculated by using a kernel function and the input vectors in the original feature space. When we compute kernel norm distance and apply the nearest-neighbor algorithm in the image feature space, we get a kernel nearest-neighbor classifier.

2.2 LSVM and NSVM Algorithms

SVM is one of the most excellent classification algorithms, especially adapted in small sized, high dimensional samples [2].

Given a linearly separable training sample $S = ((x_1, y_1), \cdots, (x_l, y_l))$, the hyperplane (w, b) that solves the optimization problem

$$min_{w,b} < w, w >, \tag{5}$$

subject to $y_i(< w, x_i > +b) \geq 1, i = 1, 2, \cdots, l$ realizes the maximal margin hyperplane with geometry margin $\gamma = \frac{1}{\|w\|_2}$.

We now consider how to transform this optimization problem into its corresponding dual problem. The primal Lagrangian is

$$L(w, b, \alpha) = \frac{1}{2} < w, w > - \sum_{i=1}^{l} \alpha_i [y_i(< w, x_i > +b) - 1], \tag{6}$$

where $\alpha_i \geq 0$ are the Lagrange multipliers.

The corresponding dual is found by differentiating with respect to w and b, imposing stationary,

$$\begin{cases} \dfrac{\partial L(w, b, \alpha)}{\partial w} = w - \displaystyle\sum_{i=1}^{l} y_i \alpha_i x_i = 0 & (7) \\ \dfrac{\partial L(w, b, \alpha)}{\partial b} = \displaystyle\sum_{i=1}^{l} y_i \alpha_i = 0 & (8) \end{cases}$$

and resubstituting the relation obtained, i.e. $w = \sum_{i=1}^{l} y_i \alpha_i x_i$ and $0 = \sum_{i=1}^{l} y_i \alpha_i$, into the primal, then we can obtain

$$\begin{aligned} L(w, b, \alpha) &= \frac{1}{2} < w, w > - \sum_{i=1}^{l} \alpha_i [y_i(< w, w_i > +b) - 1] \\ &= \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j < x_i, x_j > . \end{aligned} \tag{9}$$

Therefore, we may achieve the optimal classification function as follows.

$$f(x) = sgn\{< w^*, x > +b^*\} = sgn\{\sum_{i=1}^{n} \alpha_i^* y_i < x_i, x > +b^*)\}. \qquad (10)$$

How can the above methods be generalized to the case where the decision function is not a linear function of the data? Now suppose we first mapped the data to some other (possibly infinite dimensional) Euclidean space H, using a mapping which we will call ϕ: $\phi : R^d \rightarrow H$. Then the training algorithm would only depend on the data through dot products in H, i.e., on functions of the form $< \phi(x_i), \phi(x_j) >$. Now if there were a "kernel function" K such that $K(x_i, x_j) =< \phi(x_i), \phi(x_j) >$, we would only need to use K in the training algorithm, and would never need to explicitly even know what ϕ is. So, when the decision function is not a linear function of the data, the optimal classifier is represented as follows:

$$f(x) = sgn\{< w^*, \phi(x) > +b^*)\} = sgn\{\sum_{i=1}^{n} \alpha_i^* y_i K(x_i, x) + b^*\}. \qquad (11)$$

3 Experimental Results and Analysis

In order to compare the performance of linear with kernel based methods for face recognition, all experiments are performed on two publicly available face databases, i.e. ORL [6] and YALE [7] face databases respectively. For a full comparison of these classifiers, we make use of different features, which can be obtained by different feature extraction algorithms, i.e. Principal Component Analysis (PCA) [8], Kernel Principal Component Analysis (KPCA) [9], Fisher Discriminant Analysis (FDA) [10] and Kernel Discriminant Analysis (KDA) [11].

3.1 Experiments on ORL Face Database

ORL database consists of 400 face images from 40 different peoples with 10 images per person. The original images were all sized 92 × 112 with a 256-level gray scale. For keeping the essential feature of ORL face database, we have only a simple preprocessing, i.e. normalized to zero mean and unit variance.

In this experiment, 5 chosen randomly images for each person were used in the training set for a total of 200 training images, and 5 chosen randomly images for each person were used in the test set for a total of 200 test images. The training set did not have overlap with the test set. Error recognition rate of combining different feature extraction algorithms and different classification algorithms on ORL and YALE face database were displayed in Fig. 1 and Fig. 2, respectively. In Fig. 1 and Fig. 2, different feature extraction algorithms, i.e. PCA, KPCA, FDA, KDA, RAW (represents raw data) are displayed on horizontal axis in turn, and the recognition error rate obtained using NN, KNN, LSVM, NSVM, respectively are presented in turn as the histogram according to the gray-scale from black to white.

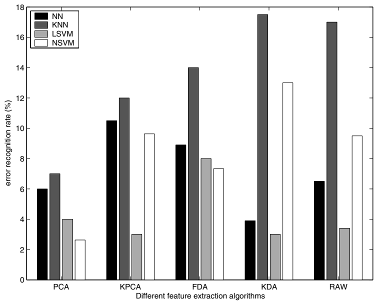


Fig. 1. Error recognition rate of combining different feature extraction with different classification algorithms on ORL database

In Fig.1, we extracted 140 principal components with the best performance in PCA and KPCA for having a contrastivity. Gaussian kernel with $\sigma = 0.05 * D$ is used as the kernel function of NSVM, while Gaussian kernel with $\sigma = 10$ is selected as the kernel function of KNN, where D is the dimensionality of samples (the same below). In addition, Gaussian kernel with $\sigma = 0.002 * D$ is chosen as the kernel function of KDA when Gaussian kernel with $\sigma = 0.004 * D$ is selected as the kernel function of KPCA.

For PCA algorithm, the best performance of 2.5% error rate is obtained when we utilized NSVM to classify. For KPCA algorithm, the best performance of 3% error rate was obtained when we utilized LSVM to classify. and the classification method with the worst performance is KNN classifier in both cases.

For FDA algorithm, in comparison with NN, KNN and LSVM, NSVM obtains the best performance with 7% error recognition rate. For KDA algorithm, the classifier with the best performance is LSVM in comparison with NN, KNN and NSVM, and its error rate is 3.4%.

According to the above results, we found that the lowest error rate can be obtained when we used the combination of linear algorithms for feature extraction and kernel based methods for classification, such as PCA+KNN, FDA+NSVM, or the combination of kernel based methods for feature extraction and linear algorithms for classification, such as KDA+NN, KPCA+LSVM. In addition, Both the combination of linear algorithms for feature extraction with linear algorithms for classification and the combination of kernel based methods for feature extraction with kernel based algorithms for classification can't be obtain better performance, such as KPCA+KNN etc..

3.2 Experiments on Yale Database

Yale database consists of 165 face images from 15 different peoples with 11 images per person.

The original images were all sized 96 × 116 with a 256-level gray scale, but the experiments adopt these images which their size is 28 × 29 by sampling for improving the speed of computation. For keeping the essential feature of

YALE face database, we have only a simple preprocessing, i.e. these images were normalized to zero mean and unit variance.

Yale database include a variable number of images for different people, but same number of training and test images for each person were chosen so that the results are not affected by number of images of any particular person. 10 runs of each experiment were performed and the results were averaged. In each experiment, 5 chosen randomly images for each person were used in the training set, and 5 chosen randomly images for each person were used in the test set. The training set did not have overlap with the test set. In Fig.2, we extracted 60 principal components with the best performance in PCA and KPCA. Gaussian kernel with $\sigma = 0.008*D$ is used as the kernel function of NSVM, while Gaussian kernel with $\sigma = 10$ is selected as the kernel function of KNN, where D is the dimensionality of samples (the same below). In addition, Gaussian kernel with $\sigma = 0.007 * D$ is chosen as the kernel function of KDA when Gaussian kernel with $\sigma = 0.006 * D$ is selected as the kernel function of KPCA.

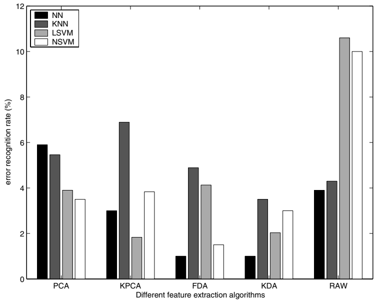


Fig. 2. Error recognition rate of combining different feature extraction with different classification algorithms on YALE database

For PCA algorithm, the best performance of 3.5% error rate was obtained when we utilized NSVM to classify, while the classifier with the best performance of 4.65% error rate is LSVM for KPCA. In addition, for FDA algorithm, both LSVM and NSVM classifiers can obtain the best performance with 1.33% error rate, whereas the classifier with the best performance is NSVM.

From the above experiments, we have some conclusions as follows:

(1) LDA/KDA outperform PCA/KPCA because of the inherent discrimination ability of the former, but there exists a singularity on ORL database.

(2) Although kernels enable us to work in a high dimension feature space, they do not always ensure better performance. Introduction of kernels might even degrade the performance if the problem is close to linearly separable. This conclusion is similar to one in the literature [5].

(3) The LSVM/NSVM classifiers outperform NN/KNN algorithms for all of feature extraction algorithms, but the computational cost NN/KNN is lower in comparison with LSVM/NSVM.

(4) Both “linear feature extraction algorithm + kernel classification method” and “kernel feature extraction algorithm + linear classification algorithm” can obtain lower error rate, whereas “kernel feature extraction algorithm + kernel classification method” can’t obtain the better performance.

4 Conclusions

We obtain some preparatory conclusions by comparing linear algorithms with kernel based algorithms in the experiments, such as both “linear feature extraction algorithm + kernel classification method” and “kernel feature extraction algorithm + linear classification algorithm” can obtain lower error rate, whereas “kernel feature extraction algorithm + kernel classification method” can’t obtain the better performance in general. In addition, kernel methods are not always better than traditional linear methods for a special recognition problem. So, we think that kernel method isn’t a replacement of linear method, but is a complement of it.

References

1. Cover, T. M., Hart, P.E.: Nearest Neihbor Patter Classification. IEEE Trans. Information theory IT-13 (1967) 21-27
2. Vapnik, V. N.: The Nature of Statistical Learning Theory. Springer-Verlag, Berlin Heidelberg New York (2000)
3. Yu, K., Ji, L., Zhang, X. G.: Kernel Nearest-Neighbor Algorithm. Neural Processing Letters **15** (2002) 147-156
4. Li, J., Zhou, S. H., Shekhar, C.: A Comparison of Subspace Analysis for Face Recognition. Proceedings of the 2003 IEEE International Conference on Acoustics, Speech & Signal Processing III (2003) 121-124
5. Gupta, H., Agrawal, A. K., Pruthi, T., et al.: An Experimental Evaluation of Linear and Kernel-Based Methods for Face Recognition. Proceedings of the Sixth IEEE Workshop on Applications of Computer Vision (2002) 13-18
6. ORL database, available from:
   http://www.uk.research.att.com:pub/data /att_faces.tar.Z.
7. Yale database, available from:
   http://cvc.yale.edu/projects/yalefaces /yalefaces.html.
8. Turk, M., Pentland, A.: Eigenfaces for recognition. J. Cogn.Neurosci. **3** (1991) 71-86
9. Schölkopf, B., Smola, A., Müller, K. R.: Nonlinear Component Analysis as a Kernel Eigenvalue Problem. Neural Computation **10** (1998) 1299-1310
10. Etemad, K.: Discriminant Analysis for Recognition of Human Face Images. Journal of the Optical Society of American A **14** (1997) 1724-1733
11. Liu, Q. S.: Face Recognition Using Kernel Based Fisher Discriminant Analysis. Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition (FGR’02) (2002) 187-191

A Study on Illumination Invariant Face Recognition Methods Based on Multiple Eigenspaces

Wujun Li[1], Chongjun Wang[1], Dianjiang Xu[2], Bin Luo[1], and Zhaojian Chen[1]

[1] National Laboratory for Novel Software Technology
Nanjing University, Nanjing , Jiangshu 210093, China
{liwujun,chjwang}@ai.nju.edu.cn
[2] Department of Computer Science
North Dakota State University, Fargo ND58105, USA
Dianxiang.xu@ndsu.nodak.edu

Abstract. This paper presents two multiple illumination eigenspaces-based methods, RDEB and BPNNB, for solving the variable illumination problem of face recognition. The experiment shows that the methods have a high recognition ratio. In particular, BPNNB has outperformed the assumptive method which knows the illumination directions of faces and completes recognition in the specific eigenspace using eigenface method[2] for each face subset with a specific illumination direction

1 Introduction

Face recognition is to identify or verify one or more persons in the given still or video images of a scene using a stored database of faces [1]. With the change of illumination condition or pose, the performance of a face recognition system may decrease significantly [3].

To address the variable illumination issue, we have developed a multiple illumination eigenspaces method (called NNEB) [5]. Based on NNEB, this paper proposes a residual description error based method, called RDEB, and a Back-Propagation neural network based method, called BPNNB. These methods, evaluated with the IAE face database [5], have demonstrated a high recognition ratio. In particular, the BPNNB method outperformed an assumptive method which knows the illumination directions of faces and completes recognition in the specific eigenspace using eigenface method [2] for each face subset with a specific illumination direction.

2 Multiple Illumination Eigenspaces

Assume that the face images in a training set, say *X*, can be divided into a number of subsets, say $\{X1,X2,...,XI\}$, according to the illumination directions when images are imaged, i.e. $X=\{X1,X2,...,XI\}$, where $Xi=\{x_{i1},x_{i2},...,x_{in_i}\}$, *I* is the number of different illumination directions, x_{ij} denotes the *jth* image corresponding to the *ith* illumi-

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 131–136, 2005.

nation direction, and n_i is the number of face images corresponding to the *ith* illumination direction. We can build *I* eigenspaces [2], each describing a region of the face space that is corresponding to a specific imaging illumination direction. Thus, we can get the corresponding eigenspace set, $S=\{S_1,S_2,...,S_I\}$, where S_i stands for an eigenspace corresponding to the *ith* illumination direction. If *I > 1*, we have multiple illumination eigenspaces.

3 The Residual Description Error Based (RDEB) Method

The RDEB-method is an extension to the multiple eigenspaces method in [4]. First, multiple illumination eigenspaces are trained as described in section 2. Second, the illumination direction of the target object is determined by selecting the eigenspace which best describes the input image. This is accomplished by evaluating the residual description error [2] using each illumination space's eigenvectors. Once the proper illumination direction is determined, the face image is projected into that eigenspace, and then recognized. The RDEB-method is outlined as follows:

(1) Build a set of eigenspaces: If the number of illumination directions is *I*, we have a set of *I* illumination eigenspaces denoted as $S=\{S_1,S_2,...,S_I\}$.

(2) Extract the set of feature vectors of the face images in the gallery: Each image in the gallery G={G1,G2,...,GI} (Gi stands for the subset of gallery with an illumination direction *i*) is projected into the corresponding eigenspace of a specific illumination direction. So we can extract the set of feature vectors denoted as $C=\{C_{g1},C_{g2},...,C_{gI}\}$, in which $C_{gi}=\{c_{gi_1},c_{gi_2},...,c_{gi_{ni}}\}$ is a subset of C with an illumination direction *i* and c_{gi_j} is the feature vector of the *jth* face image with an illumination direction *i* by projecting into the specific eigenspace and *ni* is the number of images in Gi.

(3) Compute the feature vectors of the probe face images: Given a probe face image (say *p*), we project it into every illumination eigenspace and can get *I* feature vectors ($c_{pi}=S_i^T \times p,(i=1,2,...,I)$). Then the eigenspace corresponding to *p* is determined by computing the ratio of Signal-to-Noise of the rebuilt image. For any eigenspace $S_i (i=1,2,...,I)$, the rebuilt image is denoted as $p_i=S_i \times c_{pi}$, then the ratio of Signal-to-Noise corresponding to that eigenspace is $R_i=10\lg(\frac{||p||^2}{||p-p_i||^2})$. If $R_j=\max\{R_i|i=1,2,...,I\}$, the eigenspace corresponding to *p* is S_j and the vector $c_{pj}=S_j^T \times p$ is the feature vector of *p*.

(4) Complete recognition: For any element in set C_{gj}, if $dis(c_{pj},c_{gj_k})=\min\{dis(c_{pj},c_{gj_i})\,|\,i=1,2,...,ni\}$ ($dis(x,y)$ stands for the distance between vector *x* and *y*), the probe face image *p* and gj_k in gallery are considered as imaging from the same object. The measure of distance in this paper is Euclidean distance.

4 The Back-Propagation Neural Network Based (BPNNB) Method

The architecture of BPNNB-method is shown in Figure 1. If the number of illumination directions of the face images is *I*, we can build *I* illumination eigenspaces. Extracting the feature vectors of each image in all *I* eigenspaces, we can get *I* eigenvectors. Then by combining all *I* eigenvectors one by one, we can get the combined eigenvector of the image which is used as the input of a BP-neural network with one hidden layer whose output points out the identity of the image.

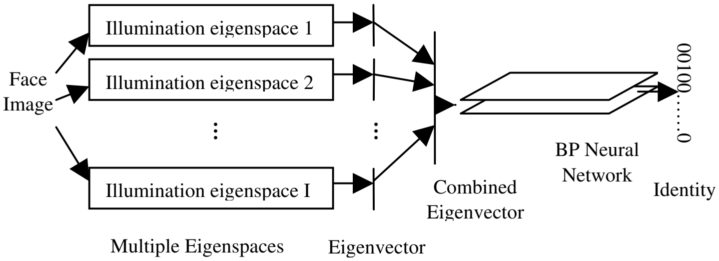


Fig. 1. Architecture Base on Back-Propagation Neural Network

At the test phase, given a probe image p, if it belongs to one of the object in the gallery, the output should tell its identity, otherwise the output is unrecognizable. So the output is a binary vector in which one bit stands for a specific object. If the architecture is designed to recognize k people, the output vector has k+1 bits in which the first k bits stand for k objects to recognize and the last bit stands for unrecognizable objects rejected by the architecture (also called negative samples). If the identity of the face image is j, the jth bit of the output vector is 1 and the other k bits are 0. And if the identity of the face image is unrecognizable, the (k+1)th bit is 1 and the other k bits are 0.

5 Experiment

While a number of benchmark face bases for face recognition, such as FERET database [3] and PIE database [6], are available, none of them are suitable for face recognition methods that are based on multiple illumination eigenspaces and neural networks, due to the need of multiple images of each object with each specific illumination direction. To evaluate our approaches, we have built the IAE (Illumination And Expression.) face base [5], a collection of 4320 face images of 27 subjects - 20 men and 7 women - among whom 15 persons wore glasses. There are four illumination directions to image the images. We call the front of a face 90' direction, the left of a face 0' direction. Accordingly, there are 30' and 60' directions in the left front of a face. For all the 27 subjects, there are 40 images for every subject in each illumination direction, so there are 4320 images in total. All images are color with 352*288 pixels in size. For the experiment of this paper, all color images were transformed into gray level images. Sample images in IAE face base are shown in Figure 2 (if the colors of the images were visible, you would tell that the illumination directions were obviously different). The rows correspond to the directions of 0', 30', 60', and 90', respectively.

The face images were normalized before they were presented to the system for training or testing. The normalized version of a face image satisfies such constraints that the face could be appropriately cropped. The constraints include that the size of the image was fixed, the line between the two eyes was parallel to the horizontal axis, and the inter-ocular distance (distance between the two eyes) was set to a fixed value. The normalized face images in this paper satisfy the constraints in Figure 3. Because our emphasis was placed upon face recognition with changeable illumination conditions, we conducted face detection by manually locating the eyes and then rotated and resized the row images using the bilinear method. After that, face images were cropped to satisfy the constraints in Figure 3. In the end, the appropriately cropped face images were histogram equalized. Hence all the training and testing sets in this paper were normalized and histogram equalized gray level face images.

The experiment was intended to recognize 10 subjects in the IAE face base and use another 10 subjects as a "rejection" subject, i.e., negative samples. Four eigenspaces, each corresponding to an illumination direction, were built. The training set for each eigenspace consisted of 270 face images among which each of the 27 persons contributed 10 images. To evaluate the NNEB-method [5], we trained 4 BP-neural networks with one hidden layer, each corresponding to an illumination direction. The number of input units of each neural network was equal to the number of the dimensions of the corresponding eigenspace. Each neural network had 11 output units among which the first 10 units were corresponding to 10 subjects to recognize and the last one was corresponding to the 10 subjects to reject, i.e., the negative samples. At the test phase, if a given face image was from the first 10 subjects, the system would tell the identity of the subject; if the image was from another 10 subjects to reject, the system would reject the image and mark it as "unrecognizable". The training set for each neural network was composed of 300 images with a corresponding illumination direction. Among the 300 images, each of the 10 persons to be recognized contributed 20 images, and each of the 10 persons to be rejected contributed 10 images. None of the 1200 (300*4) images have been used to build eigenspaces.

The training set (gallery images) for the architecture of neural network ensemble in NNEB-method, RDEB-method and BPNNB-method comprised of all the 1200 images used to train the first layer neural networks. The testing set (probe images) consisted of 800 images, 200 images from each illumination direction. Among each 200 images, each of the 20 subjects contributed 10 images. None of the 800 images was in the training set.

Taking different numbers (i.e. 10, 20, 30, 40 and 50) of dimensions for each eigenspace, we tested all the three method mentioned above (NNEB-method, RDEB-method and BPNNB-method) using the testing set with unknown illumination directions. For comparison, we also tested an assumptive method which knows the illumination directions of faces and completes recognition in the specific eigenspace using eigenface method [2] for each face subset with a specific illumination direction.

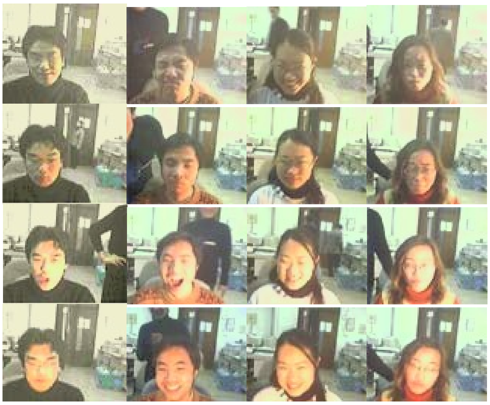


Fig. 2. Samples in IAE Face Base

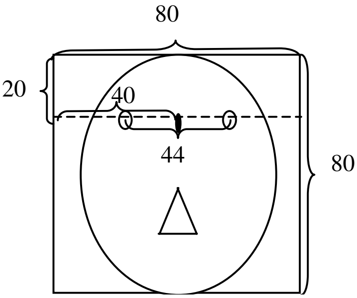


Fig. 3. Standard Face Image

Note that the parameters of neural networks have a great effect on the performance of neural networks and the number of units in the hidden layer is the most important one among all the parameters. So our experiment only took the number of units in the hidden layer into account. This number was adjusted several times and the best result was selected as the final evaluation of recognition performance.

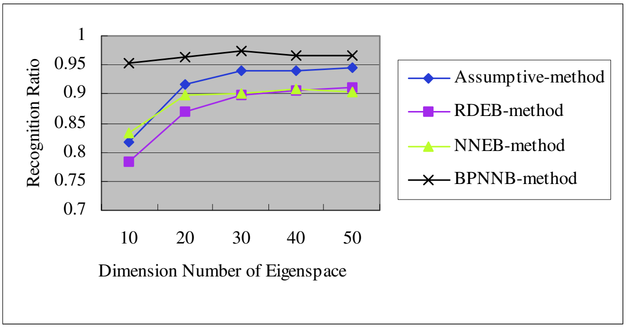


Fig. 4. Recognition Performance

The experimental result is shown in Figure 4, which shows that all the three methods based on multiple eigenspaces achieved high recognition performance. As the number of dimensions of eigenspace increases, the performance mostly increases. The performance of NNEB-method and BPNNB-method increases with a bigger acceleration when the number of dimension of eigenspace is under 20 compared to over 20. While the threshold of RDEB-method and the Assumptive-method is 30. So the methods using neural network can obtain features of more discriminative ability with a smaller dimension number. Moreover, BPNNB-method proposed in this paper achieved much higher performance than other methods, even compared to the Assumptive-method, even if the number of dimension of eigenspace is small (near 10).

6 Conclusions

We have presented two new face recognition methods based on multiple eigenspaces. They can complete recognition with high recognition accuracy. In the real world, however, illumination is complex. The illumination directions and the illumination intensity are essentially arbitrary. This is a major issue of future research. Moreover, a practical face recognition system should address both illumination and pose problems at the same time. A possible approach is to construct multiple eigenspaces from a large set of training images in a self-organizing way [7].

Acknowledgements

The National Natural Science Foundation of P.R. China under grant No. 60273033 and the Natural Science Foundation of Jiangsu Province of China under grant BK2003067 supported this research.

References

1. Zhao, W., Chellappa, R., Rosenfeld, A., and Phillips, P. J.: Face Recognition: a Literature Survey. CVL Technical Report, University of Maryland (2000)
2. Turk, M., and Pentland, A.: Eigenfaces for Recognition. Journal of cognitive neuroscience, **3** (1991) 71-86
3. Phillips, P.J., Wechsler, H., Huang, J., and Rauss, P.J.: The FERET Database and Evaluation Procedure for Face-recognition Algorithms. Image and Vision Computing, **16** (1998) 295-306
4. Pentland, A., Moghaddam, B., and Starner, T.: View-based and Modular Eigenspaces for Face Recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA (1994) 21-23
5. Li, W.J., Wang, C.J., Xu, D.X. and Chen, S.F.: Illumination Invariant Face Recognition Based on Neural Network Ensemble. In Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2004), IEEE computer society, Boca Raton, Florida (2004) 486-490
6. Terence, S., Simon, B., and Maan, B.: The CMU Pose, Illumination, and Expression (PIE) Database. In Proceedings of the Fifth International Conference on Face and Gesture Recognition (2002)
7. Leonardis, A., Bischof, H., and Maver, J.: Multiple Eigenspaces. Pattern Recognition, **35** (2002) 2613-2627

Boosted Independent Features for Face Expression Recognition

Lianghua He[1,2], Jianzhong Zhou[1], Die Hu[2], Cairong Zou[2], and Li Zhao[2]

[1] Southeast University, Research Center of Learning Science
Nanjingjiang, Jiangsu 210096, China
[2] Southeast University, Dept. of Radio Eigneering
Nanjing, Jiangsu 210096, China

Abstract. Independent Component Analysis (ICA) is used widely to extract statistical independent features for analysis and discrimination in recent years. But its random properties make it very difficult to test the efficiency and validation of the extracted independent features. In this paper, we propose a new method called BoostedICA to solve such problems by running ICA several times and boosting the selected independent components. Because of the local extremum question in calculating independent component, several times of running could get the more valid components with larger probability. The AdaBoost algorithm can guarantee the discriminating efficient of the selected features from the statistical theory. The proposed method achieves both computational efficiency and accuracy through optimizing extracting and choosing features. Finally we describe face expression recognition experiments on person-dependent and person-independent. The experimental results of 97.5% and 86% recognition rate respectively show that our method has better performance compared with other methods.

1 Introduction

Independent Component Analysis (ICA) (Comon, 1994; Jutten and Herault,1991; Hyvarinen et al., 2001b) is a signal processing technique whose goal is to express a set of random variables as linear combination of statistically independent component variable. Two promising applications of ICA are blind source separation and feature extraction. The use of ICA for feature extraction is motivated by results in neuroscience that suggest that the similar Principal of redundancy reduction [1]. In recent years, much work has been done on this about object detection, recognition and representation with good results [2,3,4,5,6]. But there are several major problems in application of ICA: one is that the reliability of the estimated independent components is not known. Another is the effectiveness of features extracted by ICA has not verified. For the last, an ICA algorithm gives a specified number of components, but it is not known which ones are to be taken seriously and which ones contain the useful discriminating information. To solve such problems, yoshinori Sakaguchi[7] presented the method of SICA by maximizing Mahalanobis distance between classes. Johan Himberg [8] proposed a method to estimated independent components in statis-

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 137–146, 2005.

tically reliable based on running the ICA many times with slightly different conditions and selecting the independent components which could be produced almost every run in the signal space and well separated form the rest of the estimates. Both of them got the excellent experiments results. But there are still some questions: First, according to [8] we need to calculate many times which could be very time-consuming, especially on the condition of large data sets. Second, it is impossible for us to get all the independent components by limited times running the algorithm. So the independent components used finally are reliable only in some degree, not absolutely. At last, both [7] and [8] separate the feature extraction from feature recognition. What they had done are all to extract reliable or efficient features without considering the processing of discriminating. So they could not guarantee the extracted features have strong ability of discrimination.

Based on the problems above, we proposed a novel method called BoostedICA considering both the features exacting and recognition ability by achieving strong independent features and boosting them. In this paper, every independent feature contains independent information and has completely different discrimination ability. In other words, they are all possible "weak classifiers" in different angle. Thus AdaBoost can find the right moderate weights automatically for us after we give it all the candidates. According to the theory of AdaBoost, we know that the new features distribution is proved reasonable in theoretical and useful in the real word. What's more, it is not necessary for us to calculate independent components too many times to assure their validity. We only need to count for few times and then boost them by Adaboost. So most of the problems in [7] and [8] could be avoided thought our BoostedICA. In order to test our proposed method, at last, we describe some facial expression experiments with many methods including BoostedICA. We can see from the results that our method has much superiority in recognition rate and generalization over others.

The rest of the paper is arranged as follows: In section 2, after describing Independent Component Analysis in facial expression recognition briefly, we mainly analyze the valid and efficient of the independent component. Later in section 3, we focus on the learning algorithm of Adaboost for face expression recognition in detail. Experimental results are provided in Section 4 before some useful conclusions are drawn in Section 5.

2 Feature Extraction with Independent Component Analysis

There are a number of algorithms for performing ICA [9,10,11]. We chose the fast fixed point algorithm proposed by A.Hyvarinen and E.Oja[11], which was derived from the principle of maximum non-Gaussian information such as kurtosis or negentropy etc. For the problem of facial expression recognition, the algorithm is motivated as follows: Supposed that there are n training facial expression images to be processed, that is , n vectors $x_1, x_2, ..., x_n$, which are assumed to be linear combination of m unknown statistical independent components $s_1, s_2, ..., s_m$. Then after we arrange the observations x_i into a matrix $X=(x_1,x_2,...,x_n)$ and the component variable s_i into a matrix $S=(s_1,s_2,...,s_m)$, the linear relationship between x_i and s_i is given as:

$$X=AS \tag{1}$$

A is the transformed mixing matrix. Because $s_1,s_2,...,s_m$ are independent each other, according to the Central Limit Theorem, $s_1,s_2,...,s_m$ also have a distribution of closing to gaussian. In this paper, we use negentropy as the measure of nongaussianity. Once we choose a non-quadratic function G which should not grow too fast, the maxima of the approximation of the negentropy of $s_1,s_2,...,s_m$ are obtained at points where $E\{Xg(W^TX)\}-\beta W=0$ with the constraint of $E\{(W^TX)^2\}=\|W\|^2=1$, where g is the derivative of the nonquadratic function G . W is the inverse of A . β is the Lagrange coefficient. Then Newton iteration method is used to solve this equation. Some of the independent components are shown in Fig 1.

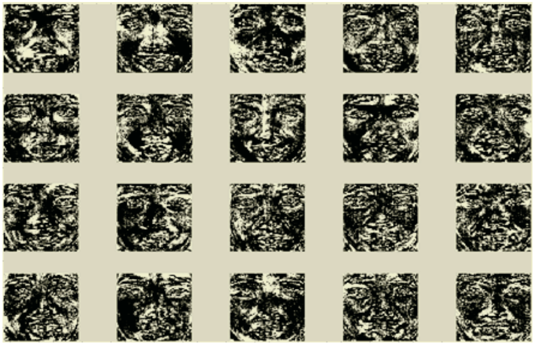


Fig. 1. Some independent components examples

But there are many ambiguities in Eq.(1): At first, the independent components are not exclusive and stable. If we run ICA several times, the results are different. What's more, this also makes the independent component order vary. We could get as many independent components as we want theoretically. Thus how many is enough and which one is the right that we need are two main problems. Second, the magnitude (energies) and the sign (direction) of the calculated independent components could not be determined. The reason is that both A and S are all unknown and the same output could be gotten by the mutual adjusting of the ambiguity and the sign between A and S .

In general, when extracting features for recognition, we should consider two factors: the information that the features contained and the general discriminating of these features. Although we do not know exactly what independent components could be got, Bartlett & Sejnowski [4] results indicates that higher-order features are useful for capturing invariant features of patterns as well as the second-order features. That is if we select the right independent components, it is possible for us to make good recognition. Then the question becomes how to select and combine these components?

In this paper, in order to get the needed independent components, we adopt the method of [8] to calculate ICA algorithm many times and save all the results because it is reasonable to run the estimation algorithm many times, using different initial values to assess the algorithmic reliability of the estimated components [8]. But differently, we do not need to run the algorithm so many times as [8] since we need not cluster these independent components to small classes and discard those look like less reliable. So, on the one hand we can save much calculating time than [8]. On the other hand, the probability of finding strong discriminating independent components is larger than [8] because the discarded components by [8] may have strong discrimination. Thus we judge the validation of those independent components from both statistical and the experiments.

Once we have gotten all these independent features, we could not use them all with equal weight for they are independent each other and express the different information of the original data. For example, some component may show the information of nose, some maybe for eyes. Obviously the eyes information is more important than the nose information in facial expression recognition. The key is we do not know which independent component is more important because of its random properties. Fortunately, AdaBoost algorithm can deal such questions perfectly. In the next section, why and how use it are described in detail.

3 Learning with AdaBoost

Many theories and algorithms have been developed for learning a general classifier. For example, Neural Networks can learn a classification function from training data and applied to solving many visual recognition/detection problems. However, it is nontrivial to decide the number of the neurons and the type of the nonlinear function. Support Vector Machines (SVM) are developed to maximize the margin of the labeled data, But choosing proper kernel functions for a specific real problem remains challenging. Moreover, the number of the support vectors that compose the decision function would dramatically increase when the decision manifold becomes complicated.

Recently, Boosting is a method of finding a highly accurate hypothesis (classification rule) by combining many "weak" hypotheses, each of which is only moderately accurate [12]. It is extended by Yoav Freund and Robert E. Schapire to AdaBoost [13] with updating the weights dynamically according to the errors in previous learning. AdaBoost is a kind of large margin classifiers and is very simple and easy to implement. It has been proven that boosting minimizes an exponential function of the margin over the training set [14]. So in this paper we use AdaBoost to select the right weight for each feature from the whole independent components.

AdaBoost proceeds in round. On each round a weak classifier is chosen. The only requirement of the weak classifier is that it has an error rate less than 0.5. Supposed that we have N independent components $S=(s_1,s_2,...,s_N)$, and for the ith training image, we got N projection coefficients $\boldsymbol{P}_i=(p_{i1},p_{i2},...,p_{iN})$ correspondingly. Our input vector is the independent component $\boldsymbol{P}_i$ and the label y_j. In this paper, each independent component is a candidate of weak features. We first make a simple judgment to find the real one according to their Euclidean-measure performance just as:

For Q training images with K classes, there are Q projection coefficient vectors for jth independent component $p_{k(i),j}$ $k=1,...,K;i=1,...,Q_k$ with the class mean value of $u_j(k)$ $k=1,...,K;$, where Q_k is the samples number for kth class and $\sum Q_k = Q$. If

$$\sum_{k(i)} f(p_{k(i),j}) > \lceil Q_k/2 \rceil \tag{2}$$

where $f(p_{k(i),j}) = \begin{cases} 1 & if\ p_{k(i),j} > u_j(k) \\ 0 & otherwise \end{cases}$ and $\lceil \cdot \rceil$ is for integer ceiling, then s_j is the valid component for class k . That is, s_j can discriminate class k better than random. If s_j shows good performance for several classes, we could believe it has much critical discriminating information and should be paid more attention.

After M independent component coefficients were selected as weak features and in order to use the continuous multi-class AdaBoost algorithm, the weak learners should be configured with a feature f , a threshold θ_k :

$$h(f,k) = \begin{cases} 1 & if\ f > \theta_k \\ 0 & otherwise \end{cases} \tag{3}$$

Denote the sample space by Φ and the label set by Ψ . A sample of a multi-class multi-label problem is a pair (X,Y), where $X \in \Phi, Y \in \Psi$. For $Y \subseteq \Psi$, define $Y[l]$ for $l \in \Psi$ as

$$Y[l] = \begin{cases} 1 & if\ l \in Y \\ -1 & if\ l \notin Y \end{cases}$$

AdaBoost Algorithm:

Input: 1) Given training examples $(\mathbf{x}_{k(i)}, y_k)$ $k=1,...,K;i=1,...,Q_k$ with the label $y_k = 1,2,...,K$, K is the number of classes. 2) the number of iteration T

Initialize: weight $D_1(i,k) = \frac{1}{QK}, i=1,...,Q; k=1,...,K$

Do for t = 1,...,T:

1. Under the distribution D_t , select a weak classifier h_t: $\Phi \times \Psi \rightarrow [-1,1]$ from the weak classifier pool to maximize the absolute value of $\varepsilon_i = \sum_{i,k} D_t(i,k) Y_i(k)[h_t(X_i,k) - y_k]$
2. Choose $h_t(*) = h_k(*)$ such that $\forall j \neq k, \varepsilon_k < \varepsilon_j$,let $\varepsilon_k = \varepsilon_j$
3. Let $a_t = \log(\frac{1-\varepsilon_t}{\varepsilon_t})$

4. Update: $D_{t+1,i}=D_{t,i}\beta_t^{e_i}$,where $e_i=1\,or\,0$ for example x_i classified correctly or incorrectly respectively with $\beta_t=\dfrac{\varepsilon_t}{1-\varepsilon_t}$.
5. Normalize the weights so that they are a distribution $D_{t+1,i}=\dfrac{D_{t+1,i}}{\sum_{j=1}^{n}D_{t+1,j}}$.

Output the final hypothesis: $H(X,l)=sign(\sum_t a_t h_t(X_t,l))$

To get a multi-class single-label classifier, the final output should be adjusted to the following form: $H(X)=\arg\max_t(\sum_t a_t h_t(X,l))$ and furthermore a confidence of H is defined as: $Conf_H(X)=\dfrac{\sum_t a_t h_t(x,H(x))}{\sum_t a_t}$.

There are several novel features in our proposed method: At first, it is first time to combine ICA with Boost to make facial expression recognition. In fact, ICA is an excellent method to extract features and Boost has perfect performance on making good use of the extracted features. Thus the new method has much superiority on these two aspects. Second, ICA is run several times though which we can get robust and valid independent components with large probability. This is vital for these components are the basic and only basement of later recognition. Third, unlike the traditional boosting method that uses the original feature vector directly, our algorithm boosting in the ICA subspace. The dimension of feature space is first greatly reduced without loss on discriminative information. After ICA the features are independent and better accuracy can be achieved

4 Experiments and Results

In this paper, the expression recognition is defined as a seven-class problem of anger, disgust, fear, happiness, neutral, sadness and surprise. The JAFFE database [15] is used to train the expression classifier. The database is a set of 213 images of 7 different expressions posed by 10 Japanese females with about 3 images for each expression. Considering the containing information and size of the processing data, all images are normalized to 100×100-pixel patches with two eyes and mouth at the same position. See Fig.2 for some samples.

There are two kinds of experiments made in this paper: person-dependent and person-independent. The former experiments put more emphases on discriminating the difference on expression itself since the difference of face or others is not much for the same person at the similar condition. On the contrary, the later experiment is designed to test the generalization ability on the standard database and non-standard database. All face images are frontal and normalized.

For person-dependent experiments, the training probe and testing gallery have the same person with different images. In this case, we use the image-pairs recognition

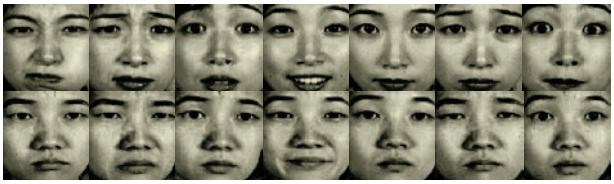


Fig. 2. Normalize expression samples. From left to right, the expressions are anger, disgust, fear, happiness, neutral, sadness and surprise.

method which used widely in face recognition. As mentioned before, there are almost 3 images I_1, I_2, I_3 for one expression one person. Thus we can have 3 different kinds of image pairs $(I_1, I_2), (I_1, I_3), (I_2, I_3)$. In total there are about 200 pairs. One third of them are used to train, the others are constructed as the test set. In order to make good use of the available data as each pattern used is used both as training and testing data. The strategy of leave-one-out cross-validation is used. That is, there are nearly 70 positive pairs and 2415 (70*69/2) negative pairs to be trained for every time. 560 independent components are obtained after running the algorithm of ICA 4 times. 367 independent components are valid according to the formulation (2). One thing should be noted that in this case the hypothesis function (3) will be changed to:

$$h(p_{I_1,j}, p_{I_2,j}) = \begin{cases} 1, & if \ \ abs(p_{I_1,j} - p_{I_2,j}) > t_j(k), \\ & \quad j = 1,\ldots,N; k = 1,\ldots,K \\ 0, & otherwise \end{cases}$$

for feature $p_{*,j}$, where (I_1,I_2) is the image pair. We run AdaBoost for 367 rounds to get a final classifier with such 367 independent features.

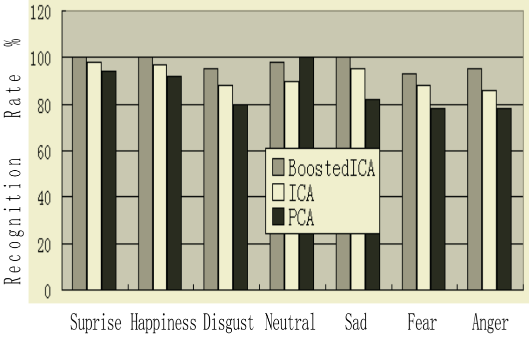


Fig. 3. Recognition rate on JAFFE images with false alarm of 0.1%

Three kinds of methods are applied: BoostICA, ICA with the same independent components as BoostICA and PCA with 99% energies reserved. All the experimental results are shown in Fig. 3 with 0.1% false alarm rate. We can see that ICA and BoostedICA perform better than PCA at almost all expressions except neutral. The reason we think is that the principal components contain the main information especially on the shape of face which match the information contained by neutral expression tightly. This shows that independent components have more discrimination ability on facial expression than principal components. The other exciting result is BoostedICA method do has higher recognition rate that ICA, which means that the new weight distribution between these independent components is critical. The similar experiment was done by [19] with GaborWavelet featrues. Compared with it, our 97.5% average recognition rate has much superiority to its 88.8% performance.

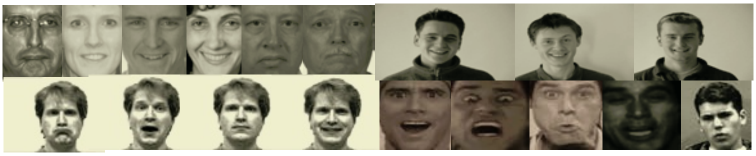


Fig. 4. Selected expression test set from all kinds of face database and videos.

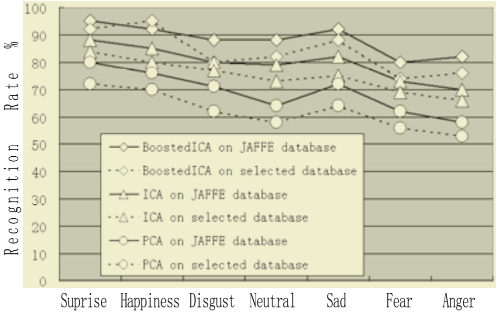


Fig. 5. Results of person-independent experiments on JAFFE database and selected database with false alarm of 0.1%.

In the experiments of person-independent, two test sets are used: standard data base and selected database. The standard database is composed by part of the standard JAFFE database. The selected database is constructed by images from the face database of FERET, ESSEX, CVL which are not for facial expression recognition or selected from videos. Some examples are shown in Fig 4. When testing on standard database, we use the leave-one-group-out cross-validation method to use the database adequately. That is, we separate the whole database according to people it contains. Only one group was used to test and the left nine groups were trained. 453 valid components in the 715 total independent components are used to boost after 4 times running the ICA algorithm. When testing on the selected set, all the images in JAFFE database are trained and 612 valid components are boosted. The results are shown in Figure 5.

We know from Fig. 5 that: At first, comparing to person-dependent, the recognition rates of three methods all decrease on the two testing sets when person-independent, especially on the selected database. This indicates that the difference beyond expression itself, such as the shape of face, the context in the image, the race of people, has much affectation on facial expression recognition. Second, the PCA feature behaves the worst of three. The reason we think is because the features are extracted by PCA mainly according to reconstruction error which means they should contain the information of original database as much as possible. Thus the generalization is poor. While for ICA, with containing dependent information in every component, it is certain that some components do only have the information of expression, such as eye blinking, mouth closing or browse frowning. So the feature extracted by these independent components must have strong generalization ability. Third, there is more distinction between BoostedICA and ICA than that in person-dependent experiments. The reason we think is boosting. As we deduced in the second, there are some expression components with strong generalization ability. At the same time, there are still have other components called sub-expression components which do not have such ability. Through rearranging the weight of these independent components by AdaBoost, sub-expression components have larger weight than expression components. Thus the final combination (BoostedICA) has stronger than the original combination (ICA).

Many methods have tested on person-independent facial expression recognition with JAFFE database. Here are some results shown in the Table 1

Table 1. Some results tested on JAFFE database

paper	Method	Recognition rate
[17]	HLAC + Fisher weight maps	69%
[18]	LNNF	50%～70%
[18]	LNNF with difference images	70%～80%
[16]	Boosted Har features	92.4%
[19]	Gabor wavelet	82%
	BoostedICA	86%

Our method has improved the recognition rate much compared with [17, 18]. Although only slight better than [19], our processed data is much less than it with Gabor coefficient so our method has much more efficiency. Unlike our experiments of person-independent, the testing data used in [16] are all from the JAFFE database by mirroring, rotating etc. which can explain why it has much higher recognition rate than us.

5 Conclusion

AdaBoost is an effective technique to enforce weak classifiers. ICA can extract features independent each other. But their validity and efficiency could not be known because of their random. So in this paper, we proposed a new method called BoostedICA by running ICA several times and boosting the selected components to improve them. The two kinds of complementary ways are integrated. Our approach not only effectively stabilizes and generalizes the classifier but also makes use of all the discriminative information. At last, a series of person-dependent and person-independent experiments were tested on JAFFE database. The results indicate that the features extracted by our method have more discriminating ability than other features, such as ICA, PCA, Gabor and NMF.

References

1. Hyvarinen A.: Fast and Robust Fixed-Point Algorithms for Independent Component Analysis. IEEE Trans. On Neural Networks, **10** (1999) 626-634
2. Tae-Kyun Kim, Hyunwoo Kim, Wonjun Hwang, Seok-Cheol Kee, Josef Kittler: Independent Component Analysis in a Facial Local Residue Space. CVPR'03
3. Chengjun Liu, Harry Wechsler. Independent Component Analysis of Gabor Features. IEEE Transactions on Neural Networks, **14** (2003) 919-928
4. Bartlett, M.S., Movellan, J.R., jnowski,T.J. Se.: Face recognition by ICA. IEEE Trans. On Neural Networks, **13** (2002)
5. Guan, A. X. , Szu, H. H.: Local Face Statistics Recognition Methodology beyond ICA and/or PCA. International Joint Conference on Neural Networks, Washington DC, **2** (1999) 1016-1021
6. Kwak, N., Choi, C. , Ahuja, N.: Face Recognition Using Feature Extraction Based on Independent Component Analysis. International Conference On Image Processing, Rochester, New York, Sept (2002)
7. Yoshinori Sakaguchi, Seiichi Ozawa, Manabu Kotani: Feature Extraction Using Supervised Independent Component Analysis by Maximizing Class Distance. ICONIP2002-Singapore, **5** (2002) 2502-2506
8. Himberg, J., Hyvärinen, A., F. Esposito: Validating the Independent Components of Neuroimaging Time-Series via Clustering and Visualization. Neuroimage, **22** (2004) 1214-1222
9. Bell, A. J. , Sejnowski, T. J.: An Information-Maximization Approach to Blind Separation and Blind Deconvolution. Neural Comput., **7** (1995) 1129-1159
10. Cichocki, A., Unbehauen, R., Rummert, E.: Robust Learning Algorithm For Blind Separation Of Signals. Electron. Lett., Vol. 30, No. 7, Pp. 1386–1387, 1994.
11. Hyvarinen, A. , Oja, E.: A Fast Fixed-Point Algorithm For Independent Component Analysis. Neural Computation, **9** (1997) 1483-1492
12. Schapire, R.E. , Singer, Y.: Improved Boosting Algorithms Using Confidence-Rated Predictions. In Colt'98 (1998) 80-91
13. 13.Yoav Freund and Robert E: Schapire. A Decision-Theoretic Generalization of Online Learning and an Application to Boosting. Journal of Computer and System Science, **55** (1997) 119-139
14. Friedman, J., Hastie, T., Tibshirani, R.: Additive Logistic Regression: A Statistical View of Boosting. The Annual of Statistics, **28** (2000) 337-374
15. Michael J. Lyons, Shigeru Akamatsu, Miyuki Kamachi , Jiro Gyoba.: Coding Facial Expressions with Gabor Wavelets. Proceedings, Third Ieee International Conference on Automatic Face And Gesture Recognition, Nara Japan, IEEE Computer Society (1998) 200-205
16. Yubo Wang, Haizhou Ai, Bo Wu, Chang Huang: Real Time Facial Expression Recognition With Adaboost. Icpr'04
17. Shinohara, Y., Otsu, N.: Facial Expression Recognition Using Fisher Weight Maps Automatic Face And Gesture Recognition, 2004. Proceedings. Sixth Ieee International Conference on , **17** (2004) 499 - 504
18. Buciu, I., Pitas, I. :Application of Non-Negative and Local Non Negative Matrix Factorization to Facial Expression Recognition. **1** (2004) 288 - 291
19. Michael J. Lyons, Julien Budynek, Shigeru Akamatsu : Automatic Classification of Single Facial Images. Ieee Transactions On Pattern Analysis And Machine Intelligence, **1** (1999) 1357-1362

Intelligent Immigration Control System by Using Passport Recognition and Face Verification

Kwangbaek Kim

Dept. of Computer Engineering, Silla University, Korea
gbkim@silla.ac.kr

Abstract. This paper proposed the intelligent immigration control system that authorizes the traveler immigration and detects the forged passports by using automatic recognition of passport codes and the picture and face verification. The proposed system extracts and deskewes the areas of passport codes from the passport image. This paper proposed a novel ART algorithm creating the adaptive clusters to the variations of input patterns and it was applied to the extracted code areas for the code recognition. After compensating heuristically the recognition result, the detection of forged passport is achieved by using the picture and face verification between the picture extracted from passport image and the picture retrieved from the database based on the recognized codes. Due to the proposed ART algorithm and the heuristic refinement, the proposed system shows the relatively better performance.

1 Introduction

Due to the globalization and the advance of travel vehicles, the number of passengers of overseas travel is gradually increasing. The current immigration control system carries out manually the passport inspection and requires a long time for immigration, putting passengers to inconvenience. And the automatic passport inspection requires the precise processing so as to execute the critical functions such as the discrimination of passport forgery, the search for a wanted criminal or a person disqualified for immigration, etc. [1]. So, for the automatic immigration management, the various automatic systems were proposed that recognize the passports using the refined neural network and the RBF network, etc. [1],[2].

This paper proposed a novel intelligent passport recognition system that is able to discriminate a passport forgery by using the automatic recognition of passport codes and the picture and face verification. The proposed system, first, extracts and deskews the areas of passport codes from the passport image by applying the edge detection algorithm and the horizontal minimum value filter to the image. And for the recognition of passport codes, we proposed the new ART algorithm creating the adaptive clusters to the variations of input patterns. So, the proposed system applies the new algorithm to the extracted code areas and compensates heuristically the recognition result to get the precise passport code information. And, using the code information recognized, the picture and related information of the passport owner are retrieved from the immigration management database. The proposed system measured the picture similarity for the face verification by using the correlation that considers totally

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 147–156, 2005.

the overall intensity, edge values and the color information, etc., and performed the verification separately in the overall picture area and in the facial area to get the higher reliability in the discrimination of passport forgery.

2 Extraction and Deskewing of Passport Codes

The passport image consists of the three areas, the picture area in the top-left part, the text information area including passport type, nationality, passport number, name, sex and birthday etc., in the top-right part, and the passport code area in the bottom part.

2.1 Extraction of Code Areas

The passport code area has the following characteristics: the passport codes consist of alphabetic and numeric characters and arrange in a row horizontally, and the code characters consist of strokes with 1 or 2 pixels, making the clear distinction between the pixels of code areas and background pixels. The above-stated characteristics are able to be embossed by applying the edge detection algorithm such as the Sobel operation to the passport image. Because the interval between code characters is in the regular range, the code characters are able to be connected by using the horizontal n*1 minimum-value filter[3], called as smearing algorithm, that replaces n's pixels being adjacent horizontally with the minimum pixel value among them. In the filter, the value for n has to be set to the maximum size among the intervals between code characters.

After applying the edge detection algorithm and the horizontal minimum-value filter to the passport image, the output image is binarized by using the iterative thresholding method. The connected objects are extracted from the binarized image by the connected component labeling method[4]. By eliminating the useless objects based on the characteristics that the code area is arranged in a row in the bottom of passport image and is separated from the boundaries of the image, the passport code areas are extracted from the passport image.

The iterative thresholding method used in this paper classifies the background and the feature pixels from input image by using the stabilized threshold, which is obtained by performing the iterative operation that divides pixels to two clusters using the given threshold and updates the threshold using mean values of new clusters until the threshold is stable. The overall process of the code area extraction is the same as Fig. 1.

2.2 Deskewing and Refinement of Passport Image

The passport image may be scanned at the skew direction and the deskewing of passport image is required essentially for the post-processing such as the code extraction and recognition and the face verification.

This paper selected the code area with the longest width among the extracted code areas and calculated the angle between the line connected between centers of the left and the right sides of the code area and the horizontal line in the image. And, the passport image is deskewed by rotating pixels as much as the angle. This paper refined the coarse strokes of code characters by using the 3×3 median filter so that preventing the decline of recognition rate on the low definition images.

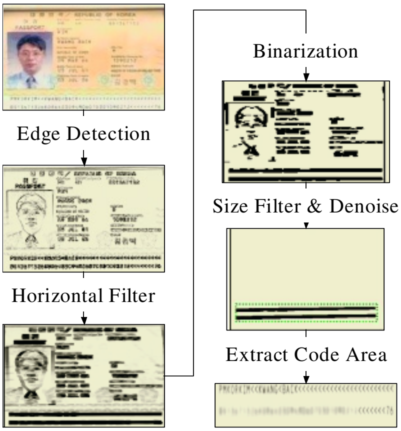


Fig. 1. Overall extraction process of passport code areas

3 Recognition of Passport Codes

The passport codes have totally 37's patterns that consist of 10's numeric characters('0'~'9'), 26's upper-case alphabetic characters('A'~'Z') and 1's special character('<'). This paper assumed that the pattern of the number '0' is the same as the character 'O' and fixed the 36's patterns to the patterns being recognized. Also, the ICAO 9303 OCR-B type font is used for the passport code.

To improve the recognition rate of passport codes with regular appearance, this paper proposed the new ART algorithm that creates the adaptive clusters to the variations of input patterns. The proposed ART learning algorithm is as follows:

Step 1. Define x_k as the *k*-th input vector, w_i as the connection weight vector of *i*-th cluster in the neural network.

Step 2. Perform the Exclusive-NOR operation on the newly given x_k and w_i, and select a winner node with maximum value.

$$\mu_{i^*} = \max\left(\frac{\left\| x_k \bullet w_i \right\|}{N}\right) \tag{1}$$

where *N* is the number of input nodes.

Step 3. Perform the similarity test for input vector. If μ_{i^*} is greater than the vigilance parameter ρ, the input pattern is determined to be similar with the winner cluster and the input vector makes to be included in the winner cluster. To reflect it, update the center vector of the winner cluster using Yager's intersection operator.

$$w_{i*}^{t+1} = 1 - \wedge\left(1, \sqrt{(1-x_k)^2 + (1-w_{i*})^2}\right) \tag{2}$$

If μ_{i*} is lower than ρ , the input pattern is determined to be not similar with the winner cluster and the new cluster including the input vector is created.

Step 4. Repeat from Step 1 to Step 3 until the learning of all input pattern is complete.

4 Picture Verification

This paper proposed the whole picture verification method for the discrimination of passport forgery, which extracts a picture area from the passport image and verifies it by measuring the similarity between the extracted picture and the picture image retrieved from the passport database.

4.1 Extraction of Picture Area

This paper extracted the picture area from the passport image using the information on the picture image retrieved from the database. This method, first, using the relative position information, estimates the search area including the picture area and secondly, compares the candidate areas in the search area with the retrieved picture image, determining the candidate area with the highest similarity as the picture area. In the search process of the picture area, the size of candidate area has to be set to the size of the picture image retrieved from the database.

4.2 Similarity Measurement

This paper uses the correlation[5] as the similarity measurement between the picture area extracted from the passport image and the picture image retrieved from the passport database.

The correlation of two-dimensional vectors is calculated by the correlation coefficient showed in Eq.(3). If the correlation coefficient goes close to 1, it means that the two vectors have the positive correlation, and if the correlation coefficient goes close to -1, it means that the two vectors have the negative correlation. If two vectors are irrelative mutually, the correlation coefficient will go close to 0.

$$r(T,R) = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(T_{ij} - \mu_T)(R_{ij} - \mu_R)}{M \times N \times \sigma_T \times \sigma_R} \tag{3}$$

where, *T* and *R* are the two-dimensional vectors indicating picture images being compared, *M* and *N* are the vector sizes of two images T and R, μ_T and μ_R are the means of *T* and *R*, and σ_T and σ_R are the standard deviations of *T* and *R*.

Fig. 2 shows the overall process of the feature extraction of picture images and the similarity measurement.

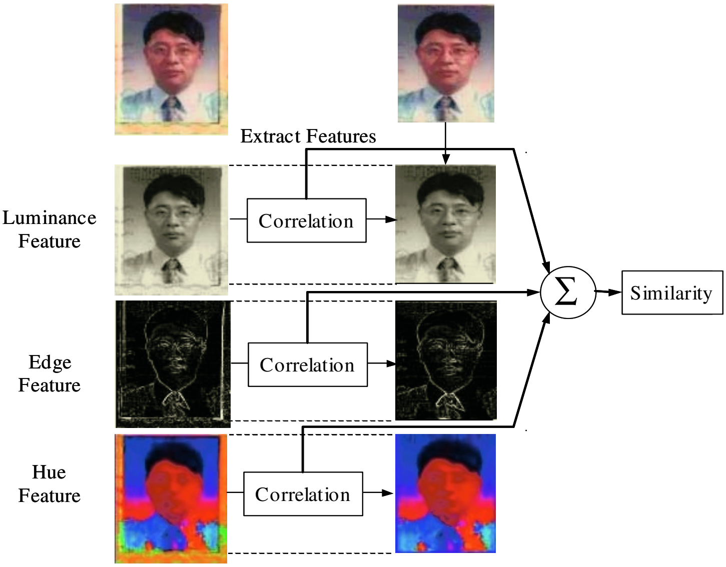


Fig. 2. Feature extraction and similarity measurement of picture images

4.3 Picture Verification of Passport Image

The features used for image matching are the color distribution, the morphological information and the frequency information, etc. [6][7]. For the picture verification, this paper used the various features such as the luminance, the edge vale, the HSV color distribution, the RGB color distribution, the normalized RGB color distribution, the YIQ color distribution and the YCbCr color distribution.

The discrimination algorithm of passport forgery using the picture verification is as follows:

Step 1. Measure every feature value from the extracted picture area and the retrieved picture image.

Step 2. Calculate the correlations between feature values of the same type.

Step 3. Calculate the weighted sum of the correlations.

Step 4. Compare the measurement with the given threshold. If the measurement is lower than the threshold, it is discriminated that the passport is possible to be forged.

This paper, for the improvement of processing performance, applied the combination of a few features having the superior discrimination to the correlation measurement.

5 Face Verification

This paper proposed the face verification of passport image for the more precise and reliable discrimination of passport forgery [8].

5.1 Construction of Face Template Database

The current passport database does not provide the additive information on the face of passport owner and the construction of face template database must precede the face verification. The construction process of the face template database is as follows: First, the face templates are constructed by extracting facial areas from multiple passport pictures and averaging them. Secondly, comparing candidate areas of the passport picture image with the face template, the area with the highest similarity is determined as the face template image of the passport image. And the template image is saved to the passport database as one field of passport information. Therefore the passport database is able to provide the whole picture image and the face template image for the face verification. The construction process of face template database is the same as Fig. 3.

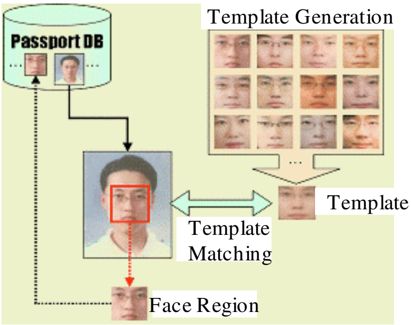


Fig. 3. Construction of Face Template Image Database

5.2 Face Verification

The face verification consists of the extraction of an area with the skin color and the matching of the extracted area and the face template. First, the extraction of a skin-color area from the passport image is achieved by using HSV color model. The features of skin-color areas such as the distributions of color, saturation and intensity are to be calculated heuristically in advance. For the removal of image noises, the passport image is reduced to the 1/10 size of original image and again, the output image is enlarged to the size of 10 times to get an image with the original size. And, the binarization of the output image assists to extract the area with skin color from the original image. The extraction of a skin-color area reduces the search area and improves the processing speed. If the extraction were failed, the whole passport image is set to the search area. This paper measured and evaluated various features from passport images used in the experiment. And, for the improvement of processing speed and reliability, this paper applied Luminance, Edge value, RGB color distribution to the correlation measurement for the picture verification and Hue, I-value of TIQ color model, Cb-value of YCbCr color model for the face verification.

6 Experiments and Performance Evaluation

This paper, for performance evaluation, implemented the proposed system in Visual C++ 6.0 and experimented on the IBM Compatible PC with Intel Pentium-IV 2.8GHz CPU and 512MB RAM. In the experiment for passport code recognition, 888's code characters with various sizes were extracted from 16's passport images originally scanned and the normalized vectors to the size of 20x20 were applied to the learning algorithm so that the 344's clusters were created in the proposed ART algorithm. And, the passport images used in the experiment were scanned at 300 DPI by HP ScanJet 6350C scanner. For the experiment of the picture and face verification, we created the 16's forged passport images in which the whole picture was replaced with another picture and the other 16's forged ones with synthesized facial areas. For the evaluation of flexibility to the various operation environments, we used 336's varied passport images that were created by applying the image operations such as noise insertion, contrast change, intensity change and saturation change to 48's original and forged passport images.

6.1 Experiment of Passport Code Recognition

Table 1 shows the recognition rate of passport codes on the 336's passport images. In the recognition experiment, the recognition of code area was regarded as a failure if all characters in the code area were not recognized. As showed in Table 1, the recognition experiment on the original passport images without any refinements showed the low recognition rate of about 23.1%. The result was incurred by the skew scanning of passport images and many errors occurred at the separation of characters form code areas. When the image deskewing and refinement operations were applied to the passport images, the better and stable recognition rate of about 98.6 % was showed.

Table 1. Performance Result of Passport Code Recognition

Image Variation	Original Images			Refined Images		
	# of Success	# of Error	Recogn. Rate(%)	# of Success	# of Error	Recogn. Rate(%)
Original Image	9	39	18.8	46	2	96.0
Noise Insertion	6	42	12.5	47	1	97.9
Contrast Increase	12	36	25.0	48	0	100
Contrast Decrease	12	36	25.0	47	1	97.9
Intensity Increase	6	42	12.5	48	0	100
Intensity Decrease	9	39	18.8	47	1	97.9
Saturation Decrease	9	39	18.5	48	0	100
Total	63	273	23.1	331	5	98.6

6.2 Similarity Measurement of Passport Pictures

For the correlation measurement in the picture and face verification, the paper measured from the passport images the various features as follows: luminance, edge value, HSV color distribution, RGB color distribution, Normalized RGB color distribution, I-value and Q-value of YIQ color model and Cb-value and Cr-value of YCbCr color distribution. And in this paper, the features were declared as Luminance, Edge, Hue, RGB, RGB-N, YIQ_I, YIQ_Q, YCbCr-Cb and YCbCr-Cr, respectively. The experiment for the picture and face verification measured and compared the correlation between the 16's passport picture saved in the database and the 112's pictures extracted from forged and varied passport images.

Table 2. Feature evaluation for the picture verification

Feature Type	Corr. with original pictures	Corr. with forged pictures	Diff.	Rank
Luminance*	0.980	0.390	0.590	3
Edge*	0.798	0.153	0.646	1
Hue	0.933	0.374	0.558	5
RGB*	0.980	0.390	0.590	2
RGB-N	0.893	0.336	0.557	6
YIQ-I	0.959	0.455	0.504	7
YIQ-Q	0.722	0.252	0.470	9
YCbCr-Cb	0.966	0.385	0.581	4
YCbCr-Cr	0.934	0.458	0.476	8
Average of selected features(*)	0.919	0.311	0.608	

Table 3. Feature evaluation for the face verification

Feature Type	Corr. with original pictures	Corr. With forged pictures	Diff.	Rank
Luminance	0.974	0.673	0.301	9
Edge	0.939	0.615	0.324	7
Hue*	0.801	0.309	0.493	2
RGB	0.972	0.659	0.313	8
RGB-N	0.923	0.529	0.394	6
YIQ-I*	0.950	0.518	0.432	3
YIQ-Q	0.855	0.457	0.398	5
YcbCr-Cb*	0.929	0.406	0.524	1
YcbCr-Cr	0.946	0.546	0.399	4
Average of selected features(*)	0.893	0.411	0.483	

Analyzing the results showed in Table 2 and 3, for the improvement of processing speed and reliability, this paper applied the selective features to the correlation measurement as follows: Luminance, Edge and RGB for the picture verification and Hue, YIQ-I, YCbCr-Cb for the face verification.

In the verification experiment, the correlation threshold values being each different were used for the original passport images and the forged ones by applying the different weights to feature values. In the case that the correlation threshold values were given to 0.8 equally, it was showed the relatively good performance with FAR(False Accept Rate) of 3.1% and FRR(False Reject Rate) of 2.7%.

Table 4. Performance of the picture and face verification in terms of threshold values

Threshold for picture verification	0.92	0.86	0.80	0.74	0.68	0.62
Threshold for face verification	0.89	0.84	0.80	0.75	0.70	0.65
Normal-Pass(%)	90.2	93.8	97.3	98.2	98.2	98.2
Normal-Alarm(%)	9.8	6.3	2.7	1.8	1.8	1.8
Forgery-Pass(%)	1.3	2.2	3.1	4.9	3.1	4.9
Forgery-Alarm(%)	98.7	97.8	96.9	95.1	96.9	95.1

7 Conclusions

This paper proposed the intelligent immigration control system that authorizes the traveler immigration and detects the forged passports by using the automatic recognition of passport codes and the picture and face verification. The proposed system extracts and deskewes the code areas from passport images by using the properties of passport code areas. And this paper proposed a novel ART algorithm and applied to the extracted code areas for the code recognition. After compensating heuristically the recognition result, the discrimination of forged passports is achieved by using the picture and face verifications between the picture extracted from passport image and the picture retrieved from the database based on the recognized codes. In the picture and face verification, the correlation was used as the verification measurement and it was measured using Luminance, Edge value, RGB color distribution for the picture verification and using Hue, YIQ-I value, YCbCr-Cb value for the face verification. When the correlation threshold of 0.8 was given, the verification experiment on the forged and varied passport images showed the relatively good performance with FAR(False Accept Rate) of 3.1% and FRR(False Reject Rate) of 2.7%.

Finally, as part of our future work, we plan to implement the recognition and complementary usage of the text information provided on the right-upper of passport image. Also for improvement of flexibility to site environment, we plan to implement the face verification on the face image entered directly through the CCD camera at the immigration gate.

References

1. Kim, K.B.: Passport Recognition using Fuzzy Binarization and Enhanced Fuzzy RBF Network. Journal of Fuzzy Logic and Intelligent Systems. **14** (2004) 22-227

2. Kim, K.B., Kim, Y.J., and Oh, A.S.: An Intelligent System for Passport Recognition using Enhanced RBF Network. Lecture Notes in Computer Science. LNCS **3314** (2004) 762-767
3. Kim, K. B. and Kim, Y. J.: Recognition of English Calling Cards by Using Enhanced Fuzzy Radial Basis Function Neural Networks. IEICE Trans. Fundamentals, **87** (2004)1355-1362
4. Kim, D. H., Kang, D. K., and Cha, E. Y.: Adaptive Image Labeling Algorithm using Non-recursive Flood-Fill Algorithm. The KIPS Transactions. **9** (2002) 337-342
5. Romdhani, S.: Face Recognition using Principal Component Analysis. The MS Thesis. University of Glasgow(UK) (1997) 1-56
6. Gonzalez, R. C.: Digital Image Processing. 2nd Edition. Prentice Hall (2002)
7. Maio, D. and Maltoni, D.: Real-time face location on gray-scale static images. Pattern Recognition. Vol.33. Issue 9. September (2000) 1525-1539
8. Li, Y., Qi, X. and Wang, Y.J.: Eye detection by using fuzzy template matching and feature-parameter-based judgment. Pattern Recognition Letters, **22** (2001) 1111-1124

Recognition of Finger Spelling of American Sign Language with Artificial Neural Network Using Position/Orientation Sensors and Data Glove

Cemil Oz and Ming C. Leu

Department of Mechanical and Aerospace Engineering
University of Missouri-Rolla, Rolla, MO 65409,USA
{ozc,mleu}@umr.edu

Abstract. An American Sign Language (ASL) finger spelling and an alphabet gesture recognition system was designed with ANN and constructed in order to translate the ASL alphabet into the corresponding printed and sounded English letters. The system uses a sensory Cyberglove and a Flock of Birds 3-D motion tracker to extract the gestures. The finger joint angle data obtained from strain gauges in the sensory glove define the hand shape while the data from the tracker describes the trajectory and orientation. The data flow from these devices is controlled by a motion trigger. Then, data is processed by an alphabet recognition network to generate the words and names. Our goal is to establish an ASL finger spelling system using these devices in real time. We trained and tested our system for ASL alphabet, names and word spelling. Our test results show that the accuracy of recognition is 96%.

1 Introduction

There are about two million deaf people in the United States. Some of them are born with hearing loss in both ears, while the others lose their hearing due to factors such as rubella, meningitis, and prolonged bouts of high fever. According to the National Center of Birth Defects and Developmental Disabilities, about one child out of every thousand is born deaf [1]. It is often very difficult for deaf people to interact with people outside the deaf community. Some devices, such as the Telecommunications Device for Deaf (TDD), help deaf people communicate [2].

American Sign Language (ASL) is the main mode of communication for most deaf people in the United States. However, most people outside the deaf community do not understand ASL. If ASL can be translated automatically into English text and speech, it will be much easier for deaf people to communicate with others. There are about 6000 signs in ASL to represent words.. The signs, which consist of complex body movements, are created by using the right hand, the left hand or both hands. Some signs also involve facial expressions.

Many researchers have studied sign language (SL) recognition and gesture recognition, but there are major difficulties due to the complexity of hand and body movements in SL expressions. SL recognition research can be categorized into three major classes: (i) computer vision based, (ii) data glove and motion sensor based, and (iii) a combination of these two methods. Computer vision based SL recognition relies on

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 157–164, 2005.

image processing and feature extraction techniques for capturing and classifying body movement and handshape when a deaf person makes a SL sign. On the other hand, data glove and motion tracker based SL recognition methods use a special data glove and a motion tracker for detecting handshape and movement. The third method includes a combination of techniques from each of these two methods.

In parallel to the advancement in sensor and computer technology, some successful computer vision based SL recognition systems have been developed [3-10]. There are also many studies in the area of glove and motion sensor based SL recognition [11-13]. These studies are generally based on Artificial Neural Networks (ANN) [3],[11],[13],[14], Hidden Markov Model (HMM) [4],[5],[7],[8],[10], and statistical methods [6],[9]. Some recently published papers include the use of both methods, which are used together for ASL recognition [14].

In this paper, we present the design of ASL alphabet recognition and a finger spelling system using ANN. The neural network for alphabet recognition is controlled by a motion trigger. If the trigger finds the velocity of the hand below a threshold value, then the data from the Cyberglove™ and Flock of Birds® will be sent to the alphabet recognition network. The data flow between the input devices and the recognition process will be turned off until a high speed hand movement reactivates it. The system is able to recognize the ASL alphabet and to spell words and names into the corresponding printed and sounded English letters with the ANN methods.

2 American Sign Language and Spelling

In the United States, American Sign Language (ASL) is the native language of many deaf people, and it is the language used by many deaf adults among themselves. There are about two million deaf people in the USA. ASL is the second most widely used non-English language in the United States after Spanish.

American Sign Language is a primary language. ASL (like other sign languages) is not derived from any other spoken language, although its coexistence with English in a bilingual environment allows it to be influenced in a number of ways. ASL's nearest sign language relative is French Sign Language (FSL). Thomas Gallaudent and Laurent Clerc brought French sign to the United States in 1817[15]. It is a fact that deaf people who use different sign languages cannot understand each other.

American Sign Language has 36 hand shapes, 26 letters and 6000 words. It is created by complex body movements, i.e., by using the right hand or the left hand or both hands, and facial expressions. Finger-spelling is the presentation of each letter of the alphabet by a sign. There are 26 signs from A to Z. Finger-spelling is used for names and in spelling words that have no signs.

3 Hardware of System

Our primary physical connection to the world is through hands. We perform most everyday tasks with them, however we use constrained peripheral devices such as mouse, keyboard and joystick to work with a computer and computer controlled applications. Glove based input devices were designed to overcome this limitation [16].

Commercial devices, such as the VPL data glove and Mattel power glove, have led to an explosion of research and development projects using electronic glove interfaces to computer applications and computer controlled devices. These applications are virtual reality, video games, scientific visualization, puppetry, and gesture based control.

We use a right-hand Cyberglove™ (Figure 1) to retrieve the finger joint angle values for gesture features. The glove has 18 sensors which measure the bending angles at various positions and the frequency of data collection is up to 150 Hz. We use 15 sensors of the glove: three sensors for the thumb, two sensors for each of the other four fingers, and four sensors between the fingers.

To track the position and orientation of the hand in 3-D space, the Flock of Birds® motion tracker (Figure 2) mounted on the hand wrist is used. The receiver is located in a DC pulsed magnetic field and its effective range is up to 8 feet around the transmitter. The measuring frequency is up to 144 Hz.

Open Inventor SDK (Software Development Kit) is used in the software development for the 3-D scene rendering and interactive programming. It is a high-level tool kit developed in OpenGL for graphic rendering and user interactions. We use the Microsoft® Speech SDK for the programming of speech synthesis.

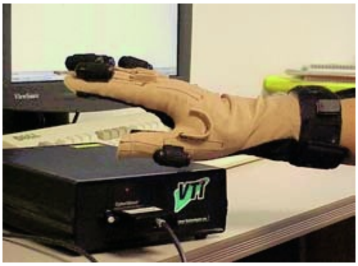


Fig. 1. Cyberglove™ with 18 sensors

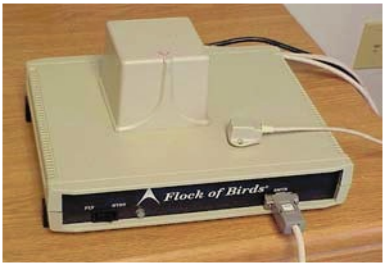


Fig. 2. Flock of birds® 3-D motion tracker

4 Artificial Neural Network Model

A backpropagation algorithm is used for training the ANN model. The basic structure and formulation of backpropagation is summarized here. Training a neural network involves computing weights so as to get an output response to the input within an error limit. The input and target vectors make up a training pair. The backpropagation algorithm includes the following steps [17]:

1. Select the first training pair and apply the input vector to the net.
2. Calculate the net output.
3. Compare the actual output with the corresponding target and find the error.
4. Modify the weights so as to reduce the error.

These steps are repeated until the error is within the accepted limits. In step 2, output sets for test inputs are calculated. If they are the same within an error range as the expected sets, then it is considered that the net has learned the problem, and the final weights are stored so that they can be reused when needed. The developed ANN has a multi-layer feedforward structure as shown in Figure 3. The variable definitions are given as follows [18],[19]: **L=0:** input layer, **L=1:** hidden layer, **L=2:** output layer, **W$_{1,ji}$:** weight matrix between the input layer and the hidden layer. **W$_{2,tj}$:** weight matrix between the hidden and the output layer, **B$_{1,j}$:** bias values of hidden neurons, **B$_{2,t}$:** bias values of output neurons.

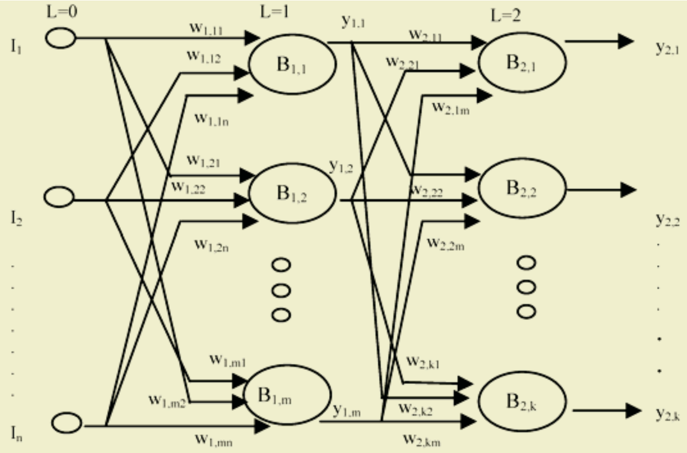


Fig. 3. A multi-layer feedforward net structure

Equation (1) gives the output of the hidden layer.

$$\begin{aligned} y_{NET\ 1,j} &= \sum_{i=1}^{n} w_{1,ji} I_i + B_{1,j}, \\ y_{1,j} &= f_j \left[y_{NET\ 1,j} \right] \quad \text{j = 1,2,..., m.} \end{aligned} \tag{1}$$

Equation (2) gives the output of the output layer.

$$\begin{aligned} y_{NET\ 2,t} &= \sum_{i=1}^{m} w_{2,tj} y_{1,j} + B_{2,t}, \\ y_{2,t} &= f_t \left[y_{NET\ 2,t} \right] \quad \text{t = 1,2,..., k.} \end{aligned} \tag{2}$$

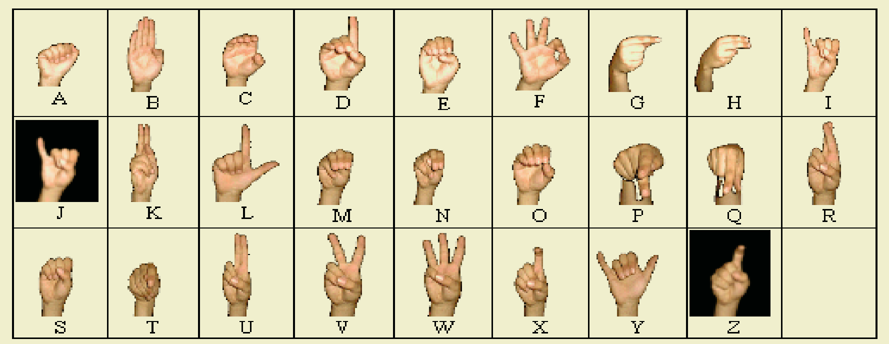
Activation function;

$$f(y_{NET}) = \frac{1}{1 - e^{-y_{NET}}}. \tag{3}$$

5 ANN Model Design for ASL Alphabet Recognition

As explained earlier, ASL finger spelling is the presentation of each letter of the alphabet by a sign. There are 26 signs from A to Z. Finger spelling is used for names and for some words which are not represented as ASL words. The alphabet set of American Sign Language used in the training and test set for the ANN are given in Table 1.

Table 1. Alphabet set of American Sign Language [15]


A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

Although most of the ASL alphabet letters depend on finger bending, some of them also depend on hand orientation and two of them are dynamic. There are some similarities between g and q, h and u, k and p. These couples have basically the same hand shape, but their hand orientation differs from the others. There are hand shape similarities between i and j, and x and z, but, j and z are dynamic characters.

A multi-layer ANN is designed to recognize the ASL alphabet. The input to the network consists of 18 data elements, fifteen from Cberglove plus three from Flock of Bird. The proposed ANN, a multi-layer feedforward network, consists of 18 input neurons, 20 hidden neurons, and 26 output neurons. A Levenberg-Marquardt back-propagation algorithm is used for training. The ANN is trained and tested for two different data sets: single-user data and multi-user data. The output vector consists of 26 elements, the maximum of which corresponds to an alphabet or sign. The training set is composed of two files, input and output. The input file contains Cyberglove data and Flock of Bird data belonging to 26 characters, and each character has 18 data values which are provided as inputs to the ANN. An example of training data coding is given in Table 2. There are 78 data sets (three data sets per character; 26*3) in the input file.

Table 2. Output characters

			Outputs					
	Samples	Gestures	**O1**	**O2**	**.**	**.**	**.**	**O26**
Alphabet	**1**		1	0	.	.	.	0
	2		1	0	.	.	.	0
	3		1	0				0
	1		0	1	.	.	.	0
	2		0	1	.	.	.	0
	3		0	1	.	.	.	0
		: **.**						
	1		0	0				1
	2		0	0	.	.	.	1
	3		0	0	.	.	.	1

The design of the recognition system is shown in Figure 4. It consists of three parts: selection of input data, trained network and output selection.

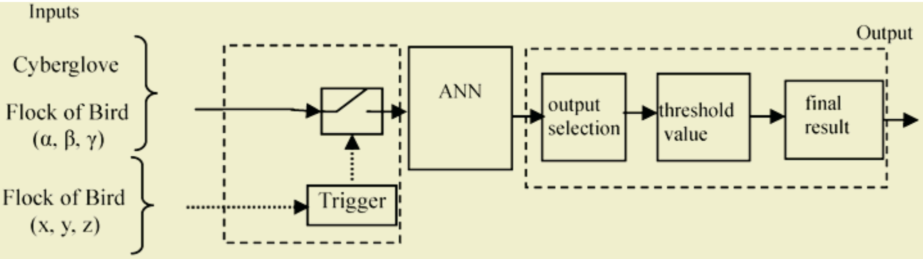


Fig. 4. Block diagram of ASL word recognition system

The trigger controls the input data of the recognition network. When the trigger finds the velocity of the hand below a threshold value, in our case 0.05 (unit/second), then the data from the Cyberglove™ and Flock of Birds® will send be sent to the alphabet recognition network. The data flow between input devices and the recognition process will be turned off until a high speed hand movement reactivates it. The reactivation velocity is set at 0.5. The three-layer word recognition network has 18 inputs, 20 hidden neurons and 26 output neurons. In the final part, the selection output block determines the maximum output of the ANN, and then the threshold block checks whether the maximum output value is acceptable or not. If it is not acceptable, the user can try the same character. The system output is stored in a variable to form a word. After finishing the word sign, the corresponting sound of the recognized word is generated by touching a virtual red ball. The recognized letters and word are displayed on the screen using Open Inventor interface. Some of the ASL letters recognized by our system are shown in Figure 5, and some of spelled words or names are shown in Figure 6.

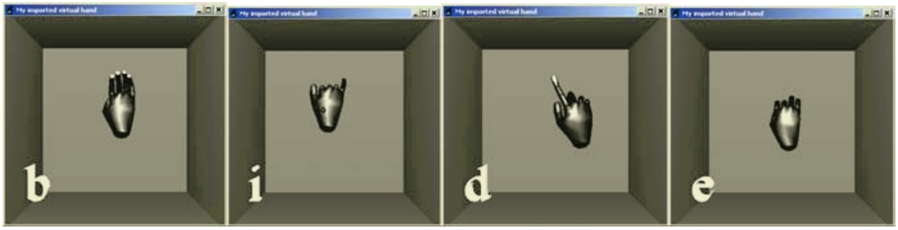


Fig. 5. Examples of recognized Alphabet gesture by our system

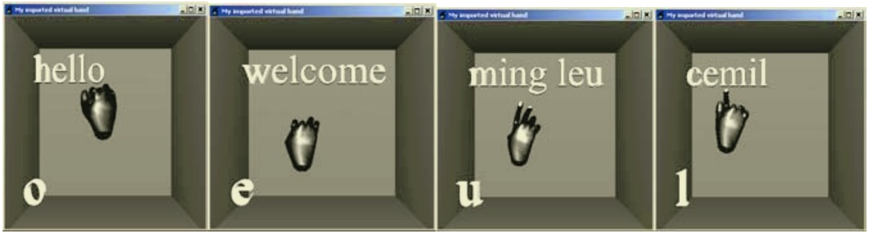
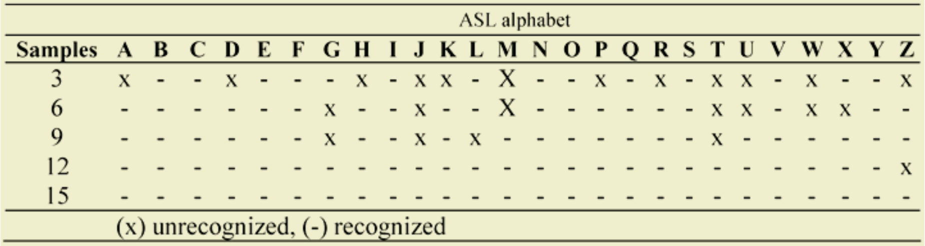


Fig. 6. Examples of Spelled words or names by our system

6 Test Results

Two ASL alphabet recognition and finger spelling models are developed. The ASL alphabet recognition with single user model was first trained with data from three samples. When that was not effective, we trained with six, nine, then twelve and finally, fifteen samples. Similarly, the multi-user model was first trained with data from three samples; when that was not effective, we trained with six, nine, than twelve and finally, fifteen data samples. At the testing stage, real time data are used. The models were tested starting with *A* and ending with *Z*. The testing results are given in Table 3 and Table 4.

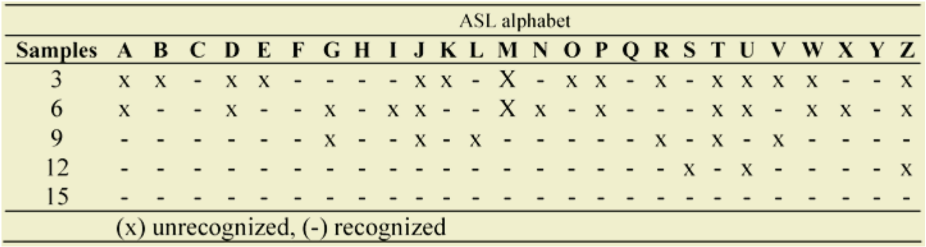
Table 3. Test result for single user



	ASL alphabet																									
Samples	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**
3	x	-	-	x	-	-	-	x	-	x	x	-	X	-	-	x	-	x	-	x	x	-	x	-	-	x
6	-	-	-	-	-	-	x	-	-	x	-	-	X	-	-	-	-	-	-	x	x	-	x	x	-	-
9	-	-	-	-	-	-	x	-	-	x	-	x	-	-	-	-	-	-	-	x	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	x
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

(x) unrecognized, (-) recognized

Table 4. Test result for multiple users



	ASL alphabet																									
Samples	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**
3	x	x	-	x	x	-	-	-	-	x	x	-	X	-	x	x	-	x	-	x	x	x	x	-	-	x
6	x	-	-	x	-	-	x	-	x	x	-	-	X	x	-	x	-	-	-	x	x	-	x	x	-	x
9	-	-	-	-	-	-	x	-	-	x	-	x	-	-	-	-	-	x	-	x	-	x	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	x	-	x	-	-	-	-	x
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

(x) unrecognized, (-) recognized

7 Conclusions

An American Sign Language alphabet recognition and finger spelling system is being developed using ANN to translate the ASL words into English. The system uses a sensory glove, Cyberglove, and a Flock of Birds 3-D motion tracker to recognize the gestures. The system was trained and tested for single and multiple users. Test results have shown that the proposed techniques are capable of real-time recognition performance. The recognition accuracy of the system is about 96%, and the system is capable of spelling all English names and words.

Acknowledgments

This research is partially supported by a National Science Foundation award (DMI-0079404) and a Ford Foundation grant in the United States, as well as by the Intelligent Systems Center at the University of Missouri–Rolla.

References

1. National Center of Birth Defects and Developmental Disabilities home page, http://www.cdc.gov/ncbddd/bd/bdres.htm
2. Chapin, W., Kramer, J., Haas, C., Leifer, L., Macken, E.: TeleSign: A Sign Language Telecomm-unication System. IEEE (1992)
3. Wysoski, S.G., Lamar, M.V., Kuroyanagi, S. , Iwata, A.: A Rotation Invariant Approach on Static Gesture Recognition Using Boundary Histograms and Neural Networks. Proceeding of the 9th International Conference on Neural Information Processing (ICONIP'02), **4** (2002) 2137-2141
4. Vogler, C. , Metaxas, D.: ASL Recognition Based on a Coupling Between HMMs and 3D Motion Analysis. Proceedings of Sixth IEEE International Computer Vision Conference, (1998) 363-369
5. Lalit, G. , Suei, M: Gesture-based Interaction and Communication: Automated Classification of Hand Gesture Contours. IEEE Transactions on Systems, Man, and Cybernetics part c: Application and Reviews, **31** (2001) 114-120
6. Yang, M.H. , Ahuja, N.: Recognizing Hand Gestures Using Motion Trajectories. IEEE (1999) 466-472
7. Stanner, T., Pentland, A.: Real-time American Sign Language Recognition from Video Using Hidden Markov Models. Proceeding of Computer Vision International Symposium, (1995) 265-270
8. Vogler, C. , Metaxas, D.: Adapting Hidden Markov Models for ASL Recognition by Using Three-dimensional Computer Vision Methods. Proceeding of the IEEE International Conference on Systems, Man , Cybernetics, Orlando (1997) 12-15
9. Cui, Y. , Weng, J.: A Learning-based Prediction and Verification Segmentation Scheme for Hand Sign Image Sequence. IEEE Transactions on Pattern Analysis and Machine Intelligence, **21** (1999) 798-804
10. Vogler, C. , Metaxas, D.: Parallel Hidden Markov Models for American Sign Language Recognition. IEEE Proceeding of International Computer Vision Conference (1999) 116-122
11. Allen, M.J., Asselin, P.K., Foulds, R.: American Sign Language Finger Spelling Recognition System. 29th Bioengineering Conference proceeding, IEEE (2003) 22-23
12. De Marco, R.M. , Foulds, R.A.: Data Recording and Analysis of American Sign Language. IEEE 29th Proceeding of Bioengineering Conference, (2003) 49-50
13. Wang, H.G., Sarawate, N.N. , Leu, M.C.: Recognition of American Sign Language Gestures with a Sensory Glove. Japan USA Symposium on Flexible Automation, Denver, CO (2004)
14. Lee, L.K., Kim, S., Choi, Y.K. , Lee, M.H.: Recognition of Hand Gesture to Human-computer Interaction. IEEE (2000) 2177-2122
15. Sternberg, M.L.A.: American Sign Language Dictionary (Revised Edition). Harper Perennial, ISBN: 0-06-273275-7 (1994)
16. Sturman, D.J. , Zelter, D.: A Survey of Glove-based Input. IEEE Computer and Applications, (1994) 30-39
17. Lippman, R.P.: An Introduction to Computing with Neural Nets. IEEE ASSP (1987) 2-22
18. Abulafya, N.: Neural Networks for System Identification and Control. MSc Thesis, University of London (1995)
19. Narendra, K.S.: Adaptive Control Using Neural Networks. In Neural Networks for Control (Miller, W.T., Sutton, R.S., and Werbos, P.J., edds) 3rd printing, MIT Cambridge (1992)

Fingerprint Minutia Recognition with Fuzzy Neural Network

Guang Yang, Daming Shi, and Chai Quek

School of Computer Engineering, Nanyang Technological University
Singapore 639798
asdmshi@ntu.edu.sg, ashcquek@ntuix.ntu.ac.sg

Abstract. Falcon-ART is a fuzzy neural network that can be used as fuzzy controllers or applied to areas such as pattern recognition, forgery detection and data analysis. Our previously proposed Falcon-DIC has stronger noise tolerance capability compared to the original Falcon-ART by employing a new clustering technique called Discrete Incremental Clustering (DIC). In this paper, Falcon-DIC is applied to perform direct gray-scale minutiae extraction. Fingerprint features extraction, or minutiae extraction, is an essential part of fingerprint identification systems. Most existing minutiae extraction methods require image preprocessing, such as binarization and thinning. Since these image processing techniques results in the loss of valuable information, our proposed approach can extract minutia directly from gray-scale fingerprint images. Experimental results show that Falcon-DIC based minutiae extraction has invariant ability to rotation and good performance on true acceptance.

1 Introduction

The most popular strategy for fingerprint identification is minutiae matching [1]. The simplest pattern of the minutiae-based representations consists of a set of minutiae, including ridge endings and bifurcations defined by their spatial coordinates. Each minutia is described by its spatial location associated with the orientation. However, the minutia correspondences are difficult to obtain due to several factors such as the rotation, translation and deformation of the fingerprints, the location and direction errors of the detected minutiae as well as the presence of spurious minutiae and the absence of genuine minutiae.

Most of the existing minutiae extraction methods require image preprocessing, such as binarization and thinning. The deficiencies of this scheme lie in the followings [2]: (1) The pre-processing procedures are tedious and time consuming, (2) Some valuable information may be lost during the process, (3) Some of the processing procedures, especially binarisation, have been proven to be unsatisfactory when used on low quality images (4) The errors generated during pre-processing tend to propagate from one stage to another.

Maio and Maltoni [3] proposed an approach to direct gray-scale minutiae detection in fingerprints. This technique is based on a ridge line following algorithm that follows the image ridge line until a termination or a bifurcation occurs. In spite of a greater conceptual complexity, they have shown that the technique has less computa-

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 165–170, 2005.

tional complexity than those requiring binarization and thinning. However, this technique was developed ad hoc, and lacked local criteria to verify the reliability of each minutiae detection. They tried to solve this problem with neural networks [4].

The remainder of this paper is organized as follows. Section 2 introduces Falcon-ART, followed by Falcon-DIC in Section 3. Section 4 describes how to apply Falcon-DIC to fingerprint minutiae extraction. Section 5 gives the experimental results and analyses, and the conclusions are presented in Section 6.

2 Work Mechanism of Falcon-ART

The *Falcon-ART* architecture developed by Lin [5,6] is a highly autonomous system. It has 5 layers and generates fuzzy rules of the form:

If *input 1 is* $L_{1\phi}$ *and input 2 is* $L_{2\phi}$ *and…input n is* $L_{n\phi}$
Then *output 1 is* $L'_{1\theta}$ *and output 2 is* $L'_{2\theta}$ *and…output m is* $L'_{m\theta}$

The above fuzzy rule has 5 elements: condition section, input linguistic terms, consequent section, output linguistic variables, and the IF-THEN rule. The labels *input 1*, *input 2*,…, *input n* denote the input linguistic variables and the labels *output 1*, *output 2*,…, *output m* denote the output linguistic variables, where *n* and *m* are the number of input and output respectively. They represent entities like height, speed and weight. The labels $L_{1\phi}$, $L_{2\phi}$,…, $L_{n\phi}$ denote the input linguistics terms and the labels $L'_{1\theta}$, $L'_{2\theta}$,…, $L'_{m\theta}$ denote the output linguistic terms. These input and output linguistic terms are represented as trapezoidal fuzzy sets in *Falcon-ART*. They represent fuzzy concepts such as tall, short, medium, fat and thin. Each of the 5 layers in *Falcon-ART* is mapped to the respective elements of the fuzzy rule.

In training, the inputs to *Falcon-ART* are complementarily coded vector **x'** and complementarily coded desired output vector **d'**. The output from *Falcon-ART* is $\mathbf{y} \equiv (y_1, y_2, \ldots, y_m)^T$. The complementary coded vector **x'** is obtained by:

$$\mathbf{x'} \equiv (''x_1, ''x_1^c, ''x_2, ''x_2^c, \ldots, ''x_n, ''x_n^c)^T = (''\dot{x}_1, 1-''x_1, ''x_2, 1-''x_2, \ldots, ''x_n, 1-''x_n)^T \quad (1)$$

Where $''\mathbf{x} = (''x_1, ''x_2, \ldots, ''x_n)^T = \mathbf{x} / \|\mathbf{x}\|$,
$\mathbf{x} \equiv (x_1, x_2, \ldots, x_n)^T$ = input training vector,
$\|\mathbf{x}\|$ = magnitude of vector **x,**
$''x_i^c = 1-''x_i$ and $''x_i \in [0, 1]$ for $i = 1 \,..\, n$ and
n = number of input linguistic variables.

The same applies for complementary coded desired output vector **d'**. In applying the complementary-coding technique to *Falcon-ART*, all training vectors (either input training vectors or desired output vectors) are transformed to their complement-coded forms during preprocessing, and the transformed vectors are then used for training.

3 Falcon-DIC

The inherent deficiencies of Falcon-ART include: (1) unsatisfactory classification performance when the classes of input data are very similar to each other; (2) susceptible to noisy training data and outlier. In this section, Falcon-DIC will be proposed to address these deficiencies. And at the same time, it holds the advantages of Falcon-ART.

Falcon-DIC employs a new clustering method at the structural learning stage, called Discrete Incremental Clustering (DIC) technique. The DIC does not require having prior knowledge of the number of clusters and is able to robustly handle noisy data. And it works on raw numerical value of a training data without preprocessing, which increase training efficiency. If the fuzzy set for a particular input/output dimension already exists, then it will not be recreated. Thus, DIC ensures that each fuzzy set is uniquely defined and this serves as a basis to form a consistent fuzzy rule base.

The DIC technique has 5 parameters to control the learning process: a plasticity parameter β , a tendency parameter TD, and input threshold IT, an output threshold OT and a fuzzy set support parameter SLOPE. Currently, the selection of these parameters is heuristic and varies with different tasks. Figure 1 below shows the details of the DIC algorithm.

```
Algorithm DIC
    Initialize Step and Slope between 0 and 0.5
    IT = $\beta$ = TD = 0.5
    For all training vectors {
        For all dimensions of training vector{
            If there is no cluster for current dimension
                Create a new cluster
            Else{
                Find the best fit cluster winner for current data point
                Winner cluster has the maximum output with current data point
                If the output is greater than the threshold
                    Update the kernelt of the winner cluster
                Else
                    Create a new cluster
            }End If-Else
        }End for all dimensions
    }End for all training vectors
End
```

Fig. 1. Algorithm of Discrete Incremental Clustering

4 Minutia Extraction with Falcon-DIC

In order to define a minutia point, we need information not only from the point itself, but also from its surrounding edges. Nine mask windows are used to model a minutiae point in this research as shown in Figure 2. There is a window *Body* and *windows #1 - #8*. For *Body*, 4×4 mask window is just about enough to cover a minutia point. For *windows #1 - #8*, it depends on the width of the edges taken. The average pixel values in each window are the input to the Falcon-DIC network.

Data from minutia (termination or bifurcation) points identified by the domain expert are recorded and subsequently used to train the network for extraction of fuzzy rule(s). We apply the approach of *adjusted fuzzy sets* with *fixed fuzzy rules* [8], in which fuzzy rules are decided at the beginning stage, and the fuzzy sets are adjusted from a set of training data.

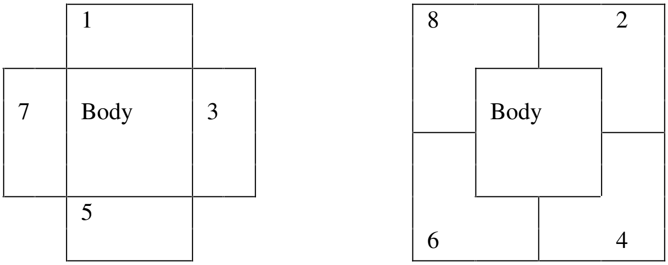
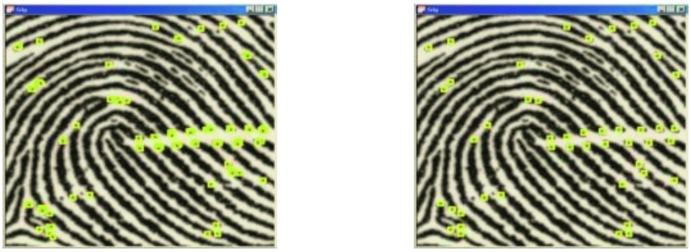


Fig. 2. Modeling windows

In order to improve the system's performance, a filtering technique is employed in this research to reduce the number of false acceptances. For each pixel in image, store its own rule strength and rule strength of surrounding pixels in defined region, then compare and find the pixel with greatest rule firing strength and its coordinate, and then set rule strength of the other pixels to zero. Figure 3 shows an example of the result for termination detection before and after filtering. We can see that some fake minutiae are removed after filtering.



(a) (b)

Fig. 3. Results before and after filtering. (a) minutia before filtering. (b) minutia after filtering

5 Experimental Results

Two sets of experiments are conducted separately on ridge-ending and bifurcation separately. In each experiment, fingerprint images from three individuals are used. In general, about 20% to 30% of the total visually detected minutiae in each image are utilized for training. In our initial experiments, we assess and analyze 6 different fingerprints. Experimental results are recorded in terms of two classes, namely *True* and *False*. The *True* class is subdivided into *True Acceptance (TA)*, i.e. correctly classified minutiae and *True Rejection(TR)*, i.e. missed minutiae. The *False* class is subdivided into *False Acceptance(FA)*, i.e. wrong minutiae and *False Rejection* (FR), i.e. correctly rejected pixels. Only *TA*, *TR* and *FA* are tabulated in the results.

5.1 Experiments on Rotational Invariance

This experiment aims to test the system's invariant ability to rotation. In this set of experiments, fingerprint images are rotated at 10°, 20°, and 30° each. Training data gathered from the non-rotated image. The trained system is then used to perform minutiae extraction on the rotated images. Due to image cropping, some minutia points in the non-rotated images are no longer found in their rotated counterparts. As such, Table 1 for recording the *Matched True Acceptance* contain results for minutiae found in both the non-rotated images and their non-rotated counterparts.

Table 1. Averaged result for Rotational Invariance

Rotation Angle		TA (%)	TR (%)	FA		Matched TA (%)
				No.	%	
10°	Ridge-ending	90.2	9.8	91	0.39	96.4
	Bifurcation	63.9	36.1	54	0.23	95.0
20°	Ridge-ending	83.0	17.0	91	0.39	90.8
	Bifurcation	60.7	39.3	68	0.29	90.1
30°	Ridge-ending	87.9	12.1	91	0.39	93.9
	Bifurcation	58.4	41.6	56	0.24	90.3

The averaged matched true acceptance for all the rotations is above 90%. This is however still considered quite high. Hence, the results indicated that the system is highly invariant to rotation.

5.2 Experiments on Robustness Against Gray Scale Change

This experiment aims to test if the fuzzy rule(s) extracted from a fingerprint image can be used on other fingerprint images. The trained system is then used to perform minutiae extraction on other individuals' fingerprint images. A fingerprint image with significant difference in gray-scale levels as compared with the rest is used here to test the system's response to the difference. Table 2 shows the average result for this experiment.

Table 2. Averaged performance on robustness against gray scale change

Minutia Type	TA (%)	TR (%)	FA	
			No.	%
Ridge-ending	80.4	19.6	57	0.24
Bifurcation	66.4	33.6	47	0.20

The numbers of FA obtained for ridge-ending and bifurcation are pretty low here. From Table 2, it can be concluded that rules from another fingerprint can be used on a different fingerprint provided the gray-scale levels of the fingerprints are about the same.

6 Conclusion

In this paper, we have presented a novel approach for extracting minutiae directly from gray-scale fingerprint images based on the Falcon-DIC fuzzy neural network.

The technique is based on defining various mask windows to capture the information (level of brightness or darkness) surrounding a minutia point, train the Falcon-DIC network with samples of this information, and have it automatically extract the minutiae after training. Overall, the system has shown high invariant ability to rotation and performed reasonably well in terms of the percentage of True Acceptance (TA). However, the TA for bifurcation is relatively lower than that for termination. This can be attributed to insufficient training sample, imbalanced training data.

References

1. Ratha, N.K., Kalle Karu: A Real Time Matching System for Large Fingerprint Databases IEEE Transactions on Pattern Analysis and Machine Intelligence, **18** (1996) 799-813
2. Liu, J., Huang, Z., Chan, K.L.: Direct Minutiae Extraction from Gray-Level Fingerprint Image by Relationship Examination. Proceedings of International Conference on Image Processing, **2** (2000) 427-430
3. Maio, D. , Maltoni, D.: Direct gray-Scale Minutiae Detection in Fingerprints. IEEE Transactions on Pattern Analysis and Machine Intelligence, **19** (1997) 27-40
4. Maio, D. ,Maltoni, D.: Neural Network Based Minutiae Filtering in Fingerprints. Proceedings of 14th International Conference on Pattern Recognition, **2** (1998) 1654-1658
5. Lin, C.T., Lee, C.S. : Neural Fuzzy Systems. Prentice Hall, Upper Saddle River, USA (1996)
6. Lin, C. T., Lin, C.J.: An ART-Based Fuzzy Adaptive Learning Control Network. IEEE Transactions on Fuzzy Systems, **5** (1997) 477-496
7. Tung, W.L., Quek, C.: A Novel Approach to the Derivation of Fuzzy Membership Functions using the Falcon-MART Architecture. Pattern Recognition Letters, **22** (2001) 941-958
8. Tanaka, K.: An Introduction to Fuzzy Logic for Practical Applications. Springer, USA (1997)

Fingerprint Classification Based on Curvature Sampling and RBF Neural Networks

Xuchu Wang[1], Jianwei Li[1], and Yanmin Niu[2]

[1] Key Lab on Opto-Electronic Technique of State Education Ministry, Chongqing University
Chongqing 400044, China
Seadrift@sina.com
[2] College of Physics and Information Techniques, Chongqing Normal University
Chongqing 400047, China
nym4612@yahoo.com.cn

Abstract. This paper presents new five-class fingerprint classification algorithms based on combination of curvature sampling and radial basis function neural networks (RBFNNs). The novel curvature sampling algorithm is proposed to represent tendencies and distributions of ridges' directional changes with 25 sampled curvature values. The normalized and organized curvature data set is as input feature vector for RBFNNs and the output is formed result. The probability density is defined to describe the clustering ability of an input vector and used to select hidden layer neurons adaptively. The algorithms are validated in fingerprint databases NIST-4 and CQUOP-FINGER, the best classification accuracy is 91.79% at 20% rejection rate. It shows good balance for classification of arch and tented arch types and it needn't detect singular points. The result indicates that this algorithm can satisfy the requirement of fingerprint classification well and provides a new and promising approach.

1 Introduction

Fingerprint classification is efficient in determining when two fingerprints do not match in large fingerprint databases. The basis of classification is that although every fingerprint is unique, its pattern can be classified into one out of five categories: whorl(W),left loop(L),right loop(R), arch(A), and tented arch(T) according to global ridge and furrow structures[1],[2],[6].

Some feasible approaches to the topic have been presented in the literatures[1],[2],[3],[4],[5],[6] and they are mainly based on ridges direction. In our experiments of fingerprint classification, we find that the dominating basis people use to classify fingerprints is the tendency of changes of ridges direction called curvature. Curvature includes information of average direction and variance direction and demonstrates the rule of ridge changes better than direction and variance respectively.

So in this paper, we propose new fingerprint classification algorithms based on curvature distributing and radial basis function neural networks (RBFNNs). Our actual algorithms proceed in two stages: (i) extracting curvature data set from fingerprint image. In order to decrease computing complexity and get representative data at the same time, we find a way to use some sampled curvatures to replace all curvatures

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 171–176, 2005.

of every points in fingerprint image and it means curvature sampling; (ii) classifying sampled curvature data set by RBFNNs, In the training algorithm of hidden layer, we define the probability density of input vector to describe the clustering ability of input vectors and use subtractive clustering method to determine clustering center vector adaptively. Our algorithms are validated in fingerprint database NIST-4 and CQUOP-FINGER and the best classification accuracy is 91.79% at 20% rejection rate. Although this accuracy is not the highest one, it has a good balance of classification for arch (A) and tented arch (T) which wilders many other researches. Furthermore, it needn't detect singular points. The result indicates that the combination of curvature sampling and RBFNNs can satisfy the requirement of fingerprint classification and provides a new and promising research approach.

2 Algorithm of Curvature Sampling

Curvature distribution in a fingerprint image is a valid feature. The main steps to extract representative curvature data include normalization, valid region extraction, block directional estimation and sampled curvatures of ridges. In this paper we focus on illustrating the innovative curvature sampling. The algorithm is as the follows:

Step 1. Location of Sampled Points

Suppose the width and height of the circumscribed rectangle of a valid region are *Wv*, *Lv* and the center point is *C(Wv/2, Lv/2)*. Separate the center point in 16 directions using 8 lines and the angle among every two lines is 22.5° (See Figure 1). Suppose the clockwise lines are *Li(i=0,1,…7)* and note that two endpoints of each line are in the boundary of valid region but not in the boundary of the circumscribed rectangle. The length of each line can be demonstrated by the following equation:

$$L_i = 2L_b + 4L_c . i = 0,1,..7 . \tag{1}$$

where L_b is a constant parameter which represents the distance of a sampled point and valid region's boundary (selected from 3 to 7). L_c is a distance of neighbor sampled points on a same line L_i. Using Eq.1, 33 sampled points are gotten and organized in three circles. Then to eliminate the points in middle circle to 8 points, some feasible ways include eliminating points by length relationships of *Li*, by directional relationships of *Li*, by the curvature value of sampled point in *Li* and etc. Here we use even or odd relationships of *Li* and get 25 sampled points.

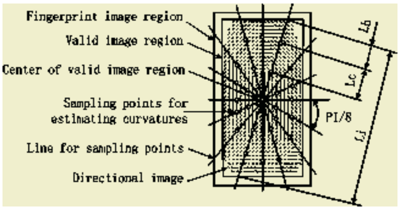


Fig. 1. Location of sampled points in directional image

Step 2. Curvature Estimation of Sampled Points.
Define a window *W_c(k)* of size $r{\times}r$ (5×5) centered at sampled point $P_k\,(0 \le k \le 24)$. Suppose the horizontal and vertical valid direction number are r_h, $r_v\,(0 \le r_h,\ r_v \le r)$, then compute mean and variance of direction in each window $W_c(k)$ by the following equations:

$$M_o(k) = \sum_{i=0}^{r_h-1}\sum_{j=0}^{r_v-1}\beta(i,j) \Big/ r_h \times r_v \,. \quad k = 0,1,2,..,24 \cdot \tag{2}$$

$$V_o(k) = \sum_{i=0}^{r_h-1}\sum_{j=0}^{r_v-1}\left|\beta(i,j) - M_o(k)\right|. \Big/ r_h \times r_v \cdot \tag{3}$$

If $V_o(k)$ is bigger than a constant parameter $V_C(k)$, $W_c(k)$ is regarded as invalid, don’t estimated its curvature, replace it with curvatures of its neighborhoods, then for each direction in $W_c(k)$, compute its curvature in a given direction $M_o(k)$. We propose a simplified curvature computing equation C_{ok}, $C_o^{'}$:

$$\begin{aligned} C_{ok}(p,s) &= \left|\beta(p^{'},s^{'}) - \beta(p,s)\right| / R \qquad p,s = 0,1,...r-1, \\ p^{'} &= p + R\cos(M_o(k)), \\ s^{'} &= s + R\sin(M_o(k)). \end{aligned} \tag{4}$$

$$C_o^{'}(k) = F\{C_{ok}(p,s)\}. \tag{5}$$

where R is a constant parameter and it can be equal to r. $F\{\cdot\}$ means a kind of operation such as computing average, maximum, middle value and etc(we use the first one). In computing procedure, the overlapping of $W_c(k)$ in some parts doesn’t influence analytical results.

Step 3. Post Treatment.
We observe in many experiments that $C_{ok}^{'}$ is in range *[0, 0.78]* because of the consistency of ridges. It is necessary to filter the sampled curvature matrix using a low pass filter in order to modify incorrect curvatures (we propose using 3x3 Gaussian filter). Stretch all of the curvature to *[0,1]* linearly and use a vector to organize them as input vector of neural networks by a stable order (we use the order which is from out circle to inner circle by clockwise). Note that this arrangement is very important because it reflects the distribution of direction changes.

$$X = \left(C_o^{'}(1), C_o^{'}(2)..., C_o^{'}(24)\right)^T. \tag{6}$$

Figure 2 shows some extraction results of curvature sampling.

3 Adaptive Determination of Clustering Centers of RBFNNs

As shown in Figure 2, sampled curvatures in different fingerprint classes shows different distributing states and there are some local peaks around singular points in a

(a) whorl, (b) left loop, (c) right loop, (d) tented arch,(e) arch.

0.38	0.29	0.08	0	0.01
0.18	**0.67**	0.02	0.17	0
0.08	**0.8**	0.13	0.43	0.11
0.67	0.34	**1**	**0.66**	0.21
0.64	0.23	0.33	**0.69**	**0.71**

(f) sampled curvatures of (a)

0.51	**0.61**	0.25	0.15	0.04
0.58	**0.56**	0.03	0.05	0.2
1	**0.79**	0.08	0	0.05
0.17	0.04	0.26	0.2	0.18
0.21	0.19	0.42	**0.69**	0.23

(g) sampled curvatures of (b)

0.04	0.19	**0.7**	**0.38**	0.11
0.12	0.09	**0.98**	0.13	0
0.08	0.05	0.06	0.09	0.02
0.1	0.14	0.1	0.16	0.01
0.15	**1**	0.04	0.18	0.1

(h) sampled curvatures of (c)

0.44	0.44	**0.54**	0.33	0.28
0.22	**0.61**	**0.66**	**0.57**	0.21
0.1	**0.67**	**1**	0.35	0.07
0.04	**0.65**	**0.74**	0.26	0.09
0.11	0.34	**0.82**	0	0.16

(i) sampled curvatures of (d)

0.17	0.35	**0.56**	0.36	0.27
0	**0.92**	**0.95**	**0.62**	0.17
0.03	**0.64**	**1**	**0.49**	0.16
0.09	**0.7**	**0.5**	**0.52**	0.26
0.15	0.39	0.34	0.28	0.15

(j) sampled curvatures of (e)

Fig. 2. Fingerprint images with sampled points and their sampled curvatures

fingerprint image although we don't detect core and delta points. We can classify the fingerprint by local combination of curvatures. So different from Ref.[4],[5],[6], we select RBFNNs as classifier. Intrinsically, whose activation function for hidden layer is locally acting and is the main reason we select it.

In order to divide feature space using least neurons of hidden layer and according to the intimate relationships of curvature data in local regions, we define probability density and construct clustering centers adaptively by distinguishing the difference degree of probability densities in input space [9]. The algorithm is as the follows:

Input training pattern $(X_1, X_2, ...X_M)$ where M is the number and define the probability density of vector X_m in a normalized hypersolid:

$$D_m = \sum_{p=1}^{M} \exp\left(-\left\|X_p - X_m\right\| / (0.5\gamma_a)^2\right) \quad m = 1,2,...M\,. \tag{7}$$

where γ_a is a receptive neighborhood of X_m. If D_m is high, it means that there are many input vectors around X_m. Compute all probability densities of training pattern and let $D_{C_1} = \max \; D_m$. So the corresponding input vector X_{C_1} is the first clustering center, then modify the probability densities of other input vectors by the following equation:

$$D_m^{'} = D_m - D_{C_1} \sum_{p=1}^{M} \exp\left(-\left\|X_p - X_{C_1}\right\| \Big/ (0.5\gamma_b)^2\right) \quad m = 1,2,...M\,. \tag{8}$$

where γ_b is a positive parameter and can distinguish two clustering centers. Only $\gamma_b > \gamma_a$, the two clustering centers can be correctly separated.

Then select next clustering center by Eq.7 and modify the probability densities of other input vectors by Eq.8 until criterion $D_{\max} < \gamma D_{C_1}$ is true. The physical meaning is when the current maximum probability density is far less than the initial probability density, in other words, the current input vector includes very little neighborhoods, we stop constructing the clustering center.

The widths of clustering centers are the average distance between clustering centers and training pattern, they are gotten by Eq.9. The linear weight value between hidden layer and output layer is computed by least squares estimate method.

$$\sigma_{C_h} = \sqrt{\frac{1}{R_{C_h}} \sum_{X \in \theta_j} (X - X_{C_h})^T (X - X_{C_h})} \qquad C_h = 1,2,...H. \tag{9}$$

where R_{C_h} is the pattern number of input training vectors θ_{C_h} who cluster around clustering center X_{C_h} .

4 Experimental Results and Conclusions

The performance of our algorithm is validated in NIST-4 standard fingerprint database [7] to test accuracy and CQUOE-FINGER database to test consistency, respectively. From each database we randomly select 800 images in which 600 images are as training set and 200 images are as test set. The output error of RBFNNs is restricted less to 0.01 and dimensions of hidden layer are determined as 12, 8 by the method mentioned in Section 3. In order to improve the robustness of RBFNNs, we train the output layer by three different types of input vectors. The first type vectors are clustering center vectors to insure their targets, the second are vectors around their clustering centers with some noise and the third are the clustering center vectors again. The classification results are shown in Table 1.

Table 1. Confusion matrix for our algorithm(at 0% rejecting rate)

True class	NIST-4					CQUOE-FINGER				
	Assigned class					Assigned class				
	W	R	L	A	T	W	R	L	A	T
W	34	3	1	2	1	38	3	2	0	1
R	1	35	0	2	1	1	45	0	1	0
L	2	1	2	39	1	2	1	57	2	1
A	1	0	41	1	1	1	0	1	22	1
T	0	1	2	3	25	0	0	2	1	18
Right rate (%)	89.47	87.5	89.13	82.98	86.2	90.47	91.84	91.94	84.6	85.71

It can be found from Table 1 that:

(i) The sampled curvatures of W, L, R have clear local peaks (see Fig.2), RBFNNs gets clear and effective response for them, so the average right rate is 88.7% in NIST-4 and 91.42% in CQUOE-FINGER at 0% rejecting rate, higher than other classes for 4.11% and 6.27%, respectively.

(ii) The right rates for A, T are 82.98%, 86.2% at 0 rejecting rate, better than which in Ref. [1],[2],[4].

The main causes of false classification lie on serious deviation of directions by blur, too wet or too dry fingertip. Some results in Table 2 shows that the performance is improved to 91.79% at 20% rejecting rate and gets practical value in civil applications. If the number of input vector training is added, the right rate could be improved more.

Table 2. Classification results of our algorithm at different rejecting rate

Rejecting Rate (%)	0	10	20
W (%)	89.47	89.89	93.51
R (%)	87.5	88.23	92.14
L (%)	89.13	90.12	94.31
A (%)	82.98	86.37	89.26
T (%)	86.2	87.3	86.75
Average (%)	87.06	88.38	91.79

It can be concluded that our algorithm doesn't only show good accuracy, stability and consistency in the two databases, but have good balance ability. In some sense, the best performance of structure classifier is related to the discrimination between A and W and flat classifier is related to L, R, T[2],[4], hence, the sampled curvatures can represent structure information very well and their distributions can represent flat information effectively after treatment by RBFNNs with local approximate ability, so the combination of them provides a promising approach for fingerprint classification.

References

1. Karu, K.,Jain,A.K.:Fingerprint Classification. Pattern Recognition, **29** (1996) 384-404
2. Marcialis, G L,Goli, F,Fransconi P.: Fingerprint Classification by Combination of Flat and Structural approaches. In: AVBPA (2001) 241-246
3. Jain, A.K,Prabhakar, S, Lin, H.: A Multichannel Approach to Fingerprint Classification. IEEE Trans. PAMI, **21** (1999) 348-359
4. Yuan, Y, Gian,Luca, M.,Massimiliano, P.,et al.: Combining Flat and Structured Representations for Fingerprint Classification with Recursive Neural Networks and Support Vector Machines. Pattern Recognition, **36** (2003) 397-406
5. Candela, G T,Grother, P J,Watson, C I,et al.: A Pattern-level Classification Automation System for Fingerprints. NIST Tech. Report NISTIR, 5647 (1995) 1-5
6. Cappilli, R,Maio, D,Maltioni, D.: Fingerprint Classification Based on Multispace KL. In: Proceedings of Workshop on Automatic Identification on Advances Technologies, Nanjing (1999) 117-120
7. Watson, C.I., Wilson, C.L.:Fingerprint Database. National Institute of Standards and Technology, Special Database 4, FPDB April (1992)
8. Chen, S, Cowan, C F N, Grant, P M. :Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks. IEEE Trans. Neural Networks, **2** (1991) 302-309
9. Ruan, X.: An Organizing Radias Basis Function Network and Its Hybrid Learning Algorithm. Journal of Beijing Polytechnic University, **25** (1999) 31-37

Palmprint Recognition Based on Translation Invariant Zernike Moments and Modular Neural Network

Yanlai Li[1], Kuanquan Wang[1], and David Zhang[2]

[1] Biometrics Research Center, Dept. of Computer Science and Engineering
Harbin Institute of Technology, Harbin, Heilongjiang 150001, China
{yanlai_l,wangkq}@hit.edu.cn
[2] Department of Computing, Hong Kong Polytechnic University, Hong Kong
csdzhang@comp.polyu.edu.hk

Abstract. This paper introduces a new approach, TIZMs & MNN, for palmprint recognition. It uses translation invariant Zernike moments (TIZMs) as palm features, and a modular neural network (MNN) as classifier. Translation invariance is added to the general Zernike moments which have very good property of rotation invariance. A fast algorithm for computing the TIZMs is adopted to improve the computation speed. The pattern set is set up by eight-order TIZMs. Because palmprint recognition is a large-scale multi-class task, it is quite difficult for a single multilayer perceptrons to be competent. A modular neural network is presented to act the classifier, which can decompose the palmprint recognition task into a series of smaller and simpler two-class sub-problems. Simulations have been done on the Polyu_PalmprintDB database. Experimental results demonstrate that higher identification rate and recognition rate are achieved by the proposed method in contrast with the straight-line segments (SLS) based method [2].

1 Introduction

Recently, computer-aided personal recognition and identification technology with biometrics feature, including iris, fingerprint, face, voice and so on, has become more and more important because of its stability and reliability. So far the most widely used feature is fingerprint while the most reliable one is the iris [1]. As a new kind of biometric feature, palmprint has several advantages compared with other ones: low-resolution imaging, low cost capture device, non-fake, stable line feature and easy self-positioning, etc. It is for these reasons that palmprint identification and recognition draws more and more researchers' attention in the last few years [2],[3],[4]. Zhang et al. [2] proposed to use a palmprint as a biometric feature for identity recognition and obtained good results in offline palmprint verification. Duta et al. [3] extracted some feature points lying along the palm-lines and verified the identity by matching these points. Zhang [4] used 2-D Gabor filters to extract the texture features from low-resolution palmprint images captured using a CCD camera and employed these features to implement a highly accurate online palmprint recognition system.

Moments that have the invariant properties have received considerable attention in recent years [7]. Zernike moments (ZMs) possess a useful rotation invariance prop-

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 177–182, 2005.

erty. Besides, they have less information redundancy and high information efficiency due to their orthogonality. In this paper, we add the translation invariance to Zernike moments and use them as the palm features. After the features are extracted, they are input to a designed classification rule to decide a labeling for the underlying image. With no doubt, palmprint recognition is a large-scale multi-class problem. Therefore, it is difficult for a single multilayer perceptrons (MLP) trained by the standard back-propagation algorithm to be competent for this task. Modular neural networks have attracted more attention for task decomposition applications recently [8],[9],[10]. In this paper, a new modular neural network structure with fast training algorithm is proposed as the classifier.

2 Brief Description of Zernike Moments (ZMs)

This section introduces the theories of Zernike moments. The radial moments of order p with repetition q are defined as

$$D_{pq} = \int_{\theta=0}^{2\pi} \int_{r=0}^{\infty} r^p e^{-jq\theta} f(r,\theta) r \, dr \, d\theta, \quad j = \sqrt{-1} \,, \tag{1}$$

where $p = 1,2,\cdots,\infty$ and q takes on any positive or negative integer values.

The kernel of Zernike moments is a set of orthogonal Zernike polynomials defined over the polar coordinate space inside a unit circle. The two- dimensional Zernike moments of order p with repetition q of an image intensity function $f(r,6)$ are defined as

$$Z_{pq} = \frac{p+1}{\pi} \int_{\theta=0}^{2\pi} \int_{r=0}^{1} V_{pq}^{*}(r,\theta) f(r,\theta) r \, dr \, d\theta \,, \quad |r| \le 1 \,, \tag{2}$$

where polynomials $\{V_{pq}(r,\theta)\}$are defined as

$$V_{pq}(r,\theta) = R_{pq}(r) e^{jq\theta} \,, \tag{3}$$

and the real valued radial polynomial $R_{pq}(r)$ is given as

$$R_{pq}(r) = \sum_{k=0}^{(p-|q|)/2} (-1)^k \frac{(p-k)!}{k!((p+|q|)/2-k)!((p-|q|)/2-k)!} r^{p-2k} \,, \tag{4}$$

where $0 \le |q| \le p$ and $p-|q|$ is even. The relationship between Zernike and radial moments is given by

$$Z_{pq} = \lambda_p \sum_{k=q}^{p} B_{pqk} D_{kq} \,, \qquad \lambda_p = \frac{p+1}{\pi} \,, \tag{5}$$

where the polynomial coefficient B_{pqk} is defined as

$$B_{pqk} = \frac{(-1)^{(p-k)/2} ((p+k)/2)!}{((p-k)/2)!((p+k)/2)!((k-q)/2)!} \,. \tag{6}$$

For digital images, the discrete approximation to their respective continuous moments is

$$D_{pq} = \lambda_d(N)\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} r_{ij}^{p} e^{-jq\theta_{ij}} f(i,j)\,, \tag{7}$$

$$\text{and}\, Z_{pq} = \lambda_z(p,N)\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} R_{pq}(r_{ij}) e^{-jq\theta_{ij}} f(i,j)\,, \tag{8}$$

where the image coordinate transformation to the interior of the unit circle is given by

$$r_{ij} = \sqrt{x_i^2 + y_j^2}\,, \quad \theta_{ij} = \tan^{-1}\left(\frac{y_j}{x_i}\right), \quad x_i = c_1 i + c_2\,, \quad y_j = c_2 j + c_1\,,$$

$$\lambda_d(N) = \frac{2}{(N-1)^2}\,, \quad \lambda_Z(p,N) = \frac{2(p+1)}{\pi(N-1)^2}\,, \quad c_1 = \frac{\sqrt{2}}{N-1}\,, \quad c_2 = -\frac{1}{\sqrt{2}}\,.$$

When calculating the Zernike moment of different orders, we choose a fast algorithm named q-recursive method [11] to save the computation time.

3 Translation Invariant Zernike Moments (TIZMS)

Zernike moments are powerful feature descriptors from which rotational invariance be easily constructed. But the general ZMs have not translation invariance. The main changes of our captured images are translation and rotation, so we consider adding the translation invariance to the ZMs through some transformations. TIZMs can be achieved by the radial central moments [6]

$$\hat{Z}_{pq} = \lambda_p \sum_{k=q}^{p} B_{pqk}\hat{D}_{pq} = \lambda_p \sum_{k=q}^{p} B_{pqk} \sum_{m=0}^{u'}\sum_{n=0}^{v'} \binom{u'}{m}\binom{v'}{n}(-A)^m(-A^*)^n D_{p''q''}\,,$$

where $p'' = (u'-m)+(v'-n)\,, \quad q'' = (u'-m)-(v'-n)\,, \quad u' = (k+q)/2\,, \quad v' = (k-q)/2\,.$ (9)

Eq. (5) can be rewritten into

$$D_{pq} = \sum_{s=0}^{a} \gamma_{(q+s)}\delta_{p,q,s} Z_{(q+s)q}\,, \tag{10}$$

where $\gamma_{q+s} = \dfrac{1}{\lambda_{q+s}B_{(q+s)q(q+s)}}\,, \quad C_{p(q+s)} = \dfrac{B_{bq(q+s)}}{B_{pqp}}\,, \quad a = p-q\,,\ a-s$ is even, and

$$\delta_{p,q,s} = \begin{cases} 1 & if\ p = q+s \\ -C_{p(q+s)} + \sum_{b=2}^{a-s-2} -C_{p(q+s+b)}\cdot\delta_{q+s+b,q,s} & else \end{cases}.$$

It has been proven that the new moments $\hat{Z}_{pq}$ preserve not only the rotation invariance, but also the translation invariance [6].

4 Modular Neural Network (MNN) and Its Training Algorithm

In this paper, we provide a modular neural network (MNN) as the classifier, which can decompose an N-class classification problem into a series of $N*(N-1)/2$ two-class problems. These two-class problems are to discriminate class C_i from class C_j, for $i=1,2,\cdots,N$, and $j=i+1$, while the existence of the training data belonging to the other N-2 classes is ignored. The architecture of the MNN is illustrated in Fig.1. The structures of all the sub-modules are chosen to be three-layered MLP. In order to accelerate the convergence, we choose the fast algorithm named SATA [12] to train each sub-module.

5 Experimental Results

Our experiments are carried out on the Polyu_palmprintDB database, which is composed of 3200 palmprint images from 320 different palms, each has ten images. All the images share a dimension of 571*800 pixels. After pre-processing such as smoothness and segmentation, the central part of each image is segmented with 128*128 pixels. We select seven random images of each palm from the database to set up the training set, while the remained three are used to be tested. Firstly, feature vectors of TIZMs with eight orders are extracted using the method introduced in section 2 and 3. Then the training and testing dataset are set up respectively. Before being input to the MNN, the problem is decomposed into 51040 (320*319/2) of two-class problems. Each two-class problem is solved by a sub-module, which is a three-layered MLP with 5 hidden units. After training, the parameters of all the sub-modules are saved. Secondly, the patterns to be tested are classified according to the saved parameters and the integrated principle. The Experimental results are listed in Table. 1. For comparison, the straight-line segments (SLS) based method [2] is also implemented in our experiments. The Receiver Operating Characteristic (ROC) curves of TIZMs & MNN approach and the SLS method are shown in Fig. 2. Their EER is 0.85% and 4.5%, respectively. From Fig. 2 and Table.1 we can see that the proposed TIZMs & MNN approach is more efficient than the SLS method in terms of identification rate and recognition rate for palmprint recognition problem.

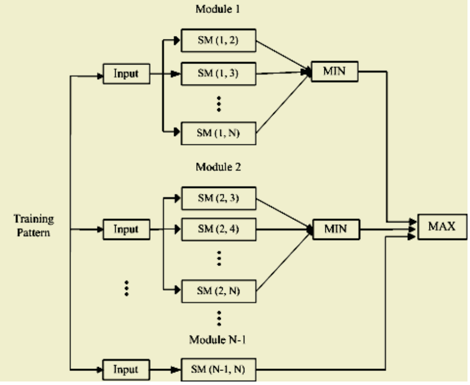


Fig. 1. The architecture of the modular neural network, where “SM” denotes “Sub-Module”

Table 1. Comparison results of the TIZMs & MNN approach and the SLS method in [2]

Method	TIZMS & MNN	SLS Method in [2]
Database Size	3,200 images (320 palms)	3,200 images (320 palms)
Identification Rate	99.5%	95.9%
Recognition Rate	98.7%	92.3%

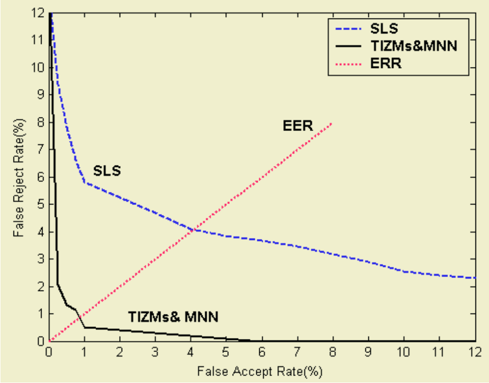


Fig. 2. Comparison ROC curves of TIZMs & MNN approach and SLS method

6 Conclusions

Pamlprint recognition is a new biometric technique for person identification and recognition. This paper presents an efficient method for palmprint recognition using the Translation Invariant Zernike moments (TIZMs) as features, and a modular neural network (MNN) as classifier. The eight orders of the TIZMs are calculated by fast algorithm and then set up the pattern set. 99.5% and 98.7% accurate rates are obtained for identification rate (one-to-one matching test) and recognition rate (one-to-many matching test), respectively. These results are much better than the SLS method proposed in [2]. Simulation results have demonstrated that the presented TIZMs & MNN approach is very efficient for palmprint recognition problem.

References

1. Zhang, D.: Automated Biometrics-Technologies and Systems. Kluwer Academic Publishers (2000)
2. Zhang, D., Wei Shu: Two Novel Characteristics in Palmprint Verification: Datum Point Invariance and Line Feature Matching. Pattern Recognition, **32** (1999) 691-702
3. Duta, N., Jain, A., Mardia, K.V.: Matching of Palmprint. Pattern Recognition Letters, **23** (2001) 477- 485
4. Zhang, D., Kong, W., You, J., Wong, M.: Online Palmprint Identification. IEEE Transactions on Pattern Analysis and Machine Intelligence, **25** (2003) 1041-1050
5. Fukumi, M.; Omatu, S.; Nishikawa, Y.: Rotation-invariant Neural Pattern Recognition System Estimating a Rotation Angle. IEEE Trans. Neural Networks, **8** (1997) 568 -581

6. Cheeway Chong, Raveendran, P., Mukundan, R.: Translation Invariants of Zernike Moments, Pattern Recognition, **36** (2003) 1765-1773
7. Raveendran, P., Sigeru Omatu, Ong, S.H.: Neuro-pattern Classification Using Zernike Moments and Its Reduced Set of Features. Intelligent Systems Engineering, winter (1994) 230-235
8. J. M., Murre, J.: Learning and Categorization in Modular Neural Networks. London, U.K. Harvester Wheatsheaf (1992)
9. Waibel, Sawa i, H., Shkano, K.: Modularity and Scalling in Large Phonemic Neural Networks. IEEE Trans. Acoust., Speech, Signal Processing, **37** (1989)
10. Lu, L., Kita, H., Nishikawa, Y.: A Multisieving Neural-network Architecture That Decomposes Learning Tasks Automatically. in proc. IEEE Conf. Neural Networks, Orlando, FL (1994) 1319-1324
11. Chong, C., Raveendran, P., Mukundan, R.: A Comparative Analysis of Algorithms for Fast Computation of Zernike Moments. Pattern Recognition, **36** (2003) 731-742
12. Li, Y., Wang, K., Zhang, D.: Step Acceleration Based Training Algorithm for Feedforward Neural Networks. Pattern Recognition, 2002. Proceedings. 16th International Conference on, **2** (2002) 84-87

Gait Recognition Using Independent Component Analysis

Jiwen Lu, Erhu Zhang, Zhigang Zhang, and Yanxue Xue

Department of Information Science, Xi'an University of Technology
Xi'an, Shaanxi 710048, China

Abstract. This paper presents a new method for automatic gait recognition using independent component analysis (ICA). Firstly, a simple background subtraction algorithm is introduced to segment the moving figures accurately and to achieve binary silhouettes. Secondly, these 2D binary silhouettes are converted into associated sequences of 1D signals and ICA is applied to get the independent components of each 2D binary silhouettes. For the sake of reducing computation cost, a fast and robust fixed-point algorithm named FastICA is adopted. A criterion that not all ICs are useful for recognition is demonstrated and a method of IC selection is put forward. Lastly, the nearest neighbor (NN) classifier for recognition is chosen. This algorithm is tested on small MUD gait database and the NLPR gait database and experimental results show that our method has encouraging recognition accuracy.

1 Introduction

The need for automatic human identification, especially at a great distance, has recently gained great interest from the computer vision researchers for it is widely used in security-sensitive environments such as banks, parks and airports. Biometrics is a powerful tool for reliable human identification and it makes use of human physiology or behavioral characteristics such as face, iris and fingerprints for identification. As a new behavioral biometric, gait recognition aims at identifying people by the way they walk. Compared with the first generational biometrics such as fingerprints and iris, gait is also affected by some factors such as drunkenness, pregnancy and injuries involving joints, but it still has the advantages of being non-contact, non-invasive, unobvious, and is the only perceivable biometric feature for human identification at a great distance. Unlike face, gait is also difficult to conceal and has great potential applications in many scenes especially for human identification at great distance.

Although gait recognition is a new research field, there have been some attempts in the recent literatures [1-5]. Currently, gait recognition approaches may be classified two main classes, namely model-based methods [5] and motion-based methods [2], [3]. Model-based methods aim to model human body and perform model matching in each frame of a walking sequence so that the parameters are measured. Since the effectiveness of model-based techniques, especially in human body modeling and parameter recovery from a walking sequence, is still limited (e.g. tracking and locating human body accurately in 2D or 3D space has been a long-term challenging and unsolved problem though there are much progresses in the past years), most existing gait recognition methods are motion-based. Hence, like other motion-based algo-

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 183–188, 2005.

rithms, we may consider gait to be composed a sequence of body poses and recognize it by the similarity of these body poses and silhouettes. Based on this assumption, this paper proposes an automatic gait recognition using independent analysis, which achieves high recognition accuracy results. The method proposed in the paper can be mainly divided into three procedures including human motion detection, gait representation and recognition. The main advantages of the approach in this paper are as bellow: (1) based on ICA, we make a meaningful attempt to human identification through gait information. (2) One-dimensional boundary unwrapped distance signals are applied to represent the changing moving silhouettes which can decrease computational cost effectively. (3) It is easy to implement and has better recognition accuracy.

2 Gait Feature Extraction Based on ICA

Before training and recognition, each sequence involving a walking figure is converted into a sequence of signals at this preprocessing stage. This procedure involves segmentation of human motion and extraction of distance signals.

2.1 Segmentation of Human Motion

Human segmentation is the first step and plays a key role in the whole recognition system. To extract the silhouettes of walking figures from the background, a simple motion detection method using median value is adopted to construct the background image from a small portion of video sequence including moving objects. Let P represents a sequence including N frames. The resulting background $p(x, y)$ can be computed as formulas (1):

$$p(x, y) = median[p_1(x, y), p_2(x, y), \cdots, p_N(x, y)] \cdot \tag{1}$$

The value of $p(x, y)$ is the background brightness to be computed in the location of pixel (x, y) and $median$ represents its median value. Here median value is taken rather than mean value of the pixel intensities over N frames, because mean value will be distorted by the large change in pixel intensities when the person moves past that pixel while the median is unaffected by spurious values, and computing median value is also faster than the least median square value [2]. The assumption made in this process is that the person does not stand still over the frames which are analyzed as in that case the background extraction will classify the person as a part of background. It should be noted that there is not a perfect image segmentation algorithm to segment the sequence images up to the present, here we adopt traditional histogram method to segment the foreground, therefore there must exist noises in the foreground, so morphological operators such as erosion and dilation are employed to further filter spurious pixels. Two examples of background subtraction can be seen in Fig.1 from (a) to (h). The MUD database and NLPR database can be obtained by [4] and [1] separately, which have been used as the data set for many other previous gait experiments.

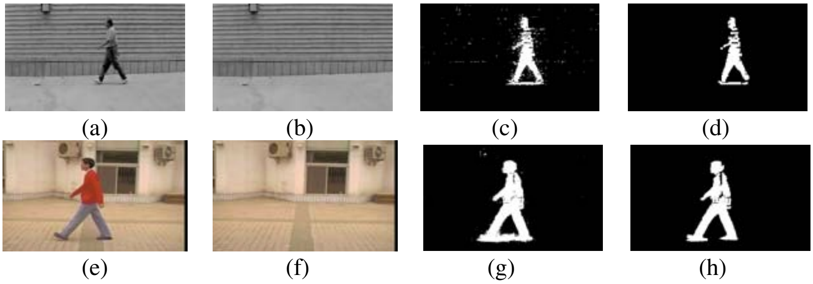

(a) (b) (c) (d)
(e) (f) (g) (h)

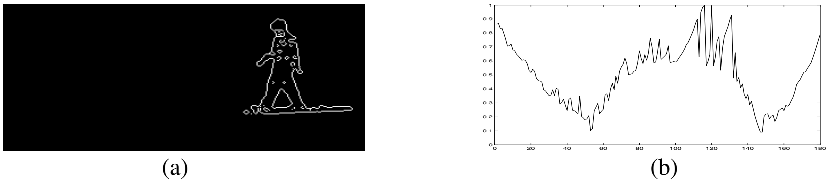
Fig. 1. Examples of background extraction from a sequence. (a), (b) (c) and (d) are from UMD database [4], (e), (f), (g) and (h) are from NLPR database [1]. (a) and (e) are original images, (b) and (f) are the background images extracted through median method, (c) and (g) are the difference images, (d) and (h) are the silhouettes after morphological processing

2.2 Representation of Human Silhouette

An important factor affecting gait recognition is how to represent the silhouettes and extract the feature. To let this method be insensitive to changes of color and texture of clothes, we only use the binary silhouette. For the sake of improving computational efficiency, we convert these 2D silhouettes into 1D signals. First, for each binary image sequence, we can compute the centroid (x_c, y_c) of each frame image's contour shape and regard this centroid as the reference point, then unwrap the counter counterclockwise to turn it into a distance signal. If there are M points in its contour, the distance signal can be computed by formulas (2):

$$d_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \;\; (i = 1,2,\cdots,M) \; . \tag{2}$$

Given n classes for training and each class represents a sequence of one subject's gait. Let $D_{i,j}$ be the jth distance signals in class i and N_i is the number of such distance signals in the ith class. The total numbers of training sequences is $N_i = N_1 + N_2 + \cdots + N_s$ and the whole training set can be represents into $D = [D_{1,1}, D_{1,2}, \cdots, D_{1,N_1}, D_{2,1}, \cdots, D_{i,N_s}]$. Considering there may be different number points in different person's silhouette, we normalize the distance signals into a fixed length and resample it (here 180 points in our experiments). One example of normalized distance can be seen in Fig.2 from (a) to (b).



(a) (b)

Fig. 2. (a) is the counter of the silhouette and (b) is the normalized distance (180 points in total)

2.3 Training and Feature Extraction

At this stage, we will extract and train gait feature using ICA. The concept of ICA can be seen as a generational of principal component analysis (PCA) and its basic idea is to represent a set of random variables using basis functions, where the components are statistically independent or as independent as possible [6].

Let us denote the observed variables x_i as a vector with zero-mean random variable $X = (x_1, x_2, \cdots, x_m)^T$, the component variables s_i as a vector $S = (s_1, s_2, \cdots, s_n)^T$ with the model AS

$$X = AS \ . \tag{3}$$

Where A is unknown $m \times n$ matrix of full rank, called the mixing or feature matrix. The columns of A represent gait features and s_i signals the amplitude of the ith feature in the observed data x. For reducing computational cost, an algorithm named FastICA [7] using a fix-point iteration algorithm finding the local extrema of the kurtosis of a linear combination of the observed variables is introduced. Applying FatstICA on gait extraction, the random variables will be the training normalized distance of gait images. We select thirty contour images for each class to construct the matrix X and make use of the fixed-point algorithm to calculate matrices A and S. Let $x_i^{'}$ be a distance vector of one contour image, we can construct a training distance set $\{x_1^{'}, x_2^{'}, \cdots, x_n^{'}\}$ with m random variables which are assumed to be linear component of n unknown ICs, denoted by $s_1^{'}, s_2^{'}, \cdots, s_n^{'}$. The relationship between X and S can be modeled as $X = AS$. For this relationship, each distance vector $x_i^{'}$ can be represented by a linear combination of $s_1, s_2, \cdots, s_n$ with weighting $a_{i1}, a_{i2}, \cdots, a_{in}$ .Therefore, the feature matrix A can be considered as the features of all the training images.

According to the ICA theory, the matrix S contains all the independent components, which are calculated from a set of training distances. The matrix AS can reconstruct the original signals X .To reduce the computation cost, we select some ICs from A in the way that the ratio of the within-class scatter and between-class scatter is minimized [8]. The method is proposed as follows.

If the matrix X contains n individual persons and each person has m frames images, a_{ij} represents the entry at the ith row and the jth column. The value W_j, which is called as the mean of within-class distance in the jth column, is then given by

$$W_j = \frac{1}{nm(m-1)} \sum_{i=1}^{n} \sum_{u=1}^{m} \sum_{v=1}^{m} (a_{(i-1)m+u,j} - a_{(i-1)m+v,j})^2 \ . \tag{4}$$

The value B_j, which is called as the mean of between-class distance in the jth column:

$$B_j = \frac{1}{n(m-1)} \sum_{s=1}^{n} \sum_{t=1}^{n} \rho(a_{s,j}^{'} - a_{t,j}^{'}) \ . \tag{5}$$

where

$$a'_{i,j} = \frac{1}{m}\sum_{u=1}^{m} a_{(i-1)m+u,j} \quad . \tag{6}$$

In this paper, we employ the ratio of within-class distance and between-class distance to select stable mixing feature from A .The ratio γ_j is defined as

$$\gamma_j = \frac{W_j}{B_j} \quad . \tag{7}$$

From the definition γ_j , the smaller γ_j is, the better the classifier will be. Using (7), we choose the smallest γ_j and select the top k ($k < n$) column features from A and S .

3 Recognition and Experimental Results

Gait recognition is a traditional pattern classification problem which can be solved by measuring similarities between the training database and the test database. The classification process is carried out through the simple different methods, namely the nearest neighbor classifier (NN) derived from the ICs. Two databases, namely University of Maryland (MUD) and Chinese National Laboratory of Pattern Recognition (NLPR), are chosen to evaluate the capability of the proposed method. Here MUD database adopted is compose of six person and one sequence for each person (it just a part of the large UMD gait database), NLPR database includes 20 subjects and four sequences for each view angle and has three angles, namely laterally (0°), obliquely (45°) and frontally (90°).Table 1 gives the final experimental results on the two datasets as follows.

Table 1. The recognition results of feature selection tested on MUD and NLPR datasets

Recognition results	Results on MUD datasets		Results on NLPR datasets	
IC Number selected	90 ICs	All ICs	300 ICs	All ICs
Rank 1	100%	100%	75.0%	75.0%
Rank 5	100%	100%	85.0%	85.0%
Rank 10	100%	100%	95.0%	95.0%

4 Conclusions and Future Work

This paper has performed a simple gait recognition method based on human silhouette using independent components analysis. From the analysis, we have found the independent components which are transformed from the normalized distances have much better discriminatory capability than other gait feature. Besides these, the median background extraction method is better than the mean method and has less computation cost than the least mean square method. To provide a general approach to automatic human identification based on gait in real environments, much remains to be done in the future. Although our recognition accuracy is high, we still cannot conclude much about gaits. Further evaluation on a much larger and most varied database is still needed. We are planning to build up such a gait database with more subjects, more sequences and more variation in conditions. The lack of general gait database, especially more viewing angles of gait database, is another limitation to most current gait recognition algorithms. Our proposed method is just recognizing human through one view, i.e. perpendicularity with the direction of human walking, so a useful experiment should be to determine the sensitivity of the features from different views which can provide a more conviction results. Another method of solving this problem is to store more training sequences taken from multiple views and classify them. At last, seeking better maturity measures, designing more sophisticated classifiers, better gait detection and segmentation algorithms deserves more attention in future work.

Acknowledgements

We would like to express our thanks to The Institute of Automatic, Chinese Academic of Science (CASIA) for Human ID image database. Portion of the research in this paper uses the CASIA Gait database collected by Institute of Automatic, Chinese Academic of Science. This work is partly supported by the Foundation of Xi'an University of Technology (No. 104-210401, 104-210413).

References

1. http://sinobiometrics.com/
2. Wang, L., Tan, T., Ning, H., Hu, W.: Silhouette Analysis-Based Gait Recognition for Human Identification. IEEE Transactions on Pattern Analysis and Machine Intelligence, **25** (2003) 1505-1518
3. Foster, Jeff P., Mark S. Nixon, Adam Prugel-Bennett.: Automatic Gait Recognition Using Area-based Metrics. Pattern Recognition Letters, **24** (2003) 2489-2497
4. http://www.doc.ic.ac.uk/~mrd98/gait
5. Wagg, D.K., Nixon, M.S.: An Automated Model-based Extraction and Analysis of Gait. IEEE Conference on Automatic Face and Gesture Recognition, Seoul, Korea, **5** (2004) 11-16
6. Hyvarinen, A., Oja, E.: Independent Component Analysis: Algorithm and Applications. Neural Networks, **13** (2000) 411-430
7. Hyvarinen, A.: A Fast and Robust Fixed-Point Algorithm for Independent Component Analysis. IEEE Transaction on Neural Networks, **10** (1999) 626-634
8. Yuen, P.C., Lai, J.H.: Face Representation Using Independent Component Analysis. Pattern Recognition, **35** (2002) 1247-1257

Nighttime Pedestrian Detection with a Normal Camera Using SVM Classifier

Qiming Tian, Hui Sun, Yupin Luo, and Dongcheng Hu

Department of Automation, Tsinghua University, Beijing 100084 China
tianqm@mail.au.tsinghua.edu.cn

Abstract. Accidents occurring at night and involving pedestrians represent a significant percentage of the total. This paper presents an approach for pedestrian detection in nighttime with a normal camera using a SVM classifier. Objects in the video are extracted with an adaptive threshold segmentation method at first. In the recognition phase, a preliminary classifier is used to discard most candidates and a SVM classifier is used in detailed shape analyzing. At last, a tracking module is used to verify the classification result. This approach is more cost-efficient than the previous approaches which are based on expensive infrared cameras. Experimental results show that the proposed approach can detect 71.26% pedestrians and run in real-time.

1 Introduction

Pedestrian accidents represent the second largest source of traffic-related injuries and fatalities, and about 62% of them occur at night. The pedestrian safety has attracted more and more attentions and several driver assistance systems for pedestrian detection have been developed in the last few years.

There are four kinds of sensors used by these systems: normal cameras, infrared cameras, laser scanners and radars. Normal cameras are often used in daytime systems ([1],[4],[5],[9],[10]). They have high resolution and are cost efficient, but they are sensitive to the light conditions. In contrast, Infrared cameras are often used in nighttime systems ([2],[3],[6],[7],[8]). They are less sensitive than the normal cameras, but their resolution is very low and they are very expensive. A detailed comparison between these two kinds of cameras in pedestrian detection can be found in [7]. Radars and laser scanners are active sensors which can operate in different environment, but it is very difficult for them to distinguish pedestrians from other objects, so that they are often used with a camera.

Most of the approaches used in the vision-based systems can be divided into two steps: selection step and recognition step. Objects in the video are separated in the selection step, and then these objects are classified with various pattern recognition technologies in the recognition step according to their size, shape and some other features.

Many methods such as image segmentation ([2],[3]), stereovision ([1],[4], [5],[10]) and edge extraction [6] are used in the selection step. In the recognition step, there are several features used for classification. The first feature

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 189–194, 2005.

is the size and the location of the object. Most approaches take the size and position constraints in the image coordinates for a direct preliminary classification. The second feature is the shape of the object. It is represented by the raw intensity image [2], the edges image [1], Harr wavelet transfer [8], intensity gradient image ([4],[9]). In classification, [3] and [10] use template matching with a single template or several templates. [6] uses a 3D model for shape matching. [6] exploit the vertical symmetry of the shape. [7] utilizes the intensity histogram. Other approaches use learning-based methods which need an offline train phase. [1],[2],[8],[9] use Support Vector Machine, while [4],[5],[10] use Neural Network.

The approach presented in this paper aims to detect pedestrians in nighttime. The main feature is that this approach is more cost efficient, because it only needs a normal camera while others use expensive infrared cameras. To our knowledge, it is the first nighttime pedestrian detection system that uses only a normal camera.

2 System Description

2.1 System Overview

Infrared cameras are always used in the previous nighttime pedestrian systems. Pedestrians appear as bright bulbs in infrared video because of the heat emitted from their body. It helps to establish a very robust object detection algorithm with a simple global intensity threshold algorithm ([2],[3]). But in winter, pedestrians often wear heavy clothes, so only their heads can be seen in infrared videos ([2],[6],[7]). It is difficult to recognize a pedestrian only by the shape of his head [2]. While in summer, the temperature of pedestrians' body is close to the environment temperature. It is difficult to separate them from the background [6]. In addition, the quality of most infrared videos is worse than that captured by a normal camera [7], especially when an un-cooled infrared camera is used [3]. It would limit the detection range of infrared systems. The most significant shortage of an infrared camera is that it is too expensive to be used in a cost efficient system. In order to develop a cost efficient system, a normal camera which operates in visible spectrum is used in the presented system instead. A sample of the nighttime video signal is shown in Figure 1.

A two-step approach is used in this system to detect pedestrians in the video as that in other systems. In the selection step, objects are separated with an adaptive threshold image segmentation algorithm which is discussed in section 2.2. In the recognition step, size, position and shape feature of the objects are used in classifying. There are three modules in this step. In section 2.2, the first module, which is a preliminary classifier and can discards most of the candidate objects, is discussed. The second module is a detailed classifier which is based on the shape feature. Because of the variations of pedestrian's shape, a SVM classifier is utilized. This module is discussed in section 2.3. The last module is a tracking module which can reduce false alarm rate based on the classification history of several recent frames. This module is discussed in section 2.3

2.2 Image Segmentation and the Preliminary Classifier

In the presented system, an adaptive threshold algorithm is used. For a foreground pixel, its intensity should be larger than a global threshold denoted as C_1 and a local threshold. The local threshold is set as C_2 times the mean intensity of the pixel's neighborhood window whose size is $(2C_3 + 1)(2C_4 + 1)$. The threshold for the pixel (i, j) is given by equation (1):

$$threshold(i, j) = \min(C_1, C_2 * mean(i, j)) \tag{1}$$

The $mean(i, j)$ in (1) represent the mean intensity of the pixels in the image window. C_1, C_2, C_3, C_4 should be optimized in experiments. C_1 represents the lowest intensity of a pedestrian and it is set as 90 which is small than 97% of the pixel intensities of pedestrians in the sample videos. C_2 is used to reduce the segmentation noise and it is set as 1.1 in the system implement. The image window is set about as large as the region of a pedestrian who stands at about 70m away from the vehicle. According to the experiments, the optimized value for C_3 and C_4 are 15. After the threshold, foreground regions are extracted by connecting the foreground pixels. The result of this step is shown in Figure 2.

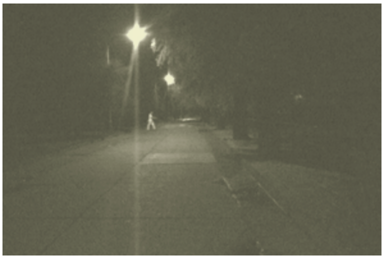


Fig. 1. The video signal

Fig. 2. The result of segmentation

These regions are filtered by a preliminary classifier at first for saving computation. Four constraints are used in the preliminary classifier: object size and object position in the image coordinates, object width/height ration and a simple shape constraint. Regions whose areas are larger than 10000 pixels or smaller than 50 pixels are discarded. Because when a pedestrian stands at 20∼80m away from the vehicle, the bottom of the region will be at the middle of the image. Regions whose bottoms are beyond the range of [0.3h,0.7h](h is the height of the frame) are discarded. The range of height/width ratio of candidate region is set as [1, 3] because the range of the ratio of a pedestrian is in [1, 3] considering accessories and poses. The last constraint is a simple shape constraint which is used to discard some road sign regions. The area ratio of an object's region and its frame is calculated, and the range of this ratio is set as [0.5, 0.95] because some road sign regions occupy only a small portion of their frame. With the preliminary classifier, a lot of objects are eliminated. Result of this step is shown in Figure 2. Only two objects (black regions in Figure 2) are not discarded.

2.3 SVM Classifier and Tracking

To capture the variation of human shape, SVM method [11] is used in the main classifier. The Support Vector Machine (SVM) is used in [1],[2],[8] and [9] for shape recognition. Neural Network ([4],[10]) and Template Matching ([3],[6],[10]) are also used in shape recognition. According to some experiments, Neural Network classifier is worse than the SVM classifier in this application. Template Matching is slower than the SVM classifier because a lot of templates are needed to match pedestrians of different poses.

There are three problems that should be considered in designing the SVM classifier. The first problem is how to preprocess the input image. Some SVM classifiers don't preprocess the raw intensity image [2], while others take binary image [2], edges image [1], Harr wavelet transform image [8] and intensity gradient image [9] as their input. Because edges are sensitive to noise, and Harr wavelet transform is close to intensity gradient, only the intensity image input, binary image input and intensity gradient image input are tested and compared in experiments. All of these three kinds of input have some disadvantages in encoding the shape information: the intensity image is a little redundant; the binary image suffers from the imperfect segmentation method; the intensity gradient image is sensitive to the noise.

The second problem is how large the input image should be. Most of the previous works normalize the candidate image windows into a fixed size for classification. But because candidate image windows vary a lot in their size and resizing may cause the losses of shape information, the efficiency of a fixed input size SVM classifier and several SVM classifiers of different input size are compared in experiments.

The third problem is whether to design several classifiers for different poses. A system of several classifiers for different poses have less support vectors because the inner class distance is smaller, but it may be difficult to classify pedestrians whose pose is not 'standard' to its classifiers. Efficiency of two-pose classifiers and a single classifier are also compared in experiments.

A lot of samples from several videos are collected for experiments, and they are divided into 12 categories according to their size and pose. The pose of a sample is decided by a simple template matching algorithm, and they are divided into two kinds of poses: front pose and side pose. The intensity image, the intensity gradient image and the binary image of a positive side pose sample (the first row), a positive front pose sample (the second row) and a negative sample (the third row) are shown in Figure 3.

For comparison, four classification approaches are designed. Three approaches contain only one SVM classifier and their inputs are intensity image, binary image and intensity gradient image respectively. The input images for these three approaches are normalized into 24×60 at first. The fourth approach contains 12 SVM classifiers, one classifier for each categories, and its inputs are intensity images. About 2000 positive samples and 3000 negative samples are selected to train these classifiers.

The rest samples are used to test these four approaches. The ROC (receiver operating characteristics) curves of them are shown in Figure 4. The results show

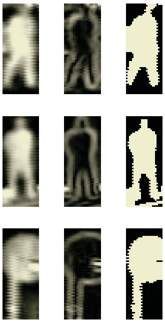


Fig. 3. Region samples

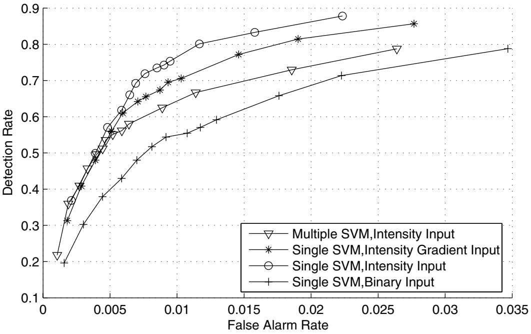


Fig. 4. ROC curve of the four approaches

that the intensity-image-input classifier is more efficient than the binary-image-input classifier and the gradient-image-input classifier. The results also show that the single classifier approach has higher performance than the multiple classifier approach. So the single classifier approach with intensity image input is used in the implement.

After the classification, a tracking module is used to verifying the classification result according to classification history of recent frames.

3 Experimental Results

The overall performance of the presented system is tested on videos captured by a normal camera in nighttime. The overall length of these videos is 44m17s and they have quite a lot of variations in scene, size and pose of people and cluster of background. There are 174 pedestrians in the sample videos and 124 of them are detected successfully. The number of all alarms is 244. So the detection ratio R_1 and the false alarm ratio R_2 are calculated as follow:

$$R_1 = \frac{\text{pedestrians detected}}{\text{all pedestrians}} = \frac{124}{174} = 71.26\% \tag{2}$$

$$R_2 = \frac{\text{false alarms}}{\text{all alarms}} = \frac{244 - 124}{244} = 49.18\% \tag{3}$$

The detection ratio in daytime system [9] is 85%∼96% with the false alarm rate at about 102 times per hour and the detection range is 0m∼30m. The presented system detects pedestrians at 20m∼80m away from the vehicle and the detection ratio is 71.26% with false alarm rate at 163 times per hour. Therefore the presented system is in line with the daytime systems. There are only frames-based results presented in the previous nighttime systems ([2],[7]) and the false alarm rate of them is too high (0.025 times per frame in [2] and 0.0313∼0.1903 times per frame in [7]) to be used in a practical driver assistance systems.

A practical pedestrian detection system should run in real time. In the presented system, the Intel IPP library is used in image processing. On a PC with a P4 2.4G CPU, the average processing time for a frame whose size is 720×480 is 27ms which is better than most of the previous pedestrian detection systems.

4 Conclusion

A new approach for pedestrian detection in nighttime driving is presented in this paper. The video sensor is a normal gray level camera. The approach has two steps. An adaptive threshold method is used in image segmentation in the selection step. In the recognition step, a SVM classifier is used with a preliminary classifier and a tracking module. Experimental results have shown that the method is efficient to detect pedestrians, and it can run at video rate on a P4 2.4G CPU.

References

1. Grubb, G., Zelinsky, A., Nilsson, L., and Rilbe, M.: 3D Vision Sensing for Improved Pedestrian Safety. In Proceedings of IEEE Intelligent Vehicles Symposium (2004)
2. Xu, F., Fujimura, K.: Pedestrian Detection and Tracking with Night Vision. In Proceedings of IEEE Intelligent Vehicles Symposium (2002)
3. Nanda, H., Davis, L.: Probabilistic Template Based Pedestrian Detection in Infrared Videos. In Proceedings of IEEE Intelligent Vehicles Symposium (2002)
4. Zhao, L., and Thorpe, C.: Stereo and Neural Network Based Pedestrian Detection. IEEE Transaction on Intelligent Transport SystemS. **1(3)** (2000) 148–154
5. Wohler, C., Aulanf, J.K., Portner, T., Franke, U.: A Time Delay Neural Network Algorithm for Real-time Pedestrian Recognition. In Proceedings of IEEE Intelligent Vehicles Symposium (1998)
6. Broggi, A., Fascioli, A., Carletti, M., Graf, T., Meinecke, M.: A Multi-resolution Approach for Infrared Vision-based Pedestrian Detection. In Proceedings of IEEE Intelligent Vehicles Symposium (2004)
7. Fang, Y., Yamada, K., Ninomiya, Y., Horn, B., Masaki, I.: Comparison between Infrared-image-based and Visible-image-based Approaches for Pedestrian Detection. In Proceedings of IEEE Intelligent Vehicles Symposium (2003)
8. Bellotti, C., Bellotti, F., De Gloria, A., Andreone, L., Mariani, M.: Developing a Near Infrared Based Night Vision System. In Proceedings of IEEE Intelligent Vehicles Symposium (2004)
9. Shashua, A., Gdalyahu, Y., Hayun, G.: Pedestrian Detection for Driving Assistance Systems: Single-frame Classification and System Level Performance. In Proceedings of IEEE Intelligent Vehicles Symposium (2004)
10. Gravrila, D.M. , Giebel, J., Munder, S.: Vision-based Pedestrian Detection_the PROTECTOR System. In Proceedings of IEEE Intelligent Vehicles Symposium (2004)
11. Vapnik, V.: The Nature of Statistical Learning Theory. Springer Verlag Berlin Heidelberg New York (2000)

Signature Recognition and Verification with Artificial Neural Network Using Moment Invariant Method

Cemil Oz

Sakarya University Computer Engineering Department, Sakarya, Turkey
ozc@umr.edu

Abstract. In this paper, we present off-line signature recognition and verification system which is based on image processing, moment invariant method and ANN. Two separate sequential neural networks are designed; one for signature recognition, and another for verification (i.e. for detecting forgery). Verification network parameters which are produced individually for every signature are controlled by a recognition network. The System overall performs is enough to signature recognition and verification.

1 Introduction

A signature is a special case of handwriting which includes special characters and flourishes. Many signatures can be unreadable. They are a kind of artistic handwriting. However, a signature can be handled as an image, and hence, it can be recognized using computer vision and artificial neural network techniques.

Signature recognition and verification involve two separate but strongly related tasks: one of them is identification of the signature owner, and the other is the decision about whether the signature is genuine or forged. Also, depending on the need, the signature recognition and verification problem is put into two major classes: (i) online signature recognition and verification systems (SRVS) and (ii) offline SRVS. Online SRVS requires some special peripheral units for measuring hand speed and pressure on the human hand when it creates the signature. On the other hand, almost all off-line SRVS systems rely on image processing and feature extraction techniques.

In the last two decades, in parallel with the advancement in sensor technology, some successful online SRVS were developed [1-3]. There are also many studies in the area of offline SRVS category [4-10]. These studies are generally based on ANN [4],[10], analysis of the geometry and topology of the signature [11], and its statistical properties [9].

In this study, we present an off-line signature recognition and verification system which is based on a moment invariant method. Two sequential neural networks are designed: one for signature recognition and another for verification (i.e. for detecting forgery). Verification network parameters are change depending on recognition process results. There is a selection mechanism which determines verification network parameters for recognized signatures.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 195–202, 2005.

2 Image Processing

The scanned real world images containing human signatures are processed using several image processing algorithms before the calculation of the moment invariants. These processes are given below.

Converting color image to gray scale image: In present technology, almost all image capturing and scanning devices use color. Therefore, we also used a color scanning device to scan signature images. A color image consists of a coordinate matrix and three color matrices. The coordinate matrix contains x, y coordinate values of the image. The color matrices are labeled as red (R), green (G), and blue (B). Techniques presented in this study are based on grey scale images, and therefore, scanned or captured color images are initially converted to grey scale using the following equation [12]:

$$\text{Gray color} = 0.299*\text{Red} + 0.5876*\text{Green} + 0.114*\text{Blue}. \tag{1}$$

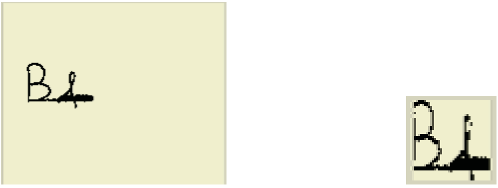
Noise reduction: Noise reduction (also called "smoothing" or "noise filtering") is one of the most important processes in image processing. Images are often corrupted due to positive and negative impulses stemming from decoding errors or noisy channels. An image may also be degraded because of the undesirable effects due to illumination and other objects in the environment. A median filter is widely used for smoothing and restoring images corrupted by noise. It is a non-linear process useful especially in reducing impulsive or salt-and-pepper type noise. In a median filter, a window slides over the image, and for each positioning of the window, the median intensity of the pixels inside it determines the intensity of the pixel located in the middle of the window. Different from linear filters such as the mean filter, median filter has attractive properties for suppressing impulse noise while preserving edges. A median filter is used in this study due to its edge preserving feature [13-16].

Background elimination: Many image processing applications require the differentiation of objects from the image background. Thresholding is the most trivial and easily applicable method for this purpose. It is widely used in image segmentation [17],[18]. We used the threshold technique for differentiating the signature pixels from the background pixels.

(a) (b)

Fig. 1. (a) Captured signature (b) signature image with background removed

Signature normalization: Signature dimensions may vary due to the irregularities in the image scanning and capturing process. Furthermore, height and width of signatures vary from person to person and, sometimes, even the same person may use different size signatures. First, we need to eliminate the size differences and obtain a standard signature size for all signatures. After this normalization process, all signatures will have the same dimensions. In this study, we used a normalized size of 80x80 pixels for all signatures that will be processed further. During the normalization process, the aspect ratio between width and height of a signature is kept intact. The normalization process is demonstrated in the following figure.



(a) (b)

Fig. 2. Signature normalization a) Original signature image b) Normalized signature

Skeletonization: The aim of the skeletonization is to extract a region-based shape feature representing the general form of an object. We used Zhang-Suen Skeletonization Algorithm [19],[20].

3 Moment Invariant Method

Moment invariants are properties of connected regions in binary images that are invariant to translation, rotation and scaling. They can be easily calculated from region properties and they are very useful in performing shape classification and part recognition. One of the techniques for generating invariants in terms of algebraic moment was originally proposed by Hu [21]. The moment invariants used in our research are computed using the equations given in Table-1(b) for all signatures at various angles[22-25].

Table 1. a) Formulas used for specific central moments b) List of the derived invariant moments

Central Moments	Derived Invariant Moments
$\mu_{00}=m_{00}$	$I_1=\eta_{20}+\eta_{02}$
$\mu_{10}=0$	$I_2=(\eta_{20}-\eta_{02})^2+4\eta_{11}{}^2$
$\mu_{01}=0$	$I_3=(\eta_{30}-3\eta_{12})^2+(3\eta_{21}-\eta_{03})^2$
$\mu_{20}=m_{20}-x'm_{00}$	$I_4=(\eta_{30}+\eta_{12})^2+(\eta_{21}-\eta_{03})^2$
$\mu_{02}=m_{02}-y'm_{01}$	$I_5=(\eta_{30}-3\eta_{12})(\eta_{30}+\eta_{12})^2((\eta_{30}+\eta_{12})^2-3(\eta_{21}+\eta_{03})^2)+(3\eta_{21}-\eta_{03})(\eta_{21}+\eta_{03})(3(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2)$
$\mu_{11}=m_{11}-y'm_{10}$	$I_6=(\eta_{20}-\eta_{02})((\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2)+4\eta_{11}(\eta_{30}+\eta_{21})(\eta_{21}+\eta_{03})$
$\mu_{30}=m_{30}-3x'm_{20}+2x'm_{10}$	$I_7=(3\eta_{12}-\eta_{30})(\eta_{30}+\eta_{12})((3\eta_{30}\eta_{12})-3(\eta_{21}+\eta_{03})^2)+(3\eta_{21}-\eta_{03})(\eta_{21}+\eta_{03})((3\eta_{30}\eta_{12})^2-(\eta_{21}+\eta_{03})^2)$

Feature vectors used for signature identification are generated using moment invariants. For this purpose, we produced twelve different sets of feature vectors for every signature where each set consisted of seven moment invariant values listed in Table 1(b). First, a feature vector for the normalized signature is produced using moment invariants, and these seven feature vector values are saved in a database. Then, the normalized signature is rotated using the two-dimensional rotation equation given below:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2}$$

Twelve different sets of feature vectors are calculated to correspond to twelve different reference angles. The normalized signature image is rotated from 0° to 360° in 30° increments. In total, 84 features were extracted for each signature. A sample feature vector is shown in Table 2.

Table 2. A sample seven feature vectors for a signature

	Moments invariants
I_1=	5.18647933247972E-0001
I_2=	1.72351384466328E-0001
I_3=	1.32449449923652E-0001
I_4=	2.13629276301316E-0001
I_5=	1.45538602648840E-0002
I_6=	8.86884674185101E-0002
I_7=	5.61213979201577E-0003

4 ANN Design for Signature Recognition and Verification

In this stage, two separate sequential neural networks are designed, one for signature recognition, and another for verification (i.e. for detecting forgery). The signature and verification system solution block diagram are given in Figure 3.

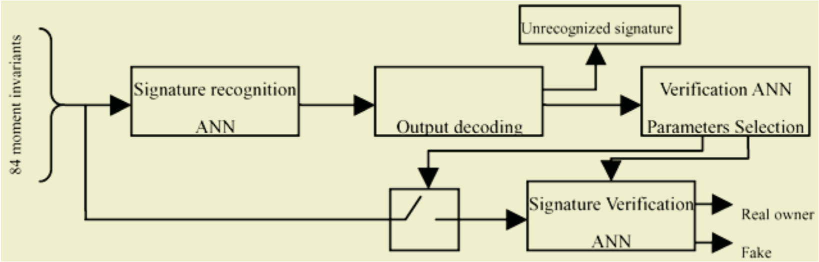

84 moment invariants
Signature recognition ANN
Output decoding
Unrecognized signature
Verification ANN Parameters Selection
Signature Verification ANN
Real owner
Fake

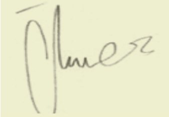
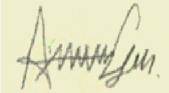
Fig. 3. The overall structure of the system

4.1 Training for Signature Recognition

We designed a multilayer feed forward artificial neural network for recognition of off-line digitized signatures. The proposed ANN consists of 84 input variables, 60 hidden neurons, and 40 output variables, and it is designed to recognize one signature at a time. Back-propagation algorithm is used for training.

First, an input/output database is created manually for training and testing the ANN. Each input vector consists of 84 moment invariants obtained for a signature at a particular rotation angle, as explained earlier in section 3. We used six samples for every signature. These six signatures are divided into two sets each containing three signatures. One of these sets is used in the training of ANN and the other set is used for testing. The database contained a total of 40 different signature images which are used for both training and testing. Since 3 input vectors for each image are used for training purposes, there are a total of 120 (40*3) input vectors (data sets) in the training set. The remaining 120 data sets are used for testing. ANN contained 40 binary output values, each corresponding to one signature being tested as shown in Table 2. Under normal (correct) operation of ANN, only one output is expected to take a value of "1", indicating the recognition of a signature represented by that particular output. The other output values must remain zero. In general, the number of outputs must be equal to the number of signatures being considered. Table 3 shows a number of real input/output vectors used in the training set, which are obtained from a set of signatures.

Table 3. A sample of output coding for signature recognition NN.

					Outputs		
	Sample	Signature images	O_1	O_2	O_3	O_4	O_{30}
	1		1	0	0	0	0
1	**2**		1	0	0	0	0
	3		1	0	0	0	0
	4		0	1	0	0	0
2	**5**		0	1	0	0	0
	6		0	1	0	0	0
	7		0	0	1	0	0
3	**8**		0	0	1	0	0
	9		0	0	1	0	0
			.	.	.		
			.	.	.		
			.	.	.		
	118		0	0	0	0	1
40	**119**		0	0	0	0	1
	120		0	0	0	0	1

4.2 Signature Verification

In this part of the study, our purpose is to authenticate a signature, i.e., to verify that the signature is not counterfeit and that it really belongs to the person who is claimed to be the owner of the signature. The ANN used for this purpose is also a multilayer feed forward network which consists of 84 input variables, 40 hidden neurons, and 2 output variables indicating whether the signature is fake or true. Backpropagation algorithm is used for training. The training data set is obtained from six original (authenticated) signatures provided by the real owner and six fake signatures. As was done for the preparation of the training data for the ANN used in recognition, twelve invariant vectors per signature are used in the training set. A sample set of three signatures belonging to the same person is shown in Figure 4.

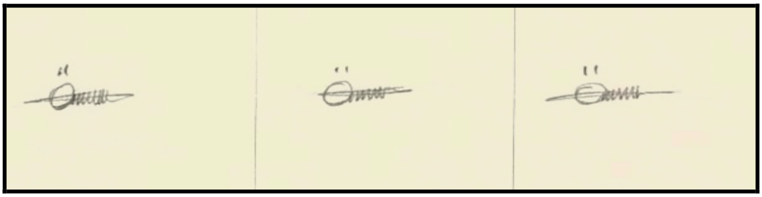


Fig. 4. Tree signatures which belong to the true owner of the signature

5 Implementation and Test Results

5.1 Signature Recognition

We used a Windows interface program. This software allowed the signature images to be loaded one at a time and used in training and testing. First, the signature image is captured using a scanner. Then, through several image processing operations, it is converted to binary and normalized to an 80*80 pixel image, and thinned, as explained earlier. Moment invariant values are obtained from the thinned image, which is then used as the input vector to the ANN. After the training of the ANN for signature recognition, the system is ready to recognize a given signature.

The signature recognition system is tested using 40 signatures chosen at random. The images were obtained using the following properties: i) Signatures were signed inside a special framed area. ii) Images were scanned with a simple A4 size scanner.

As explained in Section 4.1., 40 images in our database belonging to 40 different signatures are used for both training and testing. Since 3 (out of 6) input vectors for each image were used for training purposes, there are only 120 (40*3) input vectors (data sets) left to be used for the test set. Under normal (correct) operation of the ANN, only one output is expected to take a value of “1” indicating the recognition of a signature represented by that particular output. The other output values must remain zero. The output layer used a logic decoder which mapped neuron outputs between 0.5-1 to a binary value of 1. If the real value of an output is less than 0.5, it is represented by a “0” value. The ANN program recognized all of the 40 signatures correctly, a 100% recognition rate. We also tested the system with 10 random signatures which are not contained in the original database. Only two of these signatures which are very similar to at least one of the 40 stored images, resulted in “false positives” (output > 0.5) while the remaining 8 are recognized correctly as not belonging to the original set (the output value was <= 0.5). Since the recognition step is always followed by the verification step, these kinds of false positives can be easily caught by our verification system. In other words, the verification step serves as a safeguard against “false positives” as well as “false negatives”.

5.2 Testing the Verification System

Training for verification is explained in Section 4.2. Signatures used for testing the verification system are obtained the same way as in the recognition system. We tested the verification software using 80 signatures: 40 imitations (counterfeit signatures) and 40 true signatures. The program detected (classified) 34 true signatures and 39 counterfeits correctly. In other words, almost all counterfeit signatures were detected correctly. Only six signatures were classified as counterfeits while they were not (i.e. “false negative”). Obviously, a “false negative” should be more acceptable in comparison to a “false positive”, because the person can always be given a second chance to prove that the signature is his/hers. On the other hand, a false positive in verification carries a lot of risk.

6 Conclusion

In this study, we presented an off-line signature recognition and verification system which is based on image processing, moment invariants, and ANNs. Moment invariants which are used as input features for the ANN are obtained from thinning signature images. Two separate sequential ANNs are used, one for signature recognition and another for verification. Our recognition system exhibited a 100% success rate by identifying correctly all of the 40 signatures that it was trained for. However, it exhibited poor performance when it was presented with signatures that it was not trained for earlier. We did not consider this as a “high risk” case, because recognition step is always followed by the verification step and these kinds of false positives can be easily caught by the verification system. The verification system missed only one counterfeit signature. However, its verification for true signatures lacked some accuracy. We think that this is also acceptable because a person can always be given a second chance to prove the ownership of a signature.

Generally, the failure to recognize/verify a signature was due to poor image quality and high similarity between two signatures. Recognition and verification ability of the system can be increased by using additional features in the input data set. In the future, we plan to expand the input feature vector by the addition of some grid information and features related to texture.

References

1. Parizeu, M., Plamondon, R.,: A Comparative Analysis of Regional Correlating Dynamic Time Warping, And Skeletal Tree Matching for Signature Verification. IEEE transactions on Pattern Analysis and Machine Intelligence, **12** (1990) 710-717

2. Brault, J., Plamondon, R.: Segmenting Handwritten Signatures at their Perceptually Important Points. IEEE Transactions on Pattern Analysis and Machine Intelligence, **15** (1993) 953-957
3. Lee, L., Berger, T., Aviczer, E.: Reliable On-line Human Signature Verification Systems. IEEE Transactions on Pattern Analysis and Machine Intelligence, **18** (1996) 643-647
4. Xuhang, X., Graham, L.: Signature Verification Using a Modified Basian Network. Pergamon Pattern Recognition, **35** (2002) 983-995
5. Yuan, Y.T. , Ernest, C.M.L.: New Method for Feature Extraction Based on Fractal Behavior. Pergamon Pattern Recognition, **35** (2002) 1071-1081
6. Ismail, M.A., Samia, G.: Off-line Arabic Signature Recognition and Verification. Pergaman Pattern Recognition, **33** (2000) 1727-1740
7. Qi, Y. , Hunt, B.R.: Signature Verification Using Global and Grid Features. Pattern Recognition, **27** (1994) 1621-1629
8. Yedekcioglu, O.A., Akban, M.B. , Lim, Y.H.: Off-line Signature Verification with Thickened Templates. COMCON5 Proceedings of 5th International Conference on Advanced in Communication and Control, Crete, Greece, (1995) 131-142
9. Han, K. , Sethi, I.K.: Handwritten Signature Retrieval and Identification. Pattern Recognition Letters, **17** (1996) 83-90
10. Baltzakis, H., Papamorkos, N.: A New Signature Verification Technique Based on a Two-stage Neural Network Classifier. Pergoman Engineering Aplication of Intelligence, **14** (2001) 95-103
11. Droughord, J., Plamondon, R., Godbout, M.: A Neural Network Approach to Off-line Signature Verification Using Directional PDF. Pattern Recognition, **29** (1996) 415-424
12. Luong, C.M.: Introduction to Computer Vision and Image Processing. web site:
http://www.netnam.vn/unescocourse/computervision/comp_frm.htm
13. Lim, J.S.: Two-dimensional and Image Processing. Prentice-Hall (1990)
14. Yang, X. and Toh, P.S.: Adaptive Fuzzy Multilevel Median Filter. IEEE Transaction on Image Processing, **4** (1995) 680-682
15. Hwang, H., Haddad, R.A.: Adaptive Median Filters: New Algorithm and Results. Transactions on Image processing, **4** (1995) 449-505
16. Rosenfeld, A.: Digital Picture Processing. Academic Press Inc. (1982)
17. Erdem, U.M.: 2D Object Recognition In Manufacturing Environment Using Implicit Polynomials and Algebraic Invariants. Master Thesis, Bogazici University (1997)
18. Fu, K.S. , Mui, J.K.: A survey On Image Segmentation. Pattern Recognition, Pergoman Press, **13** (1981) 3-16
19. Parker, J., R.: Practical Computer Vision Using C. Wiley Computer Publishing (1994)
20. Sonka, M., Hlavac, V. and Boyle, R.: Image Processing, Analysis, and Machine Vision. Pws. Pub. Co. (1998)
21. Hu, M.: Visual Pattern Recognition by Moment Invariants. IRE Trans. Inform, (1962) 179-187
22. Koker, R., Oz, C. and Ferikoglu, A.: Object Recognition Based on Moment Invariants Using Artificial Neural Networks. Proceedings of 3rd International Symposium an Intelligent Manufacturing Systems, Sakarya (2001)
23. Awcock, G.J. , Thomas, R.: Applied Image Processing. McGraw Hill, Inc. (1996)
24. Reiss, T.H.: The Revised Fundamental Theorem of Moment Invariants. IEEE Transaction on Pattern Analysis and Machine Intelligence, **13** (1991) 830-834
25. Ustun, A.: Cisim Tanima Problemine Yapay Sinir Aglarinin uygulamasi. MSc Thesis, ITU (1999)

Handwritten Digit Recognition with Kernel-Based LVQ Classifier in Input Space

Hairong Lv and Wenyuan Wang

Department of Automation, Tsinghua University
Beijing 100084, China
lvhairong98@mails.tsinghua.edu.cn, wwy-dau@tsinghua.edu.cn

Abstract. In this paper, a kernel-based LVQ classifier in input space is proposed to recognize handwritten digit. Classical Learning Vector Quantization is performed in the input space through Euclidean distance, but it doesn't work well when the input patterns are highly nonlinear. In our model, the kernel method is used to define a new metric of distance in input space so we can get a direct view of the clustering result. At last, we test our model by handwritten digit recognition using MNIST database and it obtains better recognition performance than traditional LVQ.

1 Introduction

Optical character recognition is a typical field of application for automatic classification methods. It's widely used in zip code recognition, automatic reading of bank checks, etc. In this paper, we focus on the recognition of isolated handwritten digit, a task which is known to be difficult and still lacks a technically satisfactory solution.

Approaches generally used in solving this problem fall in one of two categories: the global analysis and structural analysis approaches. The use of neural networks offers an alternative and easy method for handwritten digit recognition. Many neural networks have been often exploited in handwritten digit recognition and a common practice is to train a multilayer perceptron (MLP) classifier to output one of the ten class labels [1],[2],[3],[6],[8]. By directly training the network with sufficiently large data set, the recognition rate can be quite high. But in general, an MLP classifier can yield quite different classification boundaries with respect to different initial conditions or different training sets from the same data space.

Clustering based neural networks are rarely used in handwritten digit recognition. The most well-known clustering neural net work is the Self Organizing Map (SOM) introduced by T. Kohonen [9]. Nevertheless, classical SOM suffers greatly of having a very low recognition rate when the trained network is used as a pattern recognition system. This is absolutely normal since the SOM has never been developed in order to obtain a pattern recognition system. For this reason, some modified SOMs are proposed, such as Learning Vector Quantization (LVQ) [9], Fuzzy SOM[4], ASSOM [5], combined SOM [7]. Because of adopting Euclidean distance measure, LVQ and SOM can cause bad performance when the data is non-spherical distribution, and especially contains noises or outliers.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 203–208, 2005.

In this paper, we propose a kernel-based LVQ for handwritten digit recognition. We use kernel-induced distance measures to replace original Euclidean one. It obtained better recognition performance than classical LVQ.

This paper is organized as follows. In the next section, we first review the classical Learning Vector Quantization algorithm and propose a Kernel LVQ which is called KLVQ. Section 3 presents KLVQ-based modular classification scheme for handwritten digit recognition and Section 4 gives experimental results. Finally, concluding remarks are discussed in Section 5.

2 Kernel-Based LVQ

As already mentioned in the first section, LVQ for unsupervised clustering is a special case of the SOM network. Before presenting KLVQ algorithm, we first introduce the classical LVQ.

2.1 Classical LVQ

Suppose w_i in R^s is the weight vector of the neuron i and the input sample at time t is x_k . Classical LVQ updates its neuron i as follows:

$$w_i(t+1) = w_i(t) + \alpha(t)h_{ik}\left(x_k - w_i(t)\right). \tag{1}$$

Here $\alpha(t)$ is the learning rate, $0 < \alpha(t) < 1$. h_{ik} is an indicative function whose value is 1 when i is the winner node and 0 otherwise.

In LVQ, suppose the neuron i wins the competition, it satisfies:

$$\forall j, \quad \left\|x_k - w_i(t)\right\| \le \left\|x_k - w_j(t)\right\|. \tag{2}$$

The learning rules above is called LVQ1. There are some improvements of it, such as LVQ2[9].

2.2 KLVQ

From (1), we can see that the LVQ weight-adjusting algorithm depends on metric of distance. When the input patterns are highly nonlinear, the Euclidean-norm-based LVQ doesn't work well. We should look for some other metrics of distance. The kernel method brings a promising approach[10],[11].

Given input set X and a nonlinear mapping function Φ , which maps x_k from the input space X to a new space F with higher or even infinite dimensions. The Euclidean distance can be replaced with

$$\begin{aligned} J(w_j) &= \left\| \Phi(x_k) - \Phi(w_j) \right\|^2 \\ &= \Phi(x_k)^T \Phi(x_k) + \Phi(w_j)^T \Phi(w_j) - 2\Phi(x_k)^T \Phi(w_j) . \\ &= K(x_k, x_k) + K(w_j, w_j) - 2K(x_k, w_j) \end{aligned} \tag{3}$$

Where $K(x_i, x_j) = \Phi(x_i)^T \Phi(x_j)$ and the kernel function satisfies Mercer condition. Now, the equation (1) can be rewritten as (4),

$$\begin{aligned} w_i(t+1) &= w_i(t) - \alpha(t) h_{ik} \nabla J(w_i) \\ &= w_i(t) - \alpha(t) h_{ik} \left(\frac{\partial K(w_i, w_i)}{\partial w_i} - 2 \frac{\partial K(x_k, w_i)}{\partial w_j} \right) . \end{aligned} \tag{4}$$

There are four classical kernel function that satisfies Mercer condition:
(1) Polynomial:

$$K(x, y) = (x^T \cdot y)^d, \quad d \geq 2 . \tag{5}$$

(2) radial basis:

$$K(x, y) = \exp(-\left\| x - y \right\|^2 / 2\sigma^2) . \tag{6}$$

(3) Cauchy:

$$K(x, y) = 1/(1 + \left\| x - y \right\|^2 / \sigma^2) . \tag{7}$$

(4) logarithm:

$$K(x, y) = \log(1 + \left\| x - y \right\|^2 / \sigma^2) . \tag{8}$$

According (5), (6), (7), (8), we can derive from (4) :

$$w_i(t+1) = w_i(t) - \alpha(t) h_{ik} (2d(w_i^T)^{d-1} w_i - (x_k^T w_i)^{d-1} x_k) . \tag{9}$$

$$w_i(t+1) = w_i(t) - \alpha(t) h_{ik} \frac{2}{\sigma^2} e^{-\left\| x_k - w_i \right\|^2 / \sigma^2} \bullet (x_k - w_i) . \tag{10}$$

$$w_i(t+1) = w_i(t) + \alpha(t) h_{ik} \frac{1}{\sigma^2 \left(1 + \left\| x_k - w_i \right\|^2 / \sigma^2 \right)^2} \bullet (x_k - w_i) . \tag{11}$$

$$w_i(t+1) = w_i(t) + \alpha(t) h_{ik} \frac{1}{\sigma^2 \left(1 + \left\| x_k - w_i \right\|^2 / \sigma^2 \right)} \bullet (x_k - w_i) . \tag{12}$$

(9)~(12) are weight-adjusting equations of KLVQ.
In KLVQ, the winner neuron i is defined as:

$$\begin{aligned}\left\|\Phi(x_k)-\Phi(w_i)\right\| &= \min_j \left\|\Phi(x_k)-\Phi(w_j)\right\|^2 \\ &= \min_j \left(K(x_k,x_k)+K(w_j,w_j)-2K(x_k,w_j)\right)\end{aligned}. \tag{13}$$

3 The Proposed System

In this section, we propose a novel approach for handwritten digit recognition with KLVQ classifier.

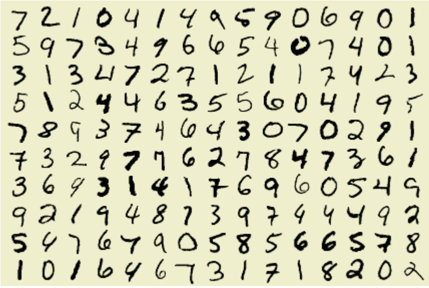


Fig. 1. Patterns in MNIST

We use MNIST to train and test our system. There are 60,000 patterns in the training set and 10,000 patterns in the testing set. The sets of writers of the training set and test set are disjoint. Some patterns in MNIST are shown in Fig. 1.

Before training/testing the KLVQ network, we just use the usual normalizing method to resize the digit images from 28×28 to 16×16 to reduce the dimension of input and then transform them to binary images.

Suppose that each digit has K subclasses, so the competitive layer has 10×K neurons and the linear layer has 10 neurons. If the *i*th neuron of linear layer outputs 1, then the recognition result is *i*. The proposed system is shown in Fig. 2.

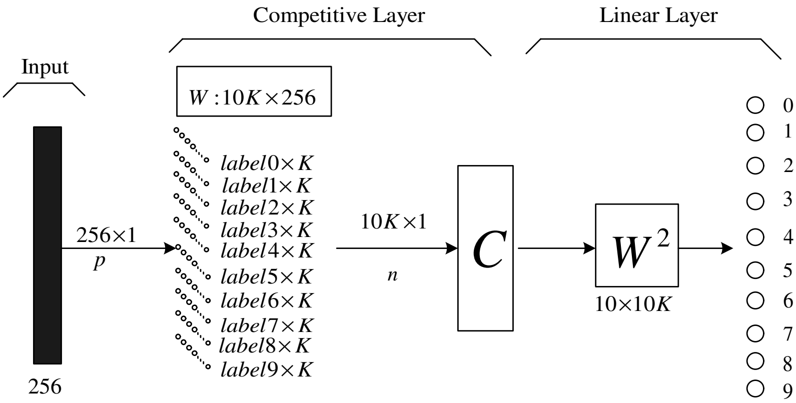


Fig. 2. Proposed Handwritten Digit Recognition System

4 Experiment Results

We test our model by changing the kernel function and the value of K. The experiment results are shown in Table 1.

Table 1. Recognition Rate in Different Conditions

K value	LVQ	KLVQ Polynomial $(d=2)$	KLVQ RBF $(\sigma=1)$	KLVQ Cauchy $(\sigma=1)$	KLVQ Log $(\sigma=1)$
10	96.23%	96.53%	97.12%	96.35%	96.36%
20	96.30%	96.61%	97.35%	96.55%	96.54%
30	96.50%	96.84%	97.60%	96.61%	96.63%
40	96.51%	96.99%	97.84%	96.77%	96.77%
50	96.56%	97.01%	98.02%	96.78%	96.81%
60	96.32%	96.91%	97.79%	96.74%	96.75%
70	96.29%	96.79%	97.67%	96.66%	96.65%
80	96.20%	96.50%	97.50%	96.54%	96.56%
90	96.13%	96.37%	97.34%	96.41%	96.40%

From Table 1, we can see that KLVQ can obtain better recognition performance than classical LVQ and RBF kernel function is the best suitable one for handwritten digit recognition. Besides, we also can know that the recognition performance varies with the number of subclass – K. In our experiment, when K is equal to 50, the system can achieve best recognition results with the recognition rate of 98.02%.

5 Conclusions

In this paper, a novel handwritten digit recognition approach is presented. The kernel-based LVQ shows appealing characteristics related to patters association as well as knowledge representation. The experiment results show that the performance of KLVQ is better than classical LVQ.

References

1. Ahmed, J., Alkhalifa, E.M.: Efficient Single Layer Handwritten Digit Recognition through An Optimizing Algorithm. Neural Information Processing, ICONIP '02. Proceedings of the 9th International Conference on, **5** (2002) 2464-2468
2. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, R., and Jackel: L.D.: Handwritten Digit Recognition with a Back-Propagation Network. In D. Touretzky (ed), Advances in Neural Information Processing Systems 2, Morgan Kaufman (1990)
3. Talukder, K.H.: Performance Analysis of Back-Propagation Learning Algorithm for Handwritten Digit Classification. Multi Topic Conference, 2002. Abstracts. INMIC 2002. International , **61** (2002)
4. Su, M., Lai, E., Tew, C.: A SOM-Based Fuzzy System and Its Application in Handwritten Digit Recognition. Multimedia Software Engineering, 2000. Proceedings. International Symposium on (2000) 253-258

5. Zhang, B., Fu, M., Yan, H., Jabri, M.A.: Handwritten Digit Recognition by Adaptive-Subspace Self-Organizing Map (ASSOM). Neural Networks, IEEE Transactions on , **10** (1999) 939-945
6. Oh, S.: Improving The Error Backpropagation Algorithm with a Modified Error Function. Neural Networks, IEEE Transactions on , **8** (1997) 799-803
7. Wu, J., Hong, Y.: Combined SOM and LVQ Based Classifiers for Handwritten Digit Recognition. Neural Networks, 1995. Proceedings., IEEE International Conference on , **6** (1995) 3074-3077
8. Sackinger, E., Boser, B.E., Bromley, J., LeCun, Y., Jackel, L.D.: Application of the ANNA Neural Network Chip to High-Speed Character Recognition. Neural Networks, IEEE Transactions on , **3** (1992) 498-505
9. Kohonen, T.: Self-organization and Associative Memory. 2nd Ed., Berlin: Springer – Verlag (1987)
10. Pan, Z.S., Chen, S.C., Zhang, D.Q.: A Kernel-based SOM Classification in Input Space. Acta Electronica Sinica, **32** (2004) 227-231
11. Andras, P.: Kernel-Kohonen Networks. International Journal of Neural Systems, **12** (2002) 117-135

Recognition of English Business Cards Using Enhanced Hybrid Network

Kwangbaek Kim[1], Jaehyun Cho[2], and Amsuk Oh[3]

[1] Dept. of Computer Engineering, Silla University, Korea
gbkim@silla.ac.kr
[2] Dept. of Computer Information, Catholic University of Pusan, Korea
jhcho@cup.ac.kr
[3] Dept. of Multimedia Engineering, Tongmyoung Univ. of Information Technology, Korea
asoh@tit.ac.kr

Abstract. In this paper, we proposed a novel method for the recognition of English business cards by using the projection method and the enhanced hybrid network. The recognition of business cards consists of the extraction phase of character areas and the recognition phase of extracted characters. In the extraction phase, first of all, noises are removed from the images of business cards, and the feature areas including character strings are separated from the business card images by using the horizontal smearing method and the 8-directional contour tracking method. And using the image projection method, the feature areas are split into the areas of individual characters. We also proposed the enhanced hybrid network that organizes the middle layer effectively by using the enhanced ART1 neural network adjusting the vigilance threshold dynamically according to the homogeneity between patterns. In the recognition phase, the proposed neural network is applied to recognize individual characters. Our experiment result showed that the proposed recognition algorithm has higher success rate of recognition and faster learning time than the conventional neural network based recognition.

1 Introduction

Recently, for the increased need of the storage and management of analog sources of personal information such as business cards and memos etc., new technologies are coming out that recognize the business cards via the photo sensor equipped on mobile phones, PDAs and like mobile devices. For the efficient management of in-disorder business cards, it is required that business cards are automatically recognized to digital data and it is saved and managed on the database.

Generally, the recognition process of business cards consists of the extraction phase of character areas and the recognition phase of extracted characters [1]. In this paper, the feature areas including character strings are separated from the original images by using the horizontal smearing algorithm and the 8-directional contour tracking algorithm in the extraction phase. And using the image projection method, the feature areas are split into the areas of individual characters. The enhanced hybrid network is proposed that organize the middle layer effectively by using the enhanced ART1 neural network adjusting the vigilance threshold dynamically according to the

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 209–215, 2005.

homogeneity between patterns. In the recognition phase, the proposed neural network is applied to recognize individual characters. Our experiment result showed that the proposed recognition method has higher success rate of recognition and faster learning time than the existing neural network based recognitions.

2 Feature Area Extraction Algorithm

In the feature area extraction phase, first of all, noises are removed from the scanned images of business cards. Next, the feature areas including non-character components such as picture and logo etc. and the ones including character strings are extracted from the images, and from the latter areas individual character areas are split into and normalized for the character recognition.

2.1 Extraction of Character String Areas

In the business cards, characters have statistically constant width and space between them. And this paper connects component areas using the horizontal smearing method that removes spaces between two areas and combines into a larger area[2], and extracts the feature areas from the images by applying 8-directional contour tracking method[3] to the smeared images.

The feature areas are classified to two types, the character string area including only characters and the non-character areas including components such as pictures, logos etc. Because this paper has a concern to the character string areas, it selects only the character string areas among the feature areas considering the statistical characteristics of characters areas in the business cards and the general design pattern of the cards. The method that determines whether the feature area is a character string area or not is in detail as follows: If we assume that C_H and C_W are the height and the width of the feature area in pixels respectively, $C(x, y)$ is the value, 0 or 1 of the pixel (x, y), and I_H is the height of the business card image.

$$H_C = I_H \times \alpha, \quad R_C = \frac{\sum_{x=1}^{C_H}\sum_{y=1}^{C_W}\{C(x,y)\times 1\}}{C_H \times C_W} \tag{1}$$

$$\begin{aligned}
&if\ \{(C_H \geq H_C) \cap (R_C \geq t_L) \cap (R_C < t_U)\} then\\
&\quad the\ feature\ area\ is\ a\ character\ string\ area\\
&else\\
&\quad the\ feature\ area\ is\ a\ non-character\ area
\end{aligned}$$

Where H_C is the allowable minimum height of the feature areas, R_C is the rate of black pixels to the total pixels in the feature area, and the constants α, t_U, t_L are empirically set as 0.09, 0.95, and 0.30 respectively, based on the statistical analysis of business card images. The character string areas selected are used in the next processing phases, extraction of character areas and character recognition.

2.2 Extraction of Individual Character Areas

This paper uses the image projection method to extract individual character areas from the character string areas in the original image. Because characters are isolated from the neighborhood ones in the business card image, the bi-directional, vertical and horizontal projection method is able to being efficiently applied. In this paper, the procedure to extract individual character areas from the character string areas consists of two steps. Step 1: Given a character string area, the vertical projection method is applied to the area, and as tracing the projection values in the horizontal direction, the position that has the projection value above the given threshold value p_V is determined as the left boundary of a individual character area. And after the left boundary is determined, the first position that becomes lower than p_V is determined as the right boundary of the individual character. The same process is repeatedly done until the left and right boundaries of all individual characters are extracted in the given feature area. Step 2: For individual areas having the left and right boundaries, the horizontal projection is applied. The same process as Step 1 is done in terms of the given threshold value p_H, and the top and bottom boundaries of the individual areas are determined. The threshold value p_V and p_H is empirically given.

3 Character Recognition using the Enhanced Hybrid Network

The RBF (Radial Basis Function) network have the characteristics such as fast learning time, generality and simplicity etc. and is mainly applied to the classification of learning data and the modeling of non-linear systems [4]. The middle layer of RBF network carries out the role of clustering the learning vectors by classifying homogenous learning vectors to the same cluster. In the clustering procedure, the homogeneity between learning vectors is represented as the distance between learning vectors, and if the distances between a learning vector and all vectors in a cluster are smaller than the given constant radius, the learning vector is classified to the cluster. But the usage of the constant radius in the clustering involves primarily the errors in clustering and therefore the decrease of the success rate of recognition [5].

In the ART1 network, the vigilance threshold determines inversely the allowable degree of mismatch between any input pattern and saved patterns [6]. The large value of vigilance threshold classifies an input pattern to new category in spite of a little mismatch between the pattern and the expected patterns, and oppositely the small value may allow the classification of the input pattern to an existing cluster in spite of a much mismatch. Moreover, because many application of image recognition based on the ART1 network give the empirical value to the vigilance threshold so that the decrease of the success rate of recognition may be incurred. To correct this defect, this paper enhances the ART1 network adjusting the vigilance threshold dynamically according to the homogeneity between patterns by using Yager's intersection operator [7], one of fuzzy connection operators. The fuzzy intersection operator has the property that the output value is not greater than the minimum value in all input values, and the Yager's intersection operator is described in Eq. (1).

$$\mu_{X_1 \cap X_2} = 1 - \min\left\{1, \left((1-\mu_{X_1})^p + (1-\mu_{X_2})^p\right)^{\frac{1}{p}}\right\} \tag{2}$$

Eq. (2) shows the equation applied to the ART1 network for refinement in this paper, which dynamically adjusts the vigilance threshold ρ by using Yager's intersection operator Eq. (1).

$$\rho(n+1) = 1 - \min\left(1, \sqrt{(1-\rho(n))^2 + (1-\rho(n-1))^2}\right) \tag{3}$$

This paper enhances the hybrid network by applying the modified ART1 network as the middle layer, as shown in Fig. 1.

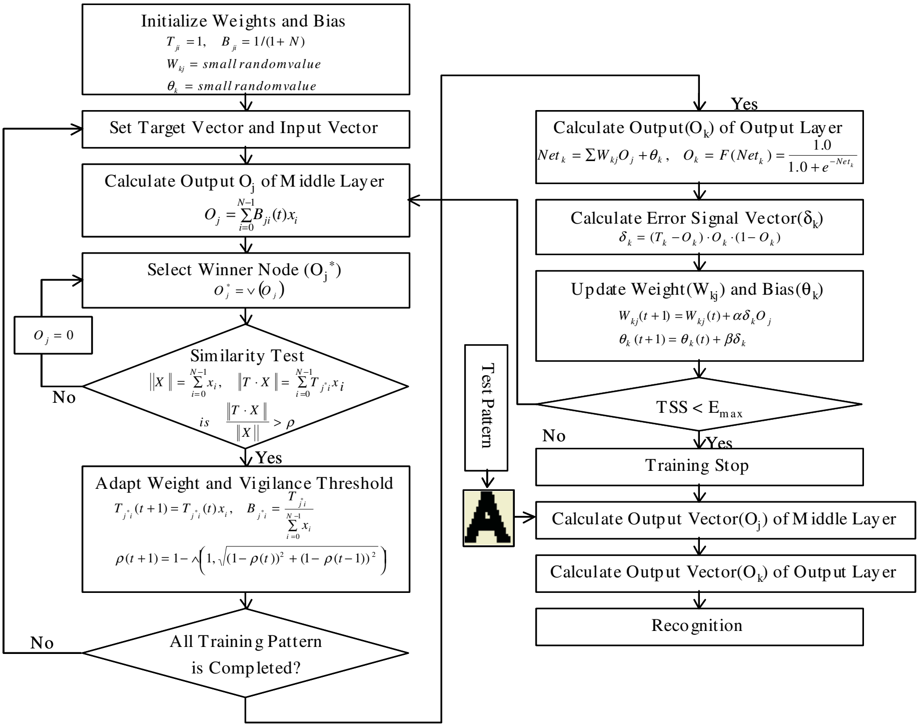

Initialize Weights and Bias
$T_{ji} = 1, \quad B_{ji} = 1/(1+N)$
$W_{kj} = small random value$
$\theta_k = small random value$
Set Target Vector and Input Vector
Calculate Output O_j of Middle Layer
$O_j = \sum_{i=0}^{N-1} B_{ji}(t) x_i$
Select Winner Node (O_j^*)
$O_j^* = \vee(O_j)$
$O_j = 0$
Similarity Test
$\|X\| = \sum_{i=0}^{N-1} x_i, \quad \|T \cdot X\| = \sum_{i=0}^{N-1} T_{j^*i} x_i$
$is \ \frac{\|T \cdot X\|}{\|X\|} > \rho$
No
Yes
Adapt Weight and Vigilance Threshold
$T_{j^*i}(t+1) = T_{j^*i}(t) x_i, \quad B_{j^*i} = \frac{T_{j^*i}}{\sum_{i=0}^{N-1} x_i}$
$\rho(t+1) = 1 - \wedge\left(1, \sqrt{(1-\rho(t))^2 + (1-\rho(t-1))^2}\right)$
All Training Pattern is Completed?
No
Yes
Calculate Output(O_k) of Output Layer
$Net_k = \sum W_{kj} O_j + \theta_k, \quad O_k = F(Net_k) = \frac{1.0}{1.0 + e^{-Net_k}}$
Calculate Error Signal Vector(δ_k)
$\delta_k = (T_k - O_k) \cdot O_k \cdot (1 - O_k)$
Update Weight(W_{kj}) and Bias(θ_k)
$W_{kj}(t+1) = W_{kj}(t) + \alpha \delta_k O_j$
$\theta_k(t+1) = \theta_k(t) + \beta \delta_k$
TSS < E_{max}
No
Yes
Training Stop
Test Pattern
A
Calculate Output Vector(O_j) of Middle Layer
Calculate Output Vector(O_k) of Output Layer
Recognition

Fig. 1. Learning and recognition algorithm of the enhanced hybrid network

4 Experiment Results

We implemented the recognition system of business cards based on the proposed algorithm by using C++ Builder tool on the Intel Pentium-IV 2GHz PC, and performed the recognition experiment by using 40 business card images with 1500x800 pixel size. Fig. 2 shows a success case of individual character extraction, and oppositely Fig. 3 shows a failure case. As shown in Fig. 3, the proposed method for individual character extraction doesn't extract the connected characters in the business card images. Table 1 shows the results of the feature area extraction in our experiment. The results show the total number of extracted areas from 40's images, the number of success of extraction and the number of failure of extraction. As shown in the Table 1, the number of extraction failure for the character string areas is ignorable compared with the number of success.

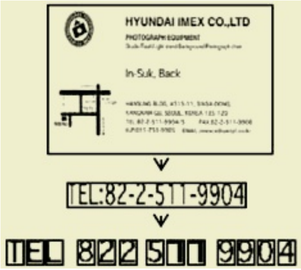


Fig. 2. Success case of individual character extraction

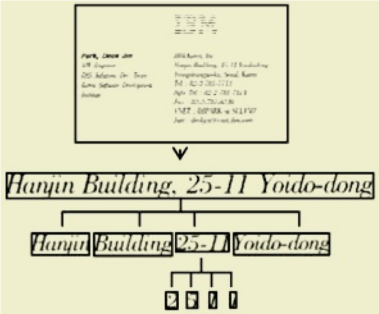


Fig. 3. Failure case of individual character extraction

Table 1. Results of feature area extraction in the experiment

	Character string area	Non-character area
Total number of extracted areas	485	50
Number of success	483	47
Number of failure	2	3

The results of the individual character extraction on the extracted feature areas are shown in the Table 2. As shown in the Table 2, the failure rate of individual character extraction is near to 9%, which is incurred from the failure of extraction of continually connected characters. The results show the defect of the individual character extraction method based on the image projection.

Table 2. Results of individual character extraction in the experiment

	Results
Total number of individual character in the images	6352
Number of success	5821
Number of failure	531

Table 3. The comparison of the performance for learning and recognition

Hybrid network based on the enhanced ART1		
	Middle layer nodes / Input patterns	The rate of success in recognition
English character	279 / 470	2767 / 2876
Number	82 / 160	2154 / 2240
Special character	13 / 25	655 / 705
Hybrid network base on the conventional ART1		
	Middle layer nodes / Input patterns	The rate of success in recognition
English character	408 / 470	2759 / 2876
Number	135 / 160	2150 / 2240
Special character	19 / 25	641 / 705

For the performance evaluation of character recognition, as showed in Table 3, the results for learning and recognition were measured and compared when the enhanced ART1 proposed in this paper and the conventional ART1 were applied to the middle layer of the hybrid network. In the recognition experiment, the individual characters were classified to the three classes, the English characters, the numbers and the special characters, and for each class, the number of nodes of the middle layer and the success rate of recognition were measured. For the hybrid network based on the conventional ART1, the initial vigilance threshold was set and fixed as 0.85, while in the hybrid network based on the enhanced ART1, the vigilance threshold was set initially as 0.85 and adjusted dynamically in the learning process. For the 470 English characters, the proposed hybrid network and the hybrid network based on the conventional ART1 generated 279 and 408 nodes respectively.

That is, the proposed hybrid network is more effective in learning individual characters on the business card images. The success rate of character recognition was measured in each character class for the performance comparison of recognition.

5 Conclusions

This paper proposed a novel recognition algorithm of English business cards. The proposed algorithm, first, removes noises from the scanned image and transforms to the binary image. Based on the statistical analysis of the design pattern of business cards, the algorithm extracts the feature areas including characters, picture and logo etc. from the preprocessed image by using the horizontal smearing method and the 8-directional contour tracking method, and extracts the individual character areas from the feature areas including only characters by using the image projection method. Also, this paper proposed the enhanced hybrid network adapting the modified ART1 network to the middle layer, and applied it to the character recognition phase.

The recognition experiments on the 40's images of English business cards showed that the feature area extraction method based on the smearing and the contour tracking methods is so very finely suitable for the business card images as the rate of failure can be ignored, and the individual character extraction method based on the image projection has the defect of the extraction failure of continually connected characters. Moreover, the enhanced hybrid network recognized effectively the individual character so that it showed the high success rate of recognition.

In the future work, we will investigate and develop the preprocessing and recognition methods for the business cards with the hand-written characters.

References

1. Kim, K. B., Kim, C. K., and Kim, J. W.: A Study on the Recognition of an English Calling Card by Using Contour Tracking Algorithm and Enhanced ART1. Journal of Korea Intelligent Information Systems, **8** (2002) 105-116
2. Gonzalez, R. C. and Wintz, P.: Digital Image Processing. Addison-Wesley Publishing Company Inc. (1977) 88-102
3. Sik, W. N. and Koo, S. Y.: A Parallel Thinning Algorithm by the 8-Neighbors Connectivity Value. Transactions of The Korea Information Processing, **2** (1995) 701-710
4. Panchapakesan, C., Ralph, D., and Palaniswami, M.: Effects of Moving the Centers in an RBF Network. Proceedings of IJCNN, **2** (1998) 1256-1260
5. Kim, K. B., Kim, Y. J., and Oh, A. S.: An Intelligent System for Passport Recognition Using Enhanced RBF Network. Lecture Notes in Computer Science, LNCS **3314**. (2004) 762-767
6. Kim, K. B. and Park, C. S.: An Enhanced Fuzzy Multilayer Perceptron. Lecture Notes in Computer Science, LNCS **3316** (2004) 977-982
7. Zimmermann, H. J.: Fuzzy set Theory and It's Applications. Kluwer Academic Publishers (1991)

A Novel Approach for License Plate Recognition Using Subspace Projection and Probabilistic Neural Network*

Yafeng Hu, Feng Zhu, and Xianda Zhang

Department of Automation, Tsinghua University, Beijing 100084, China
{yafh99,zxd-dau}@mail.tsinghua.edu.cn

Abstract. License plate recognition has many applications in traffic systems. It is very difficult because images are usually noisy, broken or incomplete. In this paper, a novel robust approach for license plate recognition is proposed, which combines subspace projection with probabilistic neural network to improve the recognition rate. Probabilistic neural network is used as a classifier to identify low-dimension test samples which are obtained from actual license plate images by subspace projection. Experiment results show the effectiveness of the proposed method.

1 Introduction

License plate recognition (LPR) systems have many applications in intelligent traffic systems, such as the payment of parking fee, highway toll fee, traffic data collection, etc. Generally, a LPR system contains three major components: license plate localization, character segmentation and character recognition. This paper presents a new robust approach for character recognition.

The images of license plate may be noisy, broken or incomplete. So character recognition techniques should be able to tolerate these defects. The approach presented in this paper combines subspace projection with probabilistic neural network (PNN). First, the standard license plate images are used to form a signal subspace. In this step, singular value decomposition (SVD) is used. Then, training and testing samples are obtained from noisy images by weighted subspace projection. Finally, a PNN is trained by training samples. Testing samples are recognized through this neural network.

The approach proposed is tested on a set of experiments using 680 segmented license plate images. Compared with the template matching method [1] [2], the results show that this approach has better performance.

2 Weighted Orthogonal Subspace Projection

Subspace methods play an important role in many applications of signal processing. These methods assume that the pattern vectors can be adequately described by the corresponding subspace which has fewer dimensions than pattern space [3].

* Research mainly supported by the NSFC (No. 60375004).

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 216–221, 2005.

Although subspace can be obtained in theory via the eigenvalue decomposition (EVD) of a covariance matrix, the cost of computing can be prohibitively high and the corresponding subspace may not represent the original signal properly. In this paper, singular value decomposition (SVD) is introduced for obtaining the orthogonal subspace [3].

Let X be a space spanned by a set of standard segmented plate license images without noise, i.e.,

$$\mathrm{X} = span\left\{x_1, x_{2,} \cdots, x_i, \cdots, x_N\right\} \tag{1}$$

where x_i is the standard character image with $l \times s$ pixels, and N is the total number of character classes.

Reshaping every image to a vector in X , we can get a new space,

$$\mathrm{H} = span\left\{h_1, h_{2,} \cdots, h_i, \cdots h_N\right\} \tag{2}$$

where h_i comes from x_i . Denote the dimension of vector h_i as M , then $M = l \times s$. So H is the signal subspace. Any vector s , which satisfies $s \perp \mathrm{H}$, is called noise that has no effect on signal classification. All noise vectors form another subspace denoted as S , which is called noise subspace. S satisfies $S \perp \mathrm{H}$ and $V = S \oplus \mathrm{H}$, where V is the whole space.

Let $A_{M\times N} = \left[h_1, h_{2,} \cdots, h_N\right]$ be an $M \times N$ matrix, where $M \times N$ and each column represents a standard character image. The SVD of A is given by,

$$A_{M\times N} = U_{M\times N} \Sigma_{N\times N} V^T_{N\times N} \ . \tag{3}$$

The diagonal elements of Σ are the singular values of A in non-increasing order. Under assumption of Gaussian noise, this rank-N linear subspace approximates noisy images optimally in the sense of minimum least squares error.

Based on the above result, the input high-dimension noisy image r can be projected onto a low-dimensional signal subspace, namely

$$\tilde{y} = U^T_{M\times N} r \ . \tag{4}$$

However every dimension in $\tilde{y}$ has the same importance for classification, a proper weighted factor matrix is needed for $\tilde{y}$. Post multiplying $\tilde{y}$ by Σ^{-1} gives

$$y = \Sigma^{-1} \tilde{y} \ . \tag{5}$$

Then, y has the same mean square value in each dimension, which is helpful for recognition.

3 Probabilistic Neural Network for Pattern Classification

Probabilistic neural network (PNN) has been widely used in various pattern classification tasks due to its robustness [4]. It is closely related to Bayes classification rule and Parzen nonparametric probability density function estimation theory [5]. More neurons may be required than standard feedforward backpropagation (BP) networks, but

PNN can be designed in a fraction of the time it takes to train standard feed-forward networks.

PNN is a four-layer feedforward network as shown in Fig.1.

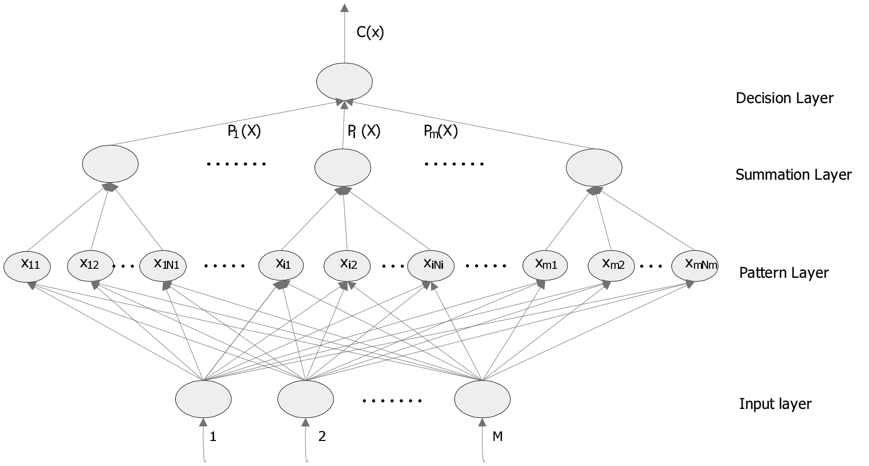


Fig. 1. Structure of the probabilistic neural network

The input layer contains M nodes to accept M -dimensional feature vectors. When an input vector x is presented, the neuron x_{ij} of the pattern layer computes distances from the input vector to the training vector, and produces its output,

$$\phi_{ij} = \frac{1}{(2\pi)^{M/2}\sigma_{ij}^{M}} \exp\left[-\frac{\left\| x - x_{ij} \right\|^2}{2\sigma_{ij}^2} \right] \tag{6}$$

where x_{ij} is the neuron vector and σ_{ij} is the smoothing parameter. The summation layer sums these contributions for each class, and produces its net outputs as the probabilities that x belongs to corresponding classes,

$$P_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \frac{1}{(2\pi)^{M/2}\sigma_{ij}^{M}} \exp\left[-\frac{\left\| x - x_{ij} \right\|^2}{2\sigma_{ij}^2} \right] \tag{7}$$

where N_i is the number of training samples in class i . Finally, the decision layer picks the maximum of these probabilities and classifies the input vector to that class.

4 Algorithm

The algorithm for segmented character image recognition from license plates has 4 steps: binarization, size normalization, signal subspace projection, and PNN training and testing.

4.1 Binarization

First, the images are converted into binary form. In this step, an adaptive threshold method is adopted, which is fairly robust for local changes of brightness [6]. Fig.2 shows two examples of binarization.

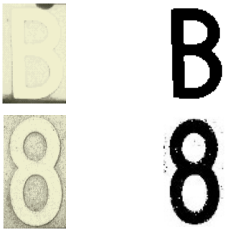


Fig. 2. Binarization result

4.2 Size Normalization

Each character image is normalized to a size of $30*20$ with background $48*32$. The barycenter of original image is mapped to the center of the normalized image, because it is less affected by noise.

4.3 Weighted Subspace Projection

A signal subspace is formed by using a set of standard character images. By SVD, $U_{M\times N}$ and $\Sigma_{N\times N}$ are obtained. Denote $P_{N\times M} = \Sigma_{N\times N}^{-1} U_{M\times N}^{T}$, and its value is saved as a weighted subspace projection matrix. Every preprocessed image x is reshaped to a vector r and multiplied by P, i.e.,

$$y = P_{N\times M}\, r \tag{8}$$

Then, y is the value of x projected on the signal space with weights.

4.4 PNN Training and Testing

PNN's design is straightforward and does not depend on training. The input layer has 34 nodes corresponding to the dimensions of input vector. The pattern layer has the same number of nodes as the training vectors. The summation layer has 34 nodes corresponding to 34 classes. The weights in summation layer are all "1". The output layer has 34 nodes too, which uses winner-take-all rule, produces a "1" for the winner and a "0" for the others.

5 Experiment Results

A database containing 680 character images is used to test the algorithm. These images are divided into 20 groups and each group contains 34 images that stand for all numbers and capital letters except "O" and "I" which seldom appear in license plate.

In the experiment, the cross validation technique is used to select learning samples. 10 groups of images are randomly drawn from the database for training and the others are for testing. Finally, the average recognition rate of 200 times is reported.

The results are compared with the template matching method [1] [2], which is one of the most tolerant methods to noise.

Fig.3 illustrates the recognition rate with the training sample number. It is obvious that the proposed approach provides much better performance.

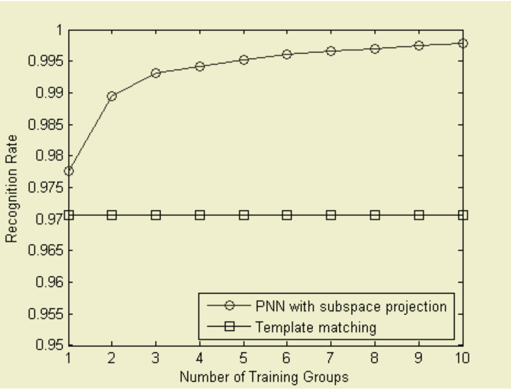


Fig. 3. Recognition rate versus the number of training groups

For testing the robustness of proposed algorithm, additive noise is then added to the testing images. The type of noise is "salt and pepper" with noise level p, where $p/2$ is the probability that a pixel flips to black or white. Fig.4 shows some binary images with additive noise.

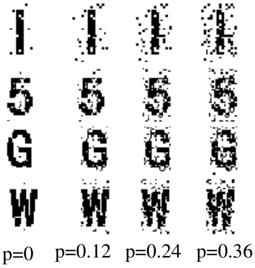


Fig. 4. Images corrupted by "salt and pepper" noise

Fig.5 illustrates the recognition rate with different level of additive noise. In Fig.5 (b), for contrasting the performance, the template is also corrupted by the additive noise in the template matching method. Compared with template matching, the proposed approach is more robust to noise.

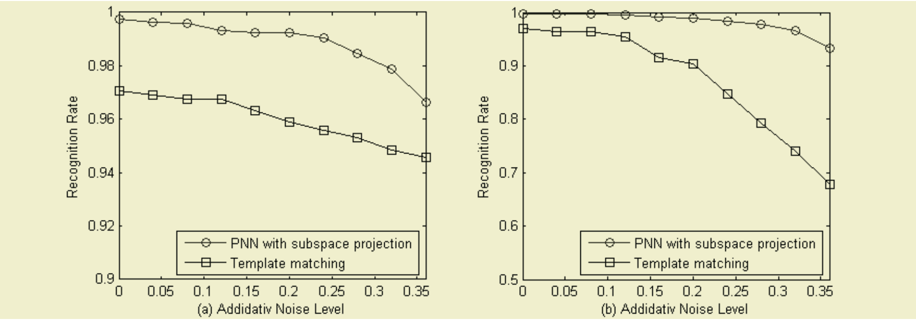


Fig. 5. Recognition rate versus additive noise level. (a) Only testing images corrupted by noise. (b) Both training and testing images corrupted by noise

6 Conclusion

In this paper, a new robust approach for license plate recognition is proposed on the basis of subspace projection and probabilistic neural network. The experiment results illustrate the effectiveness of the proposed method. The rate of recognition is obviously better than that of the conventional template matching method.

The next work is to extend this approach to other fields of image recognition.

References

1. Naito, T., Tsukada, T., Yamada, K., Kozuka, K., Yamamoto, S.: Robust License-plate Recognition Method for Passing Vehicles under Outside Environment. IEEE Trans. Vehicular Technology, **496** (2000) 2309–2319
2. Ahmed, M.J., Sarfraz, M., Zidouri, A., Al-Khatib, W.G.: License Plate Recognition System. Proc. IEEE Int. Electronics, Circuits and Systems, **2** (2003) 898–901
3. Xianda, Zhang: Matrix Analysis and Applications. Tsinghua University Press, Beijing (2004)
4. Specht, D.F.,: Probabilistic Neural Network. Neural Networks, **3** (1990) 109–118
5. Specht, D. F., Shapiro, P. D.: Generalization Accuracy of Probabilistic Neural Networks Compared with Backpropogation Networks. Proc. IJCNN'**91** (1991) 458–461
6. Chan, F.H.Y., Lam, F.K., Hui, Zhu: Adaptive Thresholding by Variational Method. IEEE Trans. Image Processing, **7** (1998) 468–473

Automatic Authentication Technique Based on Supervised ART-2 and Polynomial Spline Pyramid Algorithm

Ning Chen[1], Boqin Feng[1], Haixiao Wang[2], and Hao Zhang[3]

[1] Department of Computer Science and Technology
Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China
chenn_victor@eyou.com

[2] 29 Bayowski Road West Orange, NJ 07052, USA
jwang_nj@yahoo.com

[3] Shenzhen Traffic Control Center, Shenzhen, Guangdong 518000, China

Abstract. This paper introduced a technique for authenticating the vehicle engines by comparing the images of the imprints of the identification number acquired when the vehicle was first registered and the ones acquired from the routine yearly vehicle inspection. The images are taken by rubbing a pencil over a piece of paper covered over the images and then are scanned into a computer. Due to the nature of the acquiring technique, the acquired images have lots of artifacts caused by the shape and the condition of the engine surface and unevenness of rubbing the pencils by hand. We used the polynomial spline pyramid algorithm to acquire a training set using ART-2, which is considered a tradeoff of stability-plasticity dilemma. The experiments show an accuracy rate close to 80%.

1 Introduction

Many techniques were developed for authentication against forgeries. Digital techniques utilizing cheaper and cheaper computation cost, such as digital watermarking technologies, are studied extensively in the passing decade. In a world prevailed with computer-based technologies, many traditional ones are still of great values. For example, labels with bar codes are used for managing commodities sold in stores and imprints (i.e. countermarks) are used to track individual piece of equipments or machineries. To authenticate the labels and imprints now largely relies on discerning eyes of well-trained personals. Presently in China, the vehicle management bureau employs a method of checking the genuineness of a vehicle by comparing the images of countermark imprinted on the engine. Automation of this process would bring a huge benefit socially and commercially by improving the efficiencies, considering the large amount of vehicles. Various techniques have been tried in the past [1],[2].

There are many learning models in the pattern recognition field, which can be used to implement some kinds of automatic authentication task; however, they normally require a training set. Some of the sets are fairly large. However, in our case, we

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 222–225, 2005.

only have a single sample that can be used for authentication. Support vector Machine (SVM) based on Statistical Learning Theory (SLT) is a machine learning strategy for a finite set of samples, it is not able to handle the circumstance of single sample. The Adaptive Resonance Theory 2 Network (ART-2) was introduced by Carpenter and Grossberg in [3], whose architecture was based on the idea of adaptive resonant feedback between two layers of nodes, thus it can extract common character from a series of coarse-to-fine subimages transformed from a single example by the polynomial spline pyramid algorithm to attain convergence of the algorithm.

In this paper, we present a computational technique for analyzing images taken by rubbing pencil on a piece of paper on countermark imprinted on rigid body surfaces. The ART-2 was chosen as authentication process, a neural network based on the modeling of interplay between long-term and short-term memory in a biological system, which can deal with tradeoff of stability-plasticity dilemma [4]. A supervised version of the ART-2 is used in this paper, whose advantage when using a supervised neural network is that we always know which features describe which object that we need. But a supervised ART-2 requires training set to learn, we have only a standard image taken when the care was initially registered. So we make use of the polynomial spline pyramid algorithm to produce an image pyramid representing image at multiple resolutions [5]. It provides a coarse-to-fine strategy to improve the execution speed and convergence properties of algorithms, and is in accordance with theory of human vision. Because key features of the images are edges, we use edge detection algorithm first to transfer the gray scale images to binary images. Then, we produce a vector set using HOUGH transformation and provide it with ART-2 as input. The novel technique eliminates the need of a training set and provides an adaptive optimization of threshold selection in supervised ART-2.

2 Key Technique

The key techniques include the polynomial spline pyramid algorithm and the supervised ART-2. Kamal R. Al-Rawi etc. presented a supervised ART-2 algorithm [6] in 1999, but it should be employed when a large number of committed nodes (>1000) are expected. The supervised ART-2 algorithm presented in this paper, based on improvement on algorithm [7], can work adaptively under the circumstances of a small training set.

2.1 Polynomial Spline Pyramid Algorithm

The polynomial spline pyramid algorithm generates a class polynomial spline pyramid, which are piecewise polynomials with order n and its derivatives up to order n-1. It has more efficiency than Gaussian algorithm and asymptotic approximation to Gaussian function. The algorithm provides two operations, REDUCE and EXPAND. We transform rubbing by REDUCE to acquire a series of low-resolution image. REDUCE procedure is given in Fig.1. (See [5] for details)

$$g_1^n(k) \rightarrow (b^n)^{-1} \rightarrow b^{2n+1} \rightarrow a_0(k)$$
$$a_{(i)}(k) \rightarrow \frac{1}{2} u_2^n \rightarrow \downarrow 2 \rightarrow a_{(i+1)}(k)$$

Fig. 1. REDUCE procedure

2.2 Supervised ART-2

The ART-2 network consists of two layers (short term memory), weights connecting the two layers (long term memory), and a vigilance testing factor to control the closeness of the groups to one another. The F1 layer receives the inputs and through a short term memory process obtains a result from the F2 layer. We provide input for ART-2 as training set. Because training set from the polynomial spline pyramid algorithm belong to a class, it can adjust vigilance testing factor to make output according to real requirement. The program is iteration cycle until finishing adjustment. So, we modify ART-2 model to fit supervised algorithm.

ART-2 would compare the result from the short-term memory to long-term memory via the weights and places the input vector into a category. We modify comparison part in ART-2, i.e. we append a cache to the part. When supervised ART-2 is learning, if classification is error, then adjusting vigilance testing factor, computing again, thus vigilance testing factor would be an appropriate value.

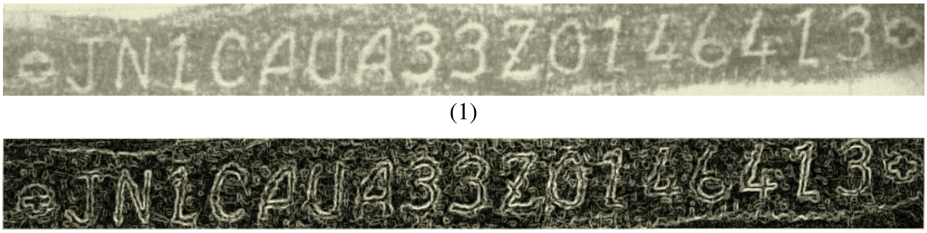
3 Authentication Algorithm

Combined with HOUGH transform, we can acquire an Authentication Algorithm detailed as follow:

Algorithm: Supervised ART-2 Algorithm
Stage I: Learning program
Step1: Transform rubbing by Polynomial Spline Pyramid Algorithm to acquire a series of low-resolution images;
Step2: Edge detection on every low-resolution image using SOBEL operation;
Step3: HOUGH transforms on image in Step2 to acquire Vector set including coordinates and number of accumulation point in the line;
Step4: Supervised ART-2 has two output, one is for class we expect, another is for error class, and has n×n input according to vector set;
Stage II: Authentication program
Step5: Edge detection on image of rubbing for authentication using SOBEL operation;
Step6: HOUGH transform;
Step7: Input resulting above into supervised ART-2.

4 Experiment Result

The algorithms were coded in java and MATLAB. The key resulting images are shown in Fig.2.



(1)

(2)

Fig. 2. (1) Standard rubbing. (2) Resulting of edge detection

We tested supervised ART-2 with 150 authentication examples, and it had 31 mismatches. The supervised ART-2 network attained an accuracy of 79.3%.

5 Conclusions and Directions for Future Research

In this paper we proposed to use a supervised ART-2 combined with the polynomial spline pyramid algorithm to authenticate engine imprints. The experiments show an accuracy of 79.3%, which is not perfect but can potentially be used to pre-screen the vast amount of images to be verified.

We would improve the technique in these sections: modification of algorithm of comparisons in supervised ART-2, such as using Hausdorff distance and appending algorithm to remove noise etc.

References

1. Lyu, S., Rockmore, D., Farid, H.: A Digital Technique for Art Authentication. PNAS, **101** (2004) 17006-17010
2. Gestal, M., Gómez-Carracedo, M.P., Andrade, J.M., Dorado, J., Fernández, E., Prada, D., Pazos, A.: Classification of Apple Beverages Using Artificial Neural Networks with Previous Variable Selection. Analytica Chimica Acta, **524** (2004) 225–234
3. Carpenter, G.A., Grossberg, S.: ART 2: Self-Organization of Stable Category Recognition Codes for Analog Input Patterns. Applied Optics, **26** (1987) 4919-4930
4. Francis, G.: The Stability-Plasticity Dilemma in Competitive Learning. Department of Psychology Sciences. Purdue University. Technical Report (1996)
5. Michael Unser, Akram Aldroubi, Murray Eden: The L2 Polynomial Spline Pyramid. IEEE Transactions on Pattern Analysis and Machine Intelligence, **15** (1993)
6. Kamal, R., Al-Rawi, Consuelo Gonzalo, and Aqueda Arquero: SUPERVISED ART-II: A New Neural Network Architecture, With Quicker Learning Algorithm, For Learning and Classifying Multivalued Input Patterns. ESANN'1999 Proceedings - European Symposium on Artificial Neural Networks (1999) 289-294
7. Carpenter, G.A.: Default ARTMAP. Proceedings of the International Joint Conference on Neural Networks, (2003) Technical Report CAS/CNS TR-2003-008

Neural Network Based Online Feature Selection for Vehicle Tracking

Tie Liu, Nanning Zheng, and Hong Cheng

the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
Xi'an, Shaanxi 710049, China
{tliu,hcheng}@aiar.xjtu.edu.cn
nnzheng@mail.xjtu.edu.cn

Abstract. Aiming at vehicle tracking with a single moving camera for autonomous driving, this paper presents a strategy of online feature selection combined with related process framework. Detected vehicle can provide more information for tracking. A principal component analysis neural network is used to select appearance features online. Then the positive and negative histogram models using selected features are found for the detected vehicle and the surroundings. A likelihood function is defined based on histogram models, and it can be used as a simple classifier. For selected multiple features, the corresponding multiple classifiers are combined with a single layer perceptron. Experimental results indicate the validity and real-time performance.

1 Introduction

Vision based vehicle tracking with a single moving camera is a critical task for real application of autonomous driving. Direct method to get the distance is the stereo system [1]. But a single camera with precise calibration [2] is easier and cheaper. Feature extraction is the most difficult task because of varieties in color, shape, size and views. Broggi [2] and other researchers developed different efficient shape based features for vehicle detection. More general methods are based on statistical learning to extract features from thousands of candidates through large numbers of samples. These methods can improve the performance for vehicle detection, but are complicated. For vehicle tracking, real-time performance is considered more, and more information can be used. The problem is how to select these features on-line for given vehicle, and keep the generalization of the algorithm at the same time.

For detected vehicle, appearance based features are much useful. Some online selection algorithms have been presented for this aim, and R. Collins [3] presented an online selection of discriminative tracking features. This method mainly process monotone background. For autonomous driving, background is much more complex and changes frequently. Make the algorithm to adapt the changing target and environment is the key problem of this paper.

The following section is the problem and setup. Section 3 is about the algorithm, and online feature selection strategy and related combination method are emphasized. Section 4 is the experiments, and conclusion is summarized in the last section.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 226–231, 2005.

2 Problem and Setup

Rear-view vehicle tracking with a single moving camera is considered here. Feature extract is one of the most important tasks. There are some universal features, such as symmetry, have been proved useful [2], but difficult to select. For autonomous driving, there are more difficulties: varied background, time-changed illumination and terrible disturbances such occlusions, shadows and so on. All these factors make the universal features disabled oftentimes. Especially, appearance based features are much useful for given target. Color is the most obvious feature for human eyes. Some vehicle images with different color are showed in figure 1.

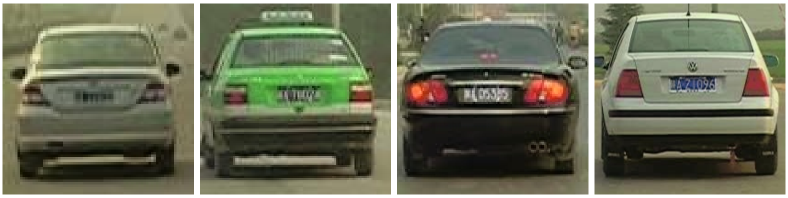


Fig. 1. Rear-view vehicle images with different color features

There are distinct differences between detection and tracking. For vehicle detection, universal features must be extracted from thousands of samples to promise that all possible vehicles are detected. Shape based features are selected usually. For vehicle tracking, more efficient features for the detected vehicle can be used, and these features are related with the special vehicle. The aim of this paper is to select features for the different detected vehicles with the different driving environment, and keep the generality of the selection strategy at the same time.

3 Algorithm Description

Vehicle tracking is based on vehicle detection. Shape based features are selected through learning, and a strong cascaded classifier is used to detect vehicles. Then the detected vehicle can be tracked using more useful and simple features with high robustness and efficiency. The online feature selection strategy for tracking is emphasized in this paper.

Detected vehicles in initial sequential images are inputted as positive samples. A principal component analysis (PCA) neural network (NN) is used to select the color features online. Based on the selected optimal features, simple classifiers are constructed. A single layer perceptron is used to combine multiple classifiers then. The algorithm framework is showed as figure 2.

The online feature selection strategy with PCA NN and the simple classifier using the selected features are introduced in details.

3.1 Online Feature Selection for Tracking

In tracking phase, vehicles and environments are given. How to extract the most efficient features is presented in this section. Color is a good feature for segmentation [4].

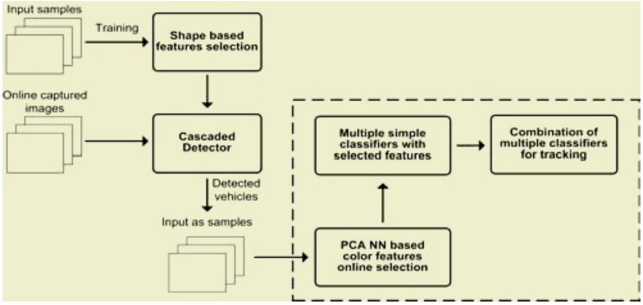

Input samples
Training
Shape based features selection
Online captured images
Cascaded Detector
Detected vehicles
Input as samples
Multiple simple classifiers with selected features
Combination of multiple classifiers for tracking
PCA NN based color features online selection

Fig. 2. Framework of the whole algorithm, and the part marked with broken line is the main contents of this paper. A PCA NN is used to select color features online, and detected vehicles are inputted to learn histogram models which construct a simple classifier. A single layer perceptron is used to combine multiple classifiers

Different color space can be used such as RGB, HSI, CIE, YUV, L*u*v, and so on [5]. Feature selection for tracking is based on the supposition: colors of any given vehicle cluster on a small range that is distinct from environments. This supposition is applicable for most cases.

PCA NN is used to find color features with most optimal discrimnability for varied environment, and it has been used for adaptive color reduction successfully [6]. An important feature of neural networks is the ability to learn from their environment, and PCA is very suited for feature selection [8]. A feedforward network with a single layer of computation nodes is used to perform feature selection. The structure of the PCA NN is showed as figure 3.

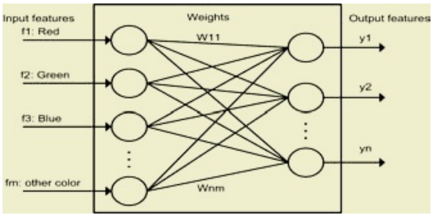


Fig. 3. The structure of PCA NN. The inputs are features with different color representation, and the outputs are the features with the high disciminability

The network is trained using the Generalized Hebbian Algorithm [9], and is implemented via the rules

$$
\begin{aligned}
&\Delta\omega_{j,i}(n)=\eta y_j(n)f_i(n)-\eta y_j(n)\sum_{k=0}^{j}\omega_{k,i}(n)y_k(n)\\
&y_j(n)=\sum_{i=0}^{m}\omega_{j,i}(n)y_j(n),\ j=1\cdots n\\
&\omega_{j,i}(n+1)=\omega_{j,i}(n)+\Delta\omega_{j,i}(n), i=1\cdots m,\ j=1\cdots n
\end{aligned}
\tag{1}
$$

where $\Delta\omega_{j,i}(n)$ is the change applied to the synaptic weight $\omega_{j,i}(n)$ at time n, and η is the learning rate parameter.

Different linear or nonlinear combinations of RGB are inputted as features, and the detected vehicle image and surrounding environmental image are used to train the weights. Outputted features are with optimal performance for distinguishing the current target and background.

3.2 Simple Classifier with Selected Features

PCA NN is used to extract efficient features with high discrimnability. But how to describe a vehicle with such features is still not mentioned. Histogram distribution is a good model for describing a target. For tracking, different histogram models for vehicle and environment can be set up steadily because of environmental continuity.

Detected vehicles and surrounding background are inputted as samples for training models. The statistical histogram distribution is set up in initial moment of tracking. The log likelihood using histogram is defined by [3]. Illumined by this work, we define the likelihood function for each selected feature

$$L(i)=\frac{t(i)-b(i)}{\max(t(i)+b(i),\delta)},i=0,l-1 \tag{2}$$

where $t(i)$ and $b(i)$ are the histogram distributions of target and background separately, δ is a small value (we set it to 0.001) the prevents dividing error, l is the length of histogram which can choose the precision of discernment.

If $L(i)$ is bigger than zero, it means that the pixels with feature's value i belongs to target with the probability corresponding to the absolute value of $L(i)$, and vice versa. A likelihood function for the selected feature can be found to segment inputted images in the following tracking. The process is showed as figure 4.

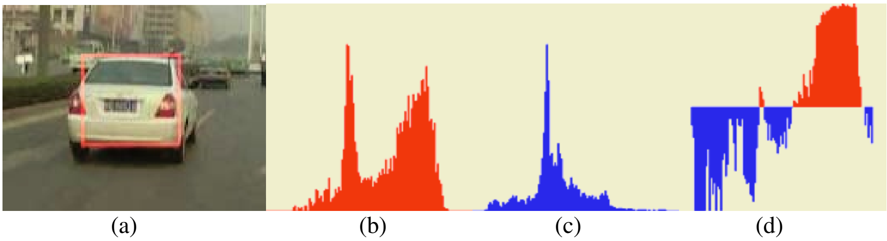


Fig. 4. Histogram of target and background and the likelihood function. (a) is the original image used for learning, where the area marked by red rectangle is inputted as the target, and the surrounding area is inputted as the background. (b), (c) and (d) are computed using the most optimal feature. (b) is the histogram of the selected vehicle. (c) is the histogram of the background. (d) is the likelihood function computed from (b) and (c), and vertical axis is the likelihood value, horizontal axis is the feature value

The likelihood function can be binarised to judge yes or no, ignoring the absolute value which scale the probability of the decision. For real application, this process can improve efficiency. Segmented image is searched with a window of fixed size, and each window is judgeed as target or non-target. The rule of judgment is the rate of target points. The segmented image can be considered as the classifier activation map directly. Then a simple classifier is realized with high efficiency.

4 Experiments

Different videos are captured with a single color camera built in our experimental car. Based on the detector, the online feature selection strategy for tracking is presented and realized on sequential images. In this paper, the vehicles are marked out by hand, and the background samples are selected randomly. Experimental process related with this paper is as the following:

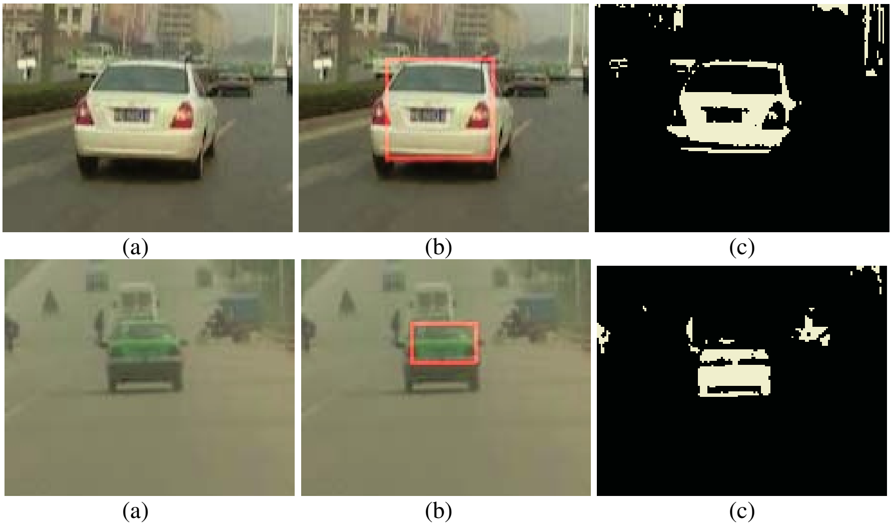
a) Sequential images including detected vehicles are inputted to train the PCA NN, which can select optimal features from different color representations.

b) Marked vehicles in sequential images are inputted to get positive histogram distribution, and surrounding areas are selected randomly to get negative histogram distribution. Then the likelihood function is computed using the positive and negative models, and a simple classifier is constructed with the likelihood function.

c) Sequential images are inputted to multiple classifiers using online selected features. Complicated detector mentioned in the framework is the supervisor. A combination network using a single layer perceptron is trained.

d) Then a complete multiple classifier system is constructed, and more sequential images are tested on this framework.

In our experiments, smaller image including detected vehicle replaces the whole captured image. Only red, green, blue, normalized red, green, blue and gray are inputted to PCA NN. And two optimal features are selected to compute the histogram distribution with 128 scales. More than ten groups of sequential images are tested, and figure 5 shows some experimental results.



(a) (b) (c)

(a) (b) (c)

Fig. 5. Experimental results (above: red vehicle; below: green vehicle). (a) is original images. Selected area with red rectangle in (b) is inputted as vehicle sample. (c) is the segmented results using the defined likelihood function

For sequential images including vehicles with distinct color, we can get good experimental results. When more disturbances with similar colors move into views, more false positive targets are detected. Two detailed aspects can be improved with the framework presented in our paper: a) more features can be as the input of PCA NN, and other appearance based features can be tested also; b) more efficient definitions of the likelihood function can be tested.

5 Conclusion

Aiming at the problem of feature selection for vehicle tracking, a strategy of online feature selection combined with related process framework is presented in this paper. PCA NN is used to select color features for the detected vehicle. Then the likelihood function using the histogram distribution corresponding to each selected feature is defined and a simple classifier is constructed. For multiple features, the corresponding multiple classifiers are combined using a single layer perceptron. Experimental results indicate the validity and real-time performance of the presented online feature selection strategy.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 60021302 and 60205001.

References

1. Bertozzi, M., Broggi, A., Fascioli, A., , Nichele, S.: Stereo Vision-Base Vehicle Detection. In: IEEE Intelligent Vehicles Symposium 2000, (Detroit, USA) (2000) 39-44
2. Broggi, A., Cerri, P., Antonello, P.C.: Multi-Resolution Vehicle Detection Using Artificial Vision. In: IEEE Intelligent Vehicle Symposium 2004, Parma, Italy, (2004) 14-17
3. Collins, R.T., Liu, Y.: On-Line Selection of Discriminative Tracking Features. In: Proceedings of the Ninth IEEE International Conference on Computer (2003) 346-352
4. Tsin, Y., Collins, R.T., Ramesh, V., Kanade, T.: Bayesian Color Constancy for Outdoor Object Recognition. In: Proceedings of the IEEE 2001 Conference on Computer Vision and Pattern Recognition, **1** (2001) 1132-1139
5. Cheng, H.D., Jiang, X.H., Sun, Y., Wang, J.: Color Image Segmentation: Advances and Prospects. Pattern Recognition, **34** (2001) 2259-2281
6. Papamarkos, N., Atsalakis, A.E., Strouthopoulos, C.P.: Adaptive Color Reduction. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, **32** (2002)
7. Meireles, M.R.G., Almeida, P.E.M., Simoes, M.G.: A Comprehensive Review for Industrial Applicability of Artificial Neural Networks. IEEE Transactions on Industrial Electronic, **50** (2003)
8. Haykin, S.: Neural Networks: A Comprehensive Foundation, Second Edition. Prentice-Hall (1999)

TextCC: New Feed Forward Neural Network for Classifying Documents Instantly*

Zhenya Zhang[1,3], Shuguang Zhang[2], Enhong Chen[3],
Xufa Wang[3], and Hongmei Cheng[4]

[1] Electronic Engineering & Information Science Department
University of Science and Technology of China (USTC), Hefei 230027, China
zhenyazhang@ustc.edu
http://mail.ustc.edu.cn/~zzychm

[2] Statistics & Finance Department of USTC, Hefei 230027, China
sgzhang@ustc.edu.cn

[3] Computer Science Department of USTC, Hefei 230027, China
{zzychm,chamos}@mail.ustc.edu.cn, xfwang@ustc.edu.cn

[4] Mathematics Department of Anhui Normal University, Wuhu 241000, China
zzychm@mail.ahwhptt.net.cn

Abstract. Corner classification (CC) network is a kind of feed forward neural network for instantly document classification. To classify text object instantly, new training algorithm, named as TextCC, for feed forward neural network is presented in this paper. To give a solution for multi-corner judging, new training algorithm for the construction of weight matrix of output layer of CC is given. Experimental results show that TextCC can work well and the precision of TextCC is higher than CC4's.

1 Introduction

With the explosive growth of information resources available on the WWW, Web search engine becomes a kind of important tools for user to find desired information available on WWW [1]. Although search engine can retrieve and recommend web pages which match user's query request, information desired by a user indeed is only very small proportion of those entire recommendations usually. To help user grasp desired information from the Web more efficiently, classification instantly for text sets is focused by lots of research [1].

In the process for classifying a group of text objects, it is very universal that the cosine similarity of two text objects is be used to judge whether those two text objects are in same class. Suppose TF$_1$=(tf_{11},tf_{12}…tf_{1n}) and TF$_2$=(tf_{21},tf_{22}…tf_{2n}) be TF vectors for text objects T$_1$ and T$_2$.The cosine similarity between T$_1$ and T$_2$ is denoted as Sim(T$_1$,T$_2$). Sim(T$_1$,T$_2$) = Cosine(TF$_1$,TF$_2$) = $\frac{TF_1 \cdot TF_2}{\|TF_1\|\|TF_2\|}$. If Sim(T$_1$,T$_2$) $\geq r$, $r \in$ [0,1], T$_1$

[*] This paper was granted financial support from "21st Century Education Revitalization Project" and "China Postdoctoral Science Foundation (2004036463)".

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 232–237, 2005.

and T_2 should be in same class. Otherwise T_1 and T_2 are in different classes. Here *r* is threshold.

Usually, each represented vector of a text object can be normalized. If $V=(v_1,v_2 \ldots v_n)$ is a vector and $\|V\| \neq 0$, $V^{'}=(\frac{v_1}{\|V\|},\frac{v_2}{\|V\|}\cdots\frac{v_n}{\|V\|})$ is the normalized form of V, here $\|V\|=\sqrt{\sum_{i=1}^{n} v_i^2}$ .It is obvious that the cosine similarity of two objects is equal to the inner product of those two represented vectors if those two vectors are normalized.

CC4 [1], [2], [3], [4], [5], [6] is a feed forward neural network for instantly classification. CC4 is derived from corner classification (CC) algorithm [2],[3],[4],[5],[6]. Each sample data for CC is considered as one point in a multi-dimension space and the neighborhood of the data is one corner of the space. If the point corresponding to a probe data is in the corner of a sample data, the probe data is in same class with the sample data.

When one query result is represented with a TF vector by CC4, the TF vector is abbreviated to a binary vector [1], [2]. That is a latent drawback of CC4. To increase the classification precision of feed forward neural network CC4 like, it is necessary for it to accept real vector as inputs. Through the L-discretization of a real vector, ExtentCC4 [2] can accept real vector as inputs. Because the time complexity and space complexity are increased remarkably, ExtendCC4 only can be used at the case with time requirement loosely [2].

This paper is organized as follows. New training algorithm, named as TextCC, is presented for classifying text objects instantly with forward neural network in session2. To judge whether a point is in more than one corner, new training algorithm for corner classification is presented in sesction3. Experimental results are shown at session4. Conclusion is given in sesction5.

2 TextCC for Classifying Text Object Instantly

TextCC is a three-layered feed forward neural network. Each training sample is required to be presented to the network only once. A TextCC network can map one input real vector X to one output binary vector Y. The topology of the network is shown at fig1. Neurons in input layer are sensors. The transfer function *f*(*x*) of each neuron in hidden and output layer is defined at equation (1). Here x is the input signal for a neuron.

$$y = f(x) = \begin{cases} 1 & x > 0 \\ 0 & x \le 0 \end{cases} \tag{1}$$

Suppose $V=(v_1,v_2 \ldots v_k)$, $f(V)=(f(v_1),f(v_2)\ldots f(v_k))$. If the dimension of vector presented to TextCC as input is N, there are N+1 neurons in input layer of TextCC. The input for the (N+1)th neuron in input layer is 1 and input for each neuron of other N input neurons is the value of one component of the input vector orderly. The number of hidden neurons is equal to the number of training samples. The number of output neurons is equal to the dimension of target vectors.

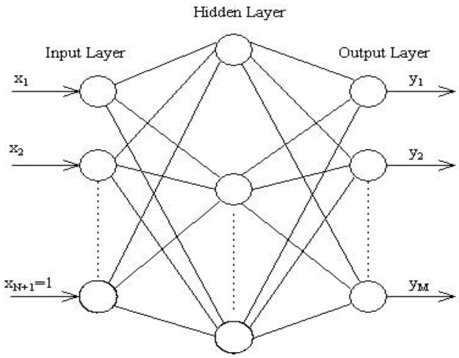


Fig. 1. The topology of TextCC network

Definition 1. Suppose X=(x_1, x_2…x_n) be an n dimension vector. And $x_i \in \{0,1\}$, $i = 1,2...n$. X is an n-dimension binary vector.

Definition 2. Let X=(x_1, x_2…x_n) be an n-dimension binary vector. X'=(x_1', x_2'…x_n') is the deriving vector for X. Here x_i' (i=1,2…n) is given at (2).

$$x_i' = \begin{cases} 1 & \text{if } x_i = 1 \\ -1 & \text{if } x_i = 0 \end{cases} \tag{2}$$

Usually, the weight matrix of hidden layer is denoted as W and the weight matrix of output layer is denoted as U. The task for a TextCC's learning is to construct W and U.

Let W_1, W_2…W_H be all N dimension sample vectors for TextCC training. W can be constructed as (3). In(3), r is parameters named as generalized radius and is a positive infinitesimal.

$$W = \begin{pmatrix} W_1^T & W_2^T & \cdots & W_H^T \\ -r+\varepsilon & -r+\varepsilon & \cdots & -r+\varepsilon \end{pmatrix} \tag{3}$$

Let U_i be the target binary vector for W_i, (i=1,2…H). U can be constructed as (4) where U_i' is the deriving vector of U_i (i=1,2…H).

$$U = \begin{pmatrix} U_1' \\ U_2' \\ \vdots \\ U_H' \end{pmatrix} \tag{4}$$

If a probe vector X=(x_1,x_2…x_N) is presented to a TextCC, Y, the output vector of the TextCC is computed according to equation (5).

$$Y=f(f((X,1)\times W)\times U)=f(f((x_1,x_2\ldots x_N,1)\times W)\times U) \tag{5}$$

Lemma 1. Let W_1, $W_2 \ldots W_H$ be all N dimension vectors for TextCC training. $X=(x_1,x_2 \ldots x_N)$ is a probe vector when TextCC is trained well. If X is in the corner represented by W_i, the output of *ith* neuron in hidden layer is 1 and others are 0.

Lemma 2. Let U_i be the target vectors for W_i, (i=1,2…H). W_1, $W_2 \ldots W_H$ be all sample vectors for TextCC network training. Le X be a probe vector for TextCC network trained with W_1, $W_2 \ldots W_H$ and U_1, $U_2 \ldots U_H$. Suppose X is in the corner represented by W_i, here $1 \leq i \leq H$ and X is not in other corners represented by W_j, here $1 \leq j \leq H$ and $j \neq i$. The output of TextCC network is U_i.

Suppose W is the weight matrix of hidden layer and U is the weight matrix of output layer. The task for the training algorithm of TextCC is to construct those two matrixes. According to lemma1, W is to be used for judging which corner the input vector is in. And according to lemma2, U is to be used for computing the code of the corner where the input vector is in.

3 New Training Algorithm for Multi-corner Judging

Lemma 3. Let U_1, $U_2 \ldots U_H$ be target vectors for W_1, $W_2 \ldots W_H$, here W_1, $W_2 \ldots W_H$, be all sample vectors for TextCC network training. Let X be one probe vector for TextCC network trained with W_1, $W_2 \ldots W_H$, and U_1, $U_2 \ldots U_H$. Suppose X is in corners represented by $W_{i_1}, W_{i_2} \cdots W_{i_k}$, here $1 \leq k, i_1, i_2 \ldots i_k \leq H$ and X is not in other corners represented by W_j, here $1 \leq j \leq H$ and $j \notin \{W_{i_1}, W_{i_2} \cdots W_{i_k}\}$. The output of the TextCC network is $f(U_{i_1}^{'} + U_{i_2}^{'} + \cdots + U_{i_k}^{'})$.

Usually, it is impossible to distinguish $U_{i_1}, U_{i_2} \cdots U_{i_k}$ from $f(U_{i_1}^{'} + U_{i_2}^{'} + \cdots + U_{i_k}^{'})$.

Proposition 1. Let E_i be target binary vector for W_i where the value for *ith* component of E_i is 1 and others are 0, here W_i be a sample vector for TextCC network training and i=1,2…H. Let U be the weight matrix of output layer. Suppose $U = \begin{pmatrix} E_1 \\ E_2 \\ \vdots \\ E_H \end{pmatrix}$.

Suppose X is in corners represented by $W_{i_1}, W_{i_2} \cdots W_{i_k}$, here $1 \leq k, i_1, i_2 \ldots i_k \leq H$ and X is not in other corner represented by W_j, here $1 \leq j \leq H$ and $j \notin \{i_1, i_2 \cdots i_k\}$. The output of the TextCC network is $f(E_{i_1} + E_{i_2} + \cdots + E_{i_k}) = E_{i_1} + E_{i_2} + \cdots + E_{i_k}$.

If U, the weight matrix of output layer of TextCC is constructed after proposition1, it is very funny that the number of 1 in the output vector of TextCC is the corner's number that the probe vector is in.

4 Experimental Results

CC4 and TextCC are two feed forward neural network for instantly classification. In our experiments, performances of those two networks on text classification are tested with data set download from **http://kdd.ics.uci.edu**. Text objects in the data set can be divided into 20 groups and there are 1000 articles in each group. Articles in same group are downloaded from same newsgroup in Internet. All tests require that CC4 and TextCC can judge which newsgroup is in when a probe article is submitted. The weight matrix of output layer of CC4 and TextCC are constructed according to proposition1. Articles for CC4 and TextCC training are regular sampled from the data set.

Each article is represented as a normalized TF vector. Words in the dictionary for the TF vector constructing are words in sample articles and noise words are omitted. Each word is stemmed before it is added into the dictionary.

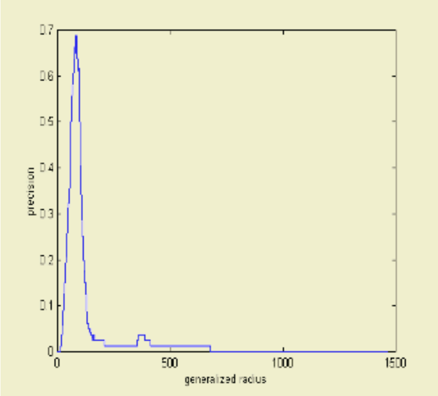


Fig. 2. The influence of radius on precision of CC4

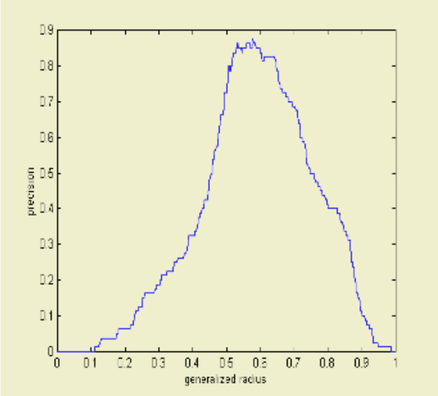


Fig. 3. The influence of radius on precision of TextCC

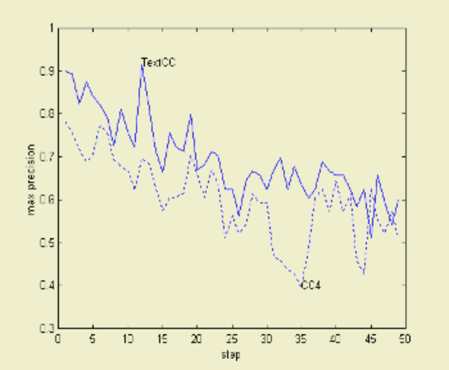


Fig. 4. The precision of CC4 and TextCC (2 corner)

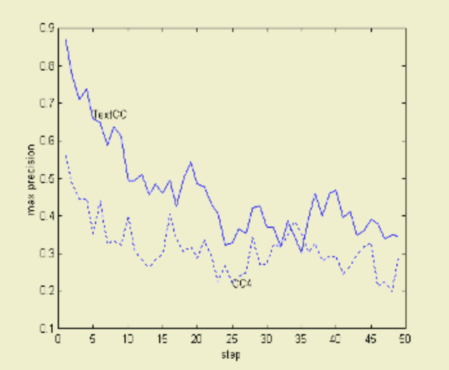


Fig. 5. The precision of CC4 and TextCC (4 corner)

Fig2 and Fig3 show the influence of generalized radius on the classification precision of CC4 and TextCC4. In this experiment, two newsgroups are selected and the sample step is 4. It is obvious that the precision of CC4 and TextCC are sensitive for generalized radius and the peak value of precision of TextCC is higher than CC4's. Fig4 and fig5 give the comparisons on classification precision of CC4 and TextCC. The number of class (corner) is 2 in fig4 and 4 in fig5. In this experiment, the sample ratio is decreased from 50% to 2% via the sample step is increased from 1 to 49. Fig4 and fig5 show that the classification precision of TextCC is increased more than CC4's at most case.

5 Conclusion

TextCC can be used by system on online information retrieval and Chance Discovery validly. The data's type is not considered in the analyses for the time consumed by CC4 and TextCC for same classification task. Because TextCC should be working with data's type as double and CC4 is only working with data's type as integer, it is reasonable that the time consumed by TextCC is less more than CC4's for same classification task. Because TextCC can do classification task instantly as CC4 and the classification precision of TextCC is higher more than CC4's, the pity of TextCC on time consumed can be omitted.

The key for CC algorithms is the definition for the sample data's neighborhood. The definition is depended on domain knowledge of application. New definition for the sample data's neighborhood is an important focus in our future research.

References

1. Shu, B., Kak, S.: A Neural Network-based Intelligent Metasearch Engine. Information Sciences, **1** (1999) 1-11
2. Chen, E., Zhang, Z., Wang, X.: An Extended Corner Classification Neural Network Based Document Classification Approach. Journal of Software, (2002) 871-878
3. Tang, K.W., Kak, S.C.: A New Corner Classification Approach to Neural Network Training. Circuits Systems Signal Processing, **17** (1998) 459-469
4. Raina, P.: Comparison of Learning and Generalization Capabilities of the Kak and the Back Propagation Algorithms. Information Sciences, **81** (1994) pp261-274
5. Kak, S.: New Algorithms for Training Feed Forward Neural Networks. Pattern Recognition Letters, **15** (1994) 295-298
6. Kak, S.: On Generalization by Neural Networks. Information Sciences, **111** (1998) 293-302

A Neural Network Model for Hierarchical Multilingual Text Categorization

Rowena Chau, Chunghsing Yeh, and Kate A. Smith

School of Business Systems
Faculty of Information Technology
Monash University
Clayton, Victoria 3800, Australia
{Rowena.Chau,ChungHsing.Yeh,Kate.Smith}@infotech.monash.edu.au

Abstract. Enabling navigation via a hierarchy of conceptually related multilingual documents constitutes the fundamental support to global knowledge discovery. This requirement of organizing multilingual document by concepts makes the goal of supporting global knowledge discovery a concept-based multilingual text categorization task. In this paper, intelligent methods for enabling concept-based hierarchical multilingual text categorization using neural networks are proposed. First, a universal concept space, encapsulating the semantic knowledge of the relationship between all multilingual terms and concepts, which is required by concept-based multilingual text categorization, is generated using a self-organizing map. Second, a set of concept-based multilingual document categories, which acts as the hierarchical backbone of a browseable multilingual document directory, are generated using a hierarchical clustering algorithm. Third, a concept-based multilingual text classifier is developed using a 3-layer feed-forward neural network to facilitate the concept-based multilingual text categorization.

1 Introduction

The rapid expansion of the World Wide Web throughout the globe means electronically accessible information is now available in an ever-increasing number of languages. In a multilingual environment, one important motive of information seeking is global knowledge discovery. Global knowledge discovery is significant when a user wish to gain an overview of a certain subject area covered by a multilingual document collection before exploiting it. In such a situation, concept navigation is required. The basic idea of concept navigation is to provide the user with a browseable document hierarchy that gives a fair indication of the conceptual distribution of all multilingual documents over the domain. This requirement of organizing multilingual documents by concepts makes the goal of supporting global knowledge discovery a concept-based multilingual text categorization task.

Text categorization is a classification problem of deciding whether a document belongs to a set of pre-specified categories of documents. In a monolingual environment, text categorization is carried out within the framework of the vector space model [16]. In the vector space model, documents are represented as feature vectors in a multi-dimensional space defined by a set of terms occurring in the document

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 238–245, 2005.

collection. To categorize documents, this set of terms become the features of the classification problem. Documents represented by similar feature vectors belong to the same category. However, multilingual text represents a unique challenge to text categorization, due to the *feature incompatibility* problem contributed by the vocabulary mismatch phenomenon across languages. Different languages use different sets of terms to express a set of universal concepts. Hence, documents in different languages are represented by different sets of features in separate feature spaces. This language-specific representation has made multilingual text incomparable. Text categorization methods that rely on shared terms (features) will not work for multilingual text categorization. To overcome this problem, a universal feature space where all multilingual text can be represented in a language-independent way is necessary. Towards this end, a concept-based strategy aiming at unifying all existing feature spaces by discovering a new set of language-independent semantic features is proposed. The basic idea is: given a new universal feature space defined by a set of language-independent concepts, multilingual text can then be uniformly characterized. Consequently, multilingual text categorization can also take place in a language-independent way.

In what follows, the architecture of the neural network model for concept-based hierarchical multilingual text categorization is introduced in Section 2. Then, an unsupervised learning approach for discovering a universal concept space from a training parallel corpus using self-organizing map [10] is presented in Section 3. Following this, the generation of a hierarchy of concept-based multilingual document categories is discussed in Section 4. In Section 5, development of the multilingual text classifier using a 3-layer feed-forward neural network [6] is explained. Finally, a conclusive remark is presented in Section 6.

2 A Neural Network Model for Hierarchical Multilingual Text Categorization

The architecture of the neural network model for the concept-based hierarchical multilingual text categorization is depicted in Figure 1.

First, a set of multilingual index terms is extracted from a parallel corpus, which is a collection of documents and their translated versions in multiple languages. By analysing co-occurrence statistics collected in term vectors of the set, conceptual related multilingual index terms are sorted into clusters (i.e. concepts) as they are organized onto a concept space by the virtue of the self-organizing map algorithm. This represents the crucial step for overcoming the feature incompatibility problem that is unique to a multilingual environment, by grouping multilingual index terms that describe similar concepts in different languages into nodes on the same map. Based on the multilingual semantic knowledge encoded by the concept space, language-independent concept-based document vectors representing multilingual documents are generated. This is achieved by indexing multilingual documents with the language-independent concepts in place of their original language-specific index terms.

Second, using the concept-based document vectors of all training parallel documents as inputs to a hierarchical clustering algorithm, a hierarchy of concept-based multilingual document categories is generated.

Third, given the multilingual documents which have been clustered to respective document categories as training data, a concept-based multilingual text classifier is developed by training a 3-layer feed-forward neural network. This trained neural network is then used to categorize any unseen multilingual documents. By associating each category with a list of relevant multilingual documents, the resulting document hierarchy will provide a contextual overview of the document collection. Thus, it can be used as a browseable document directory in user-machine interaction. By enabling navigation via a hierarchy associating conceptually relevant multilingual documents, global knowledge discovery is thus facilitated.

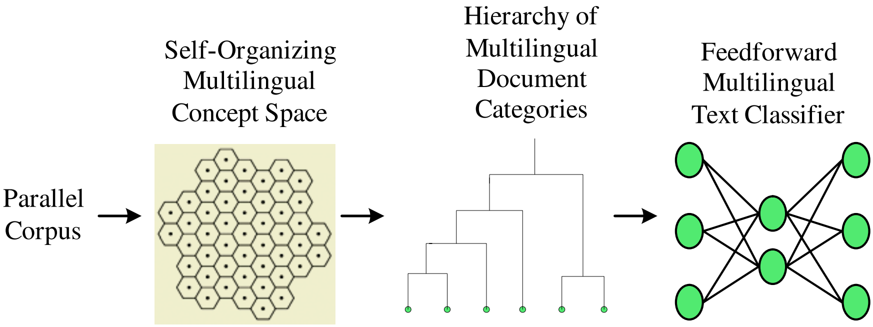

Parallel Corpus
Self-Organizing Multilingual Concept Space
Hierarchy of Multilingual Document Categories
Feedforward Multilingual Text Classifier

Fig. 1. Architecture of the neural network model for concept-based hierarchical multilingual text categorization

3 Concept Discovery and Document Representation Using Self-organizing Map

Self-organizing map, which is an unsupervised neural network, is applied to discover the multilingual semantic knowledge by generating a universal concept space from a set of multilingual terms, extracted from a parallel corpus. This concept space is formed by grouping semantically related multilingual terms into concepts, thus revealing the multilingual semantic knowledge of the relationship between all multilingual terms and concepts. The self-organizing map has been successfully applied to organize textual data in the monolingual environment [3],[7],[8],[12]. Our research work attempts to extend the use of self-organizing maps to process multilingual text.

For constructing the self-organizing multilingual concept space, co-occurrence statistics of multilingual index terms across a parallel corpus are used to determine groups of semantically or conceptually related multilingual terms. To acquire these co-occurrence statistics, corpus processing techniques commonly used for automatic thesaurus development [4],[9],[13] is applied. Analysis of co-occurrence statistics obtained from parallel corpus has been proven to be effective for inferring the most likely translation of terms between languages in the corpus [2],[5],[11].

Given a parallel corpus, meaningful terms from every language covered by the corpus are extracted. They form the set of multilingual index terms for constructing the self-organizing multilingual concept space. Each term is represented by an *n*-

dimensional term vector. Each feature value of the term vector corresponds to the weight of the n^{th} document indicating the significance of that document in characterizing the meaning of the term. Parallel documents which are translated versions of one another within the parallel corpus, are considered as the same feature. To determine the significance of each document in characterizing the contextual content of a term based on the term's occurrences, the standard TF.IDF weighting scheme [16] is used.

Let $\mathbf{x}_i \in R^N$, for $1 \leq i \leq M$, be the term vector of the i^{th} multilingual term, where N is the number of documents existing in the parallel corpus for a single language (i.e. the total number of documents existing in the parallel corpus divided by the number of languages the corpus supports) and M is the total number of multilingual terms. Using these term vectors as the training inputs to the map, the self-organizing map algorithm is applied to form a concept space. The map consists of a regular grid of nodes. Each node is associated with an N-dimensional model vector. Let $\mathbf{m}_j = \left[m_{jn} \middle| 1 \leq n \leq N\right]$, for $1 \leq j \leq G$, be the model vector of the j^{th} node on the map. The algorithm for forming the concept space is given as follows:

Step 1: Select a training multilingual term vector $\mathbf{x}_i$ at random.

Step 2: Find the winning node s on the map with the vector $\mathbf{m}_s$ which is closest to $\mathbf{x}_\mathrm{i}$ such that

$$\left\|\mathbf{x}_i - \mathbf{m}_s\right\| = \min_j \left\|\mathbf{x}_i - \mathbf{m}_j\right\| . \qquad (1)$$

Step 3: After the winning node s is selected, update the weight of every node in the neighbourhood of node s by

$$\mathbf{m}_t^{new} = \mathbf{m}_t^{old} + \alpha(t)(\mathbf{x}_i - \mathbf{m}_t^{old}) . \qquad (2)$$

where $\alpha(t)$ is the gain term at time t ($0 \leq \alpha(t) \leq 1$) that decreases in time and converges to 0.

Step 4: Increase the time stamp t and repeat the training process until it converges.

After the training process, each multilingual term is mapped to a grid node closest to it on the self-organizing map. A self-organizing multilingual concept space is thus formed.

The purpose of constructing the multilingual concept space is to overcome the feature incompatibility problem by generating a set of universal language-independent indexing features accommodating all languages. Given the concept space, semantically related multilingual terms are now organized into clusters (i.e. concepts). These concepts, defined by semantically related multilingual terms, are thus used to index multilingual documents in place of the documents' original language-specific index terms. As such, a concept-based document vector that explicitly expresses the conceptual context of a document regardless of its language can be obtained. The term-based document vector of the vector space model, which suffers from the feature incompatibility problem, can now be replaced with the language-independent concept-based document vector.

To realize this, every multilingual document is indexed by mapping its text, term by term, onto the concept space whereby statistics of its 'hits' on each concept are recorded. This is done by counting the occurrence of each term on the concept space at the node to which that term has been associated. This statistics of concept occurrences can be interpreted as a kind of transformed ‘index’ of the multilingual document to produce a language-independent concept-based document vector. Concept-based document vectors thus obtained, which are language-independent, are essential for enabling concept-based multilingual document clustering. Using these language-independent concept-based document vectors as input to some appropriate clustering algorithm, multilingual documents, which are originally syntactically incomparable can then be grouped based on the conceptual similarity they convey.

4 Generation of a Multilingual Document Hierarchy

A hierarchy of multilingual document categories is generated via a concept-based hierarchical document clustering algorithm. This is realized with the application of the complete linkage hierarchical clustering algorithm [1] using the concept-based document vectors of the training parallel corpus as inputs. The concept-based hierarchical document clustering algorithm is given below.

Step 1: Construct a $K \times Z$ concept-document matrix, *CD*, to represent *K* concepts, $c_{k \in 1,\ldots,K}$, defined by the multilingual concept space, and the parallel corpus, *D*, of *Z* documents, $d_{z \in 1,\ldots,Z}$, such that:

$$CD = \left[w_{kz}\right] \quad k = 1,\ldots,K \; ; \; z = 1,\ldots,Z \, . \tag{3}$$

where w_{kz} is the occurrence of the concept c_k in document d_z

Step 2: Obtain a document association matrix $D \times D$ such that

$$D \times D = \lfloor A_{xy} \rfloor \quad x = 1,\ldots,Z \;\; ; \;\; y = 1,\ldots,Z \, . \tag{4}$$

where A_{xy} is the coefficient of association between each pair of documents $d_{x \in 1,\ldots,Z}, d_{y \in 1,\ldots,Z} \in D$ calculated using cosine similarity measure as defined by the following equation:

$$A_{xy} = \frac{\sum_{k=1}^{K} w_{kx} \cdot w_{ky}}{\sqrt{\sum_{k=1}^{K} w_{kx}^2} \cdot \sqrt{\sum_{k=1}^{K} w_{ky}^2}} \, . \tag{5}$$

Step 3: Apply the complete linkage hierarchical clustering algorithm to the document association matrix $D \times D$ to determine the document clusters. The taxonomy of document clusters resulted from this process is thus a hierarchy of document categories representing the contextual content relevant to the domain of the training parallel corpus. This taxonomy, providing a hierarchical schema, will thus act as the structural backbone of a multilingual document directory to facilitate categorization of other domain-relevant multilingual documents either collected from the Web or residing in some multilingual document repository.

5 Development of a Multilingual Text Categorization Neural Network

To develop a concept-based multilingual text classifier, a 3-layer fully connected feed-forward neural network, also known as multi-layer perception, as depicted in Figure 2, is employed. It is composed of an input layer, a hidden layer and an output layer. All neurons in the neural network are non-linear units with the sigmoid function as the activation function. In the input layer, the number of input units (***K***) is equal to the dimensionality of the concept space. In the output layer, the number of output units (***J***) is equal to the number of all existing document categories as identified during the generation of the multilingual document hierarchy. The number of hidden units in the neural network which depends on the size of the training set and the complexity of the text categorization task the neural network is trained for, is determined empirically using a forward selection criteria [14] based on evaluation of the network's categorization performance. It starts from selecting a small number of hidden units, train the network and record its performance. Then we repeat by slightly increasing the number of hidden units until the error is acceptably small, or no significant improvement is noted, whichever comes first.

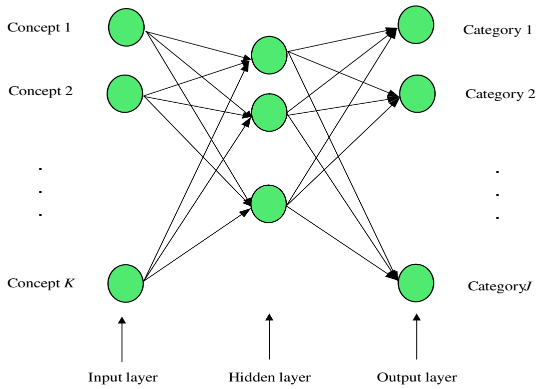


Fig. 2. Architecture of the multilingual text categorization neural network

Training of the multilingual text categorization neural network is achieved with the Backpropagation learning rule [15] based on supervised learning. It is trained using parallel documents which had been categorized during the hierarchical document clustering process as inputs. Formally speaking, all training documents, together with a specification of their pre-defined categories these documents belong to, are presented as a set of training examples, *T*, in the form of input-output pairs as:

$$T = \{(d_i, r_i)\}. \tag{6}$$

where d_i is the concept-based document vector of the i^{th} training document, and r_i is the desired document categories corresponding to d_i. During training, the connection weights of the neural network are initiated to some small random values. The training examples in the training set T are then presented to the neural network in random order, and the connection weights are adjusted according to the Backpropagation learning rule. This process is repeated until the learning errors falls below a predefined tolerance threshold.

For categorization of unseen multilingual documents, concept-based vectors representing these documents are fed to the input layer of the multilingual text categorization neural network as input signals. These input signals are then propagated forward through the neural network so that the output of the neural network is computed in the output layer. As the sigmoid function is used as the activation function in the output units, the output of the neural network is a real-valued categorization vector with element values ranging between *0* and *1*. The categorization vector thus represents a graded categorization decision in which the j^{th} vector element indicates the relevance of the input document to the j^{th} document category. If binary categorization decision is preferred, a threshold can be set such that a document is considered relevant to the j^{th} document category only if the j^{th} element of the categorization vector is greater than the threshold.

6 Conclusions

In this paper, a neural network model for concept-based hierarchical multilingual text categorization is proposed. The key to its effectiveness is the discovery of the multilingual semantic knowledge with the formation of a universal concept space that overcomes the feature incompatibility problem by facilitating representation of documents in all languages in a common semantic framework. By enabling navigation via a hierarchy of conceptually related multilingual documents, global knowledge discovery is facilitated. This form of concept-based multilingual document browsing is particularly important to users who need to stay competent by keeping track of the global knowledge development of a certain subject domain regardless of language.

References

1. Anderberg, M.R.: Cluster Analysis for Applications. Academic Press, Inc., New York (1973)
2. Carbonell, J.G., Yang, Y., Frederking, R.E., Brown, R.D., Geng, Y., Lee, D.:Translingual Information Retrieval: a Comparative Evaluation. In Pollack, M. E. (ed.) IJCAI-97 Proceedings of the 15th International Joint Conference on Artificial Intelligence (1997) 708-714
3. Chen, H., Schuffels, C., Orwig, R.: Internet Categorization and Search: A Self-Organizing Approach. Visual Communications for Image Representation, **7** (1996) 88-102

4. Crouch, C.: An Approach to the Automatic Construction of Global Thesaurus. Information Processing and Management, **26** (1990) 629-640
5. Davis, M.: New Experiments in Cross-Language Text Retrieval at Nmsu'S Computing Research Lab. In Proceedings of the Fifth Retrieval Conference (TREC-5) Gaithersburg, MD: National Institute of Standards and Technology (1996)
6. Haykins, S.: Neural Networks: A Comprehensive Foundation. (2^{nd} edition) Upper Saddle Rive, New Jersey, Prentice Hall (1999)
7. Honkela, t., Kaski, S., Lagus, K. , Kohonen, T.: Newsgroup Exploration with WEBSOM Method and Browsing Interface. Technical Report A32. Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland (1996)
8. Kaski, S., Honkela, T., Lagus, K., Kohonen, T.: WEBSOM — Self-Organizing Maps of document collections. Neurocomputing, **21** (1998) 101-117
9. Jing, Y , Croft, W. B.: An Association Thesaurus for Information Retrieval. Technical Report 94-17, Department of Computer Science, University of Massachusetts, Amherst (1994)
10. Kohonen, T.: Self-Organizing Maps. Springer-Verlag, Berlin (1995)
11. Landauer, T. K. , Littman, M. L.: Fully Automatic Cross-Language Document Retrieval. In Proceedings of the Sixth Conference on Electronic Text Research, (1990) 31-38
12. Lin, X., Soergel, D., Marchionini, G.: A Self-Organizing Semantic Map for Information Retrieval. In Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval, Chicago (1991) 262-269
13. Qiu, Y.: Automatic Query Expansion Based on a Similarity Thesaurus .PhD thesis, Swiss Federal Institute of Technology (1995)
14. Ripley, B. D.: Pattern Recognition and Neural Networks. Cambridge University Press (1996)
15. Rumelhart, D.E., Hinton, G.E. , Williams, R.J.: Learning Internal Representations by Error Propagation. In Parallel distributed processing : explorations in the microstructure of cognition, chapter 8. MIT Press, Cambridge, MA (1986)
16. Salton, G. Automatic: Text Processing: The Transformation, analysis, and Retrieval of Information by Computer. Addison-Wesley, Reading. MA (1989)

Chinese Syntactic Category Disambiguation Using Support Vector Machines

Lishuang Li[1], Lihua Li[2], Degen Huang[1], and Heping Song[2]

[1] Department of Computer Science and Engineering, Dalian University of Technology
Dalian, Liaoning 116023, China
[2] Department of Mathematics-Physics, Hebei Normal University of Science & Technology
Qinhuangdao, Hebei 066004,China
nlp@cjmt.com

Abstract. This paper presents a method of processing Chinese syntactic category ambiguity with support vector machines (SVMs): extracting the word itself, candidate part-of-speech (POS) tags, the pair of candidate POS tags and their probability and context information as the features of the word vector. A training set is established. The machine learning models of disambiguation based on support vector machines are obtained using polynomial kernel functions. The testing results show that this method is efficient. The paper also gives the results obtained with neural networks for comparison.

1 Introduction

Part-of-speech (POS) tagging is fundamental in nature language processing, its potential applications exist in many areas including speech recognition, speech synthesis, machine translation and information retrieval. Syntactic category disambiguation is the key to part-of-speech tagging. Words are often ambiguous in their part-of-speech. The Chinese word "*Lingdao*" for example can be either a noun or a verb. In an utterance, this ambiguity is normally resolved by the context of a word: e.g. in the sentence "*Shengzhengfu Lingdao Quan Sheng Renmin*", "*Ling dao* "can only be a verb. The actual category of a word depends on the syntactic (and sometimes semantic) context. So tagging must perform syntactical disambiguation.

In the last several years, many tagging systems for different languages [1],]2],[3] have been developed based on various techniques. For Chinese POS tagging, the used methods include probabilistic models [4], neural networks [5] and hybrid systems [3].

In this paper, we use a method based on support vector machines to disambiguate the ambiguity of part-of-speech categories in Chinese text corpora. It needs less training samples, and has better generalization performance. We also carry out experiments with neural networks, and the precision produced by the support vector machine method is higher.

2 Support Vector Machines (SVMs)

Support vector machines first introduced by Vapnik [6] are a supervised machine learning algorithm for binary classification. SVMs have advantages over conventional

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 246–250, 2005.

statistical learning algorithm: SVMs have high generalization performance independent of dimension of feature vectors; SVMs can carry out their learning with all combinations of given features without increasing computational complexity by introducing the kernel function. SVMs have been applied to many pattern recognition problems. In the field of natural language processing, they are applied to text categorization [7], Chinese P-Name identification [8], etc.

2.1 Optimal Hyperplane

Given training examples

$$S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_\ell, y_\ell)\},\ \boldsymbol{x}_i \in R^n,\ y_i \in \{-1, +1\} \tag{1}$$

$\boldsymbol{x}_i$ is a feature vector (n dimension) of the i-th sample. y_i is the class (positive(+1) or negative(-1) class) label of the i-th sample. ℓ is the number of the given training samples. SVMs find an "optimal" hyperplane: $(\boldsymbol{w} \cdot \boldsymbol{x} + b) = 0$ to separate the training data into two classes. The optimal hyperplane can be found by solving the following quadratic programming problem:

$$\begin{array}{ll} \text{max} & \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i y_i \alpha_j y_j K(\boldsymbol{x}_i \cdot \boldsymbol{x}_j) \\ \text{subject to} & \sum_{i=1}^{\ell} y_i \alpha_i = 0, \quad 0 \le \alpha_i \le c, i = 1,2,\ldots,\ell \end{array} . \tag{2}$$

The function $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i) \cdot \phi(\boldsymbol{x}_j)$ is called kernel function. Given a test example, its label y is decided by the following function:

$$f(\boldsymbol{x}) = \text{sgn}[\sum_{\boldsymbol{x}_i \in sv} \alpha_i y_i K(\boldsymbol{x}_i \cdot \boldsymbol{x}) + b] \cdot \tag{3}$$

2.2 Multi-class Classifiers

Basically, SVMs are binary classifiers, thus we must extend SVMs to multi-class classifiers in order to solve multi-class discrimination problems. There are two popular methods to extend a binary classification task to that of K classes: *one class vs. all others* and *pairwise*. Here, we employ the simple *pairwise* method. This idea is to build K×(K-1)/2 classifiers considering all pairs of classes, and final decision is given by their weighted voting.

3 Disambiguate the Ambiguity of POS Categories in Chinese Text Corpora Using SVMs

Unlike English, written Chinese does not delimit words by spaces and there is no clue to tell where the word boundaries are. So it is necessary to segment Chinese texts before processing. In this case, a sentence is expressed as follows:

$S = (w^1, w^2, \ldots, w^n)$, where w^i is the *i*-th word in the sentence. POS tagging is the application of a POS tag to each word, the result of POS tagging is expressed as follows: $T = (t^1, t^2, \ldots, t^n)$ where t^i is the tag for the POS of word w^i. Our goal is to assign the correct POS tag for each ambiguous word. Therefore, this problem can thus be regarded as classification problems and can be handled by SVMs.

Firstly, we segment words to each sentence of the text corpora using a Chinese automatic segmentation system, and assign each word its candidate POSs using a word dictionary, then extract features corresponding to every word. Finally, a machine learning model SVMs-based to disambiguate the ambiguity of part-of-speech categories is set up by choosing proper kernel functions.

3.1 POS Categories

We define 26 kinds of POS categories in Chinese texts in advance including noun (/n), verb (/v), adjective (a/), adverb (/d), etc.

3.2 Feature Extraction (Information Used in Classification)

As mentioned above, when the result of word segmentation of a sentence in Chinese texts is input, we output the POS for each ambiguous word. Therefore, the features are extracted from the input Chinese sentences. Here, we define the following items as features.

Table 1 summarizes the types of features and their values.

Table 1. Summary of features and their values

Type of feature	Value
The candidate POS tags	n, v n
The pair of candidate POS tags and their occurrence probability	n, 0.71,v, 0.29 n,1
Word	surface form of the word itself

The candidate POS tags are determined in advance for each word by using a word dictionary, they can be ambiguous or unique.

The pair of candidate POS tags and their occurrence probability are the important features. The probability that the POS tag might be assigned to word w^i is estimated from the training data as

$$P(t^p | w^i) = \frac{|t^p, w^i|}{|w^i|} \tag{4}$$

where $|t^p, w^i|$ is the number of times that t^p (a kind of POS) might be assigned to w^i, and $|w^i|$ is the total number of times that might appear in the training set.

The "word" is surface form of the word.

The POS of a word depends on the context of the word. Therefore, we use contextual information of two previous and two subsequent words of the current word as features: the candidate POS tags, the pair of candidate POS tags and their occurrence probability, and the word itself, the same as the current word.

3.3 Choosing Kernel Functions

Here, we choose polynomial kernel functions: $K(\boldsymbol{x}, \boldsymbol{x}_i) = [(\boldsymbol{x} \cdot \boldsymbol{x}_i) + 1]^d$ to build an optimal separating hyperplane by testing d=1, 2, 3 respectively.

4 Experiment Results and Comparison with Neural Networks

Table 2. Precision for disambigution with SVM

polynomial order	Precision
d=1	89.10
d=2	92.11
d=3	91.93

Table 3. Comparison of SVM and NN

Method	Precision
SVM (ignoring word information)	89.81 (*d*=2)
NN	88.74

We use less than one-month news of year 1998 from the People's Daily as the training and testing corpus (divided into 2 parts randomly with a size ratio for training/testing of 3/1) to conduct an open test experiment. It contains about 200000 words, and about 75000 ambiguous words. The results of our experiments are given in Table 2.

Experiments are also performed with neural network: we use three-layer BP network. This method does not use word information directly: only use the probability of the occurrence of a POS in each word. We eliminate the feature of word information in the support vector machine method, the comparison of precision with these two methods is shown in Table 3: the precision of the SVM method is higher.

5 Conclusions and Future Work

We disambiguate the ambiguity of part-of-speech categories in Chinese texts using support vector machine. The results show that SVMs can avoid overfitting in the high dimension feature space and can obtain higher precision with smaller training sets. For the future, we would develop a general Chinese POS tagger.

To increase the precision, we can add more features related to the current word, such as the semantic information, the position of each word in the sentence, etc. We also can experiment with different context window size, different multi-classification method, and combine with rule-based model.

In our experiment, the support vector machine method produces better precision than the neural network. For the future work, we should conduct experiments by using word information in NN method and expect to have better performance.

References

1. Murata, M., Ma, Q., Isahara, H.: Part of Speech Tagging in Thai Language Using Support Vector Machine. In: Isahara, H., Ma, Q. (eds.): Proceedings of the Second Workshop on Natural Language Processing and Neural Networks, Japan NLPRS2001, Tokyo (2001) 24-30
2. Weischedel, R., Meteer, M., Schwartz, R., Ramshaw, L., Palmucci, J.: Coping with Ambiguity and Unknown Words through Probabilistic Models. Computational Linguistics, **19** (1993) 359–382
3. Huang, D.G., Zhang, L.J., Zhang, Y.L., Yang, Y.S.: Disambiguation Mechanism Using Rule Techniques and Statistics Techniques. Mini-Micro Systems. **24** (2003) 1252–1255
4. Wei, O., Wu, J., Sun, Y.: Analysis and Improvement of Statistic-Based-Chinese Part-of-Speech Tagging. Journal of Software. **4** (2000) 473–480
5. Yu, X., Zhu, F.S.: Chinese Syntactic Category Disambiguation with the Neural Networks. Computer Research & Development. **4** (1998) 367–369
6. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer-Verlag, Berlin (1995)
7. Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In: Nedellec, C., Rouveirol, C. (eds.): Machine Learning. Lecture Notes in Computer Science, **1398** (1998) 137–142
8. Li, L.S., Chen, C.R., Huang, D.G., Yang, Y.S.: Identifying Pronunciation-Translated Names from Chinese Texts Based on Support Vector Machines. In: Yin, F.L., Wang, J., Guo, C.A. (eds.): Advances in Neural Networks-ISNN 2004. Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg **3173** (2004) 983-988

A Clustering Algorithm for Chinese Text Based on SOM Neural Network and Density

Zhiqing Meng[1], Hongcan Zhu[2], Yihua Zhu[1], and Gengui Zhou[1]

[1] College of Business and Administration
Zhejiang University of Technology, Hangzhou, Zhejiang 310032, China
[2] College of Information Engineering
Xiangtan University, Xiangtan, Hunan 411105, China

Abstract. This paper introduces a clustering algorithm for Chinese text based on both SOM (Self-Organizing Map) neural network and density. The algorithm contains two stages. During the first stage, Chinese text are transformed into text vectors, which are used as training data of SOM and mapped by training SOM so that an initial clustering result for text data, i.e., a virtual coordinates set, is obtained. Then, during the second stage, the virtual coordinates set is further clustered according to density. It should be pointed out that the proposed algorithm in the first stage is different from the existing ones. Moreover, in the second stage, it outperforms other algorithms in computing time due to decreasing dimension. Numerical experiment shows that the algorithm is efficient for clustering text data and high multi-dimensional data.

1 Introduction

In the last years, study of cluster has become increasingly popular and important in data mining. Researchers have presented variety of clustering algorithms based on density for large-scale databases[1-4]. These algorithms can accelerate clustering velocity and improve efficiency by means of decreasing the number of core points[1], using reference points[2], or plotting out grid of space data[3]. But, there exist some difficulties when they are used to cluster high dimensional data. To solve these problems, Juha Vesanto presents a two-stages clustering method[4], where a SOM neural network is used during the first stage and a hierarchical clustering algorithm or k-means algorithm is used during the second stage. The two-stages clustering method is very efficient for high dimensional data, since the SOM method is good at disposing large-scale data.

In recent years, some algorithms based on SOM, suitable for English and Chinese textual documents, were studied[5-9]. Dittenbach et al constructed several independent SOMs, whose sizes and hiberarchy increase gradually[5]; Hung et al considered dynamic units in SOM[6]; Elias Pampalk proposed a modification version of the SOM method by introducing tension and mapping ratio extension[7]; Kohonen et al gave a dynamic structure of SOM, some factors of which can be adjusted by users[8]; and besides, Xu et al studied clustering Chinese text by static SOM[9].

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 251–256, 2005.

In this study, we proposed a clustering algorithm based on both SOM neural network and density. The process of our clustering algorithm can be partitioned into two stages. The objective of the first stage is to map text data into SOM's virtual coordinates, which constitutes a virtual coordinates space, i.e., an initial clustering result. Moreover, the second stage aims to work out final clustering results by virtue of a density algorithm. Our algorithm features in that, during the first stage training, weights of the SOM are only used to adjust parameters of virtual coordinates and that, during the second stage, only two-dimension coordinates than high-dimension ones are clustered, which decrease computing time dramatically.

2 Algorithm

2.1 Algorithm Description

Now, we introduce our algorithm based on SOM neural network and density. This algorithm consists of two-stages clustering. The first clustering stage focuses on clustering samples by means of training SOM neural network, while the second one aims to cluster virtual coordinates by a density algorithm. By training SOM, n given sample data produced from text data or high multi-dimensional databases are transformed into two-dimensional virtual coordinates (sequence data) represented with rectangles.

It is not easy to cluster factual data that exhibit various shapes for high dimensional data. Fortunately, clustering algorithm based on density during the second clustering stage, where the input data have been transformed into two-dimensional space by the SOM, can overcome the above shortcomings. In order to improve clustering efficiency, during the training process of SOM, we use virtual coordinates to adjust its value in neighborhood of each winner so that other units are close to the winner.

2.2 Training Algorithm of SOM

The SOM consists of disciplinarian neuron of two dimensional matrix $m \times m$, which a weight vector from a input vector X_i to each neuron or connected unit j in output layer is defined as a vector w_{ij}, where the dimension value of X_i is equal to that of w_{ij}. The number of neuron from ten to thousand decides efficiency and validity of the SOM. The given initial coordinates (ax_i, ay_i) at unit i is location of grid of SOM. The training algorithm of SOM is as follows:

(a) In each step of the iterative training, choose a textual vector X_i as input vector and compute distances from vector X_i to all units of SOM so that the winner unit, denoted by w_{ib}, is obtained. Here, w_{ib} is the closest vector to X_i among all vectors w_{ij}, i.e.,

$$\|X_i - w_i\| = \min_j \|X_i - w_{ij}\| \tag{1}$$

(b) Select weigh value w_{ij} of point in the neighborhood of the winner b to close input vector by training SOM. The iteration equality is satisfied in grid as follows.

$$w_{ij}(t+1) = w_{ij}(t) + \partial(t)[X_i - w_{ij}(t)] \tag{2}$$

where t is the number of iteration, $\partial(t)$ is learning rate in $(0,1)$. Additionally, w_{ij} approaches to X_i , when t increases.

(c) Train coordinates value (ax_i, ay_i) of other unit i close to the winner unit b according to equation (3) and (4).

$$ax_i(t+1) = ax_i(t) + \partial(t)exp(-|\triangle B_i(t)|)(ax_b(t) - ax_i(t)) \tag{3}$$

$$ay_i(t+1) = ay_i(t) + \partial(t)exp(-|\triangle B_i(t)|)(ay_b(t) - ay_i(t)) \tag{4}$$

where $\triangle B_i(t)$ is a distance error satisfying

$$\triangle B_i(t) = \|Xi - w_{ij}(t)\| - \|X_i - w_{ib}\| \tag{5}$$

Similarly, coordinates value (ax_i, ay_i) of unit i in the neighborhood of the winner approaches coordinates value (ax_a, ay_b) of the winner b, when t increases.

(d) Obtain new coordinates (ax_i, ay_i) in the neighborhood of the winner as a mapping structure of input vector X_i , denoted by $X_i(ax_b, ay_b)$, for SOM.

(e) Then, we repeat to do (a) to (d) till all textual vectors is trained.

(f) Get all mapping structure set $X_i(ax_b, ay_b)$ of all input vector X_i corresponding virtual coordinates set (ax_b, ay_b).

Finally, we obtain a initial results of cluster according to the mapping structure set $X_i(ax_b, ay_b)$ with the above algorithm of SOM. It should be stressed that the virtual coordinates set (ax_b, ay_b) will be further clustered by density algorithm in the second stage.

2.3 A Density Algorithm

We give some definitions before we move on to the density algorithm.

Definition 1 (Neighborhood) Neighborhood of point p in space is defined as all the points within a circle with center p and radius r. Besides, it is denoted by $NEps(p,r) = \{q \in D | dist(p,q) \leq r\}$, where D is a database and r is a real number.

Definition 2 (Density) Density of point p (with respect to r) in space is defined as the number of all the points within $NEps(p,r)$ and is denoted by $D(p,r)$.

Definition 3 (CorePoints) Point p is called core point if $D(p,r) > MinPts$, a given value.

Definition 4 (Directly Density-Reachable) A point p to a point q is called directly density-reachable if $p \in NEps(q)$ and q is a core point.

If a point p to a point q is directly density-reachable, then it is denoted by $a(p,q) = 1$; otherwise, $a(p,q) = 0$.

Definition 5 (Density-Reachable) A point p to a point q is called density-reachable if p_i to p_{i+1} is directly density-reachable for $i = 1, 2, \cdots, n$,where $p_1 = p, p_n = q$.

Definition 6 (Density-Connectible) A point p to a point q is called density-connectible, if there exists another point s such that both s to p and s to q are density-reachable.

Definition 7 (Cluster) A nonempty set C is called a cluster in database D if and only if C satisfies following two conditions:

(1) for $p \in C$, that p to q is density-reachable implies $q \in C$;

(2) p to q is density-connectible for any $p, q \in C$.

Definition 8 (Noise) All points in database D are noise if they do not belong to any cluster.

We give a clustering algorithm based on density algorithm.

```
Step 1. Let $C = \{p_i, i = 1, 2, \cdots, m\}$ and initialize $r, MinPts$.
Step 2. For the given $r$ and $MinPts$. Compute all core points set $CP$ and
        their directly density-reachable matrix $DDRM$ in the output grid
        space.
Step 3. Based on density-reachable and density-connectible, find out
        cluster results.
Step 4. Obtain the cluster result $C = \{C_1, \cdots, C_i, \cdots, C_k\}$, and find out
        all noise points, then stop.
```

2.4 Main Algorithm

Initially, we obtain data samples by the vector-space model[5](VSM) for the Chinese text. A text unit is mapped to a text vector $V(d)$ which contains many words and weight by VSM. Here,

$$V(d) = (t_1, w_1(d); \cdots; t_i, w_i(d); \cdots; t_n, w_n(d)),$$

where t_i is a word, $w_i(d)$ is weight of t_i in a text-data set d. $w_i(d)$ is defined as a frequency function for t_i in d. A text set is denoted by matrix $T(n, w)$, where n is the number of all texts, w is the number of all words.

Now, we elaborate on the algorithm based SOM and density in details. Let r be radius of a neighborhood of point, and $MinPts$ be a given number.

```
Input:   $T(n, w)$, $n, w, r, MinPts$.
Output: A cluster $C = \{C_1, \cdots, C_i, \cdots, C_k\}$, where $C_i$ is sub-cluster in $T$.
Step 1.  Train data set $T$ by using the training algorithm of SOM and get
         a output grid space with virtual coordinates $\{(ax_i, ay_i)\}$.
Step 2.  Compute all core points set $CP$ and their directly density-
         reachable matrix $DDRM$ in the output grid space for $r$ and
         $MinPts$.
Step 3.  Find out cluster results based on density-reachable and density-
         connectible in the output grid space.
Step 4.  Find out all noise points, then stop.
```

Let the number of data be n, the number of grid of SOM be $m(m << n)$, and the number of cluster be C. It is easily known that the time complexity in the first stage and in the second stage are respectively $O(nm)$ and $O(cm^2)$. Therefore, the total time complexity of the algorithm is $O(nm + cm^2)(nm + cm^2 << n^2)$.

3 Experiment Results

In order to investigate the efficiency of algorithm we do experiments to compare our algorithm with hierarchical clustering algorithm based on the SOM and k-means algorithm. We choose Chinese People Daily as textual data source for clustering experiment, where dimension of textual matrix is 500. The efficiency of cluster is measured by rate of integrity Re and rate of precision Pr. Re indicates integrity of cluster and Re exactness of cluster.

When training SOM, we choose learning rate $\partial(t) = 1 - t/R$, where R is a constant larger than the largest number of training and t is the number of training. Let grid of SOM be 10×10 matrix, $t = 500, r = 0.7$, and $MinPts = 3$. Running our programs made by Microsoft Visual C++ over Windows 2000 leads to Table 1 and Table 2. Table 1 is the results of our algorithm and hierarchical clustering algorithm based on the SOM and k-means $(k = 4)$ algorithm.

Table 1. Results of integrity Pr and exactness Re

4 subcluster (310 texts)	k-means $(k = 4)$ algorithm		Our algorithm		Hierarchical clustering algorithm	
	Pr	Re	Pr	Re	Pr	Re
Gymnasium	67	70	69	76	67	67
Medicine	70	74	80	72	66	60
Politics	74	56	68	65	68	59
Economics	64	55	65	66	68	64
Average	69	64	71	70	68	63

It can be clearly seen from Table 1 that our algorithm is also efficient for high dimensional data. Furthermore, our algorithm has higher Re and Pr in comparison with hierarchical clustering algorithm based on the SOM and k-means $(k = 4)$ algorithm. In addition, the isolated point could be easily found out in our algorithm.

Computation time of our algorithm and that of hierarchical clustering algorithm are included in Table 2, and besides, the tendency of computation time demonstrates that computation time of our algorithm is less than that of hierarchical clustering algorithm, when the number of texts increases.

4 Conclusions

The proposed algorithm, based on SOM and density, is efficient for such high dimension data as Chinese test. It significantly decreases computation time and

Table 2. Results of computation time

Number of Texts	Our algorithm	Hierarchical clustering algorithm
50	5s	5s
100	11s	9s
200	15s	25s
300	23s	36s
400	30s	47s

improves efficiency in clustering process and outperforms existing two-stage clustering algorithm. Unlike other SOM-based algorithm that some clustering tasks have to be done manually, this algorithm is able to extract clustering results automatically.

Our further study is likely to cover how to improve integrity and precision by changing the way of choosing virtual coordinates and how to enhance efficiency of clustering text data by modifying the clustering algorithm used in the second stage.

References

1. Zhou, S.G., Zhou, A.Y.: FDBSCAN:A Fast DBSCAN Algorithm. Journal of Software, **11** (2000) 735-744
2. Ma, S., Wang, T.J., Tang, S.W., Yang, D.Q., Gao, J.: A Fast Clustering Algorithm Based on Reference and Density. Journal of Software(in Chinese), **14** (2003) 1089-1095
3. Chen, N., Chen, A.: An Incremental Density-Based Clustering Algorithm. Journal of Software, **11** (2002) 1-7
4. Juha, V.: Clustering of the self-organizing Map. IEEE Transaction on Neural Networks, **11** (2000) 586-600
5. Rauber, A., Merkl, D.: The Growing Hierarchical Self-Organizing Maps Exploratory Analysis of High-Dimensional Data. IEEE Transactions on Neural Networks, **13** (2002) 1331-1341
6. Hung, C., Wermter, S.: A Self-Organising Hybrid Model for Dynamic for Text Clustering. http://citeseer.ist.psu.edu/646370.html. (2003)
7. Elias, P.: A New Approach to Hierarchical Clustering and Structuring of Data with Self-Organizing Maps. Intelligent Data Analysis Journal, Extended version, (2003) 1-23
8. Kohonen, T., Kaski, S., Lagus, K., Saloj?rvi, J., Honkela, J., Paatero, V., Saarela, A.: Self Organization of a Massive Document Collection. IEEE Transactions on Neural Networks, **11** (2000) 574-585
9. Xu, J.S., Wang, Z.O., Wang L.: A Novel Approach of Chinese Text Clustering Based on Self-Organizing Neural Network, Journal of Information(in Chinese), **22** (2003) 676-680

Automatic Caption Detection in Video Frames Based on Support Vector Machine

Jianfeng Xu and Shaofa Li

School of Computer Science, South China University of Technology
Guangdong, Guangzhou 510640, China
xu_jian_feng@tom.com, csshfli@scut.edu.cn

Abstract. Video captions can be used to index large video archives in digital libraries. In this paper, an algorithm for detecting captions in video frames using support vector machine (SVM) is proposed. First, the input video frame is divided into square sub-blocks and a trained SVM is used to identify whether each sub-block is a caption block or not. Second, horizontal projection and vertical projection are performed to locate the candidate caption regions. Finally, false alarms are reduced by caption region verification. Experimental results show that the algorithm has a low missed rate and false alarm rate.

1 Introduction

With the rapid development of multimedia technology, more and more video clips are added to the digital libraries. They need an automatic method to be indexed and retrieved efficiently.

Caption in video frames is an important source of high-level semantics and may do great help to indexing and retrieving.

Jain and Yu propose a text-localization algorithm based on connected component analysis [1]. But connected-component-based approaches [1][2] require text or its background to be monochrome and work not well with small text font where no dominant color prototype can be found.

A caption region usually has dense transitions of hue or brightness against its background. So the edge information can be exploited to find captions. Smith et al. introduce an edge-based method to detect text in video frames [3].

Wu et al. introduce a text-extraction method that treats text as a special texture [4]. First, candidate text regions are located by using second-order derivations of Gaussians. Then vertical strokes are extracted and grouped into tight bounding boxes. Because they propose no method to determine text color, two binary images are used for the text recognition of each text box.

SVM is a new popular tool for data classification [5]. It has also been successfully applied for texture classification [6].

We proposed a new caption-detection method using support vector machine. The rest of this paper is organized as follows. Section 2 describes the caption sub-block

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 257–261, 2005.

detection. Section 3 elaborates on the text region extraction. In Section 3, we present the experiment results. The conclusion is made in section 4.

2 Caption Sub-block Detection

2.1 SVM-Based Texture Classifier

A classification task involves with training and testing data. Each instance in the training set consists of a class label and several features. The goal of SVM is to create a model that determines the class label in a testing instance that is given only the features.

Given a training set of instance-class pairs $(x_i, y_i) \in R^N \times \{\pm 1\}$, I=1,…, l, an SVM construct a classifier in a higher dimensional feature space F. It finds a separating hyper-plane with the maximal margin between the nearest examples call support vectors (SVs) in space F

$$f(z) = \sum_{i=1}^{m} \lambda_i y_i \Phi(x_i)\Phi(z) + b. \tag{1}$$

where $K(x_i, z) = \Phi(x_i)\Phi(z)$ is called the kernel function. We select the Radial basis function (RBF) as the kernel defined by

$$K(z, x_i) = e^{-\gamma \|z - x_i\|^2}. \tag{2}$$

where the parameter γ is determined by cross-validation and the classifier is trained using standard quadratic programming technique.

The gray levels of an enough amount of $N \times N$ (8×8 in the experiment) sub-blocks and their real classes (caption or non-caption) are used as the training set.

After training, the SVM classifier is used to determine the texture classes of $N \times N$ sub-blocks of the input image. If a sub-block is classified as caption block, all the pixels in it are set as white pixels, otherwise black.

2.2 Candidate Text Region Generation and Verification

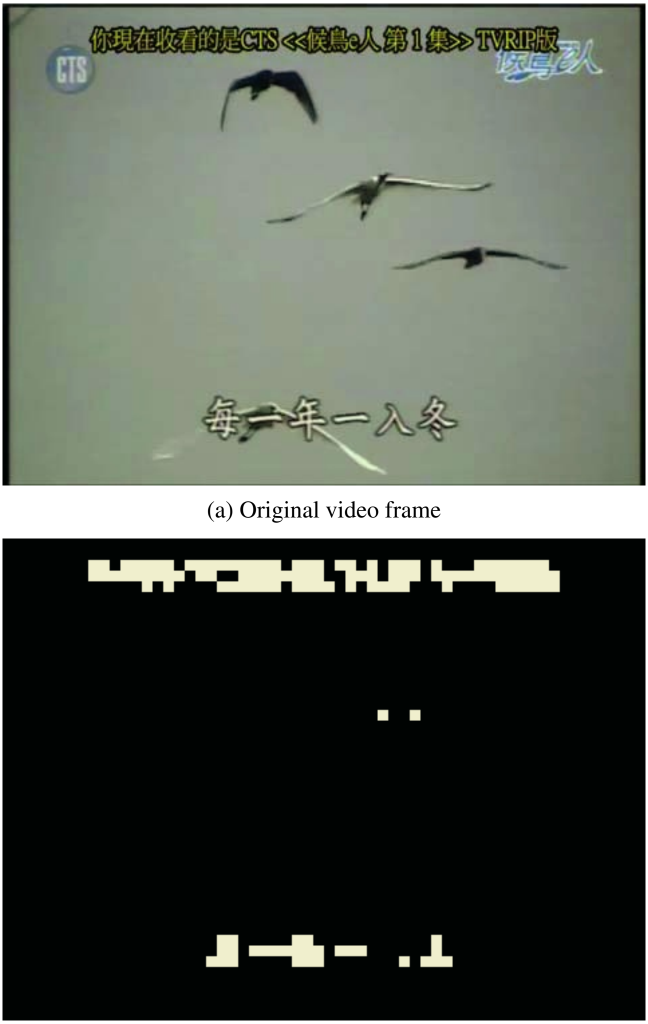
From the texture-classified image, we use horizontal projection and vertical projection to find candidate text areas.

(1) Horizontal projection and vertical merging

We consider only those text occurrences that consist of at least 2 letters and/or digits. If a horizontal scan line contain less than 17 white pixels, we remove the white pixels in it. Those adjacent scan lines each of which contains more than 17 white pixels are merged to form horizontal bands. The generated horizontal bands are than subject to high verification. If the high of a horizontal band is less than 9 pixels, it is discarded.

(2) Vertical projection and horizontal expand-and-merge process

The vertical projection is applied to each horizontal band and vertical lines that contains 8 or more white pixels are detected. We call these detected vertical lines as



(a) Original video frame

(b) After texture classification

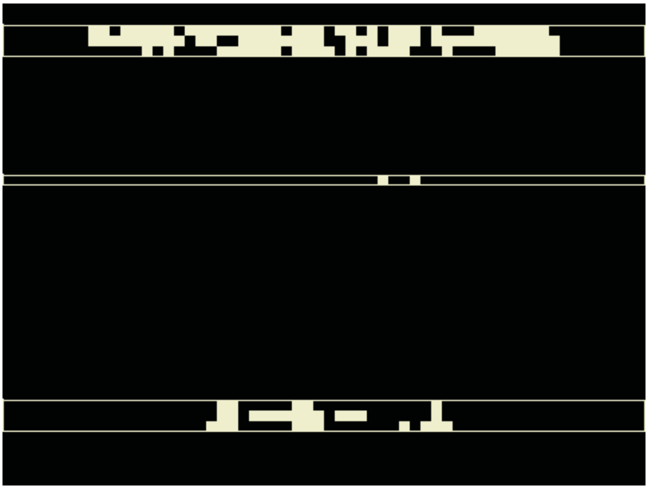
Fig. 1. Intermediate and final results of segmentation

potential text area segments. Potential text area segments are then expanded and merged to form candidate text boxes. Supposed the high of the horizontal band is H, When 2 potential text area segments are within a distance less than 2×H to each other, they are merged into one candidate text area.

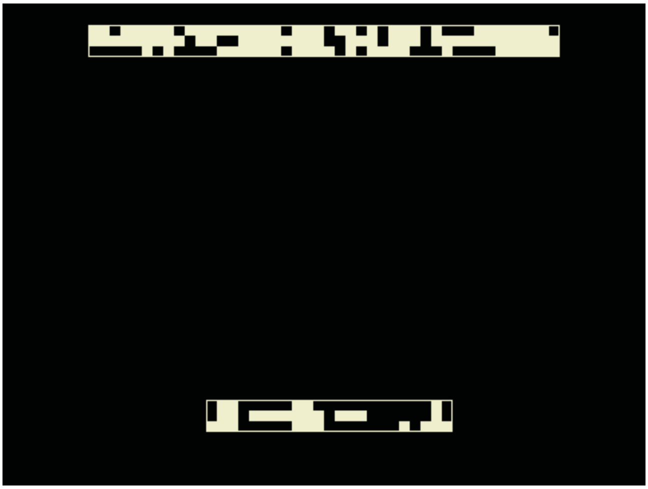
(3) Step (1) and step (2) are repeated on each candidate text areas until no more candidate text areas are generated.

(4) The detected candidate text areas are subject to a shape verification process based on their high and high-to width ratio. If the computed values fall outside the pre-specified ranges, the candidate text area is removed.

Fig. 1 shows the intermediate and final results of detection.



(c) After horizontal projection and vertical merging



(d) After vertical projection, horizontal expand-and-merge process, and shape verification

Fig. 1. (Continued)

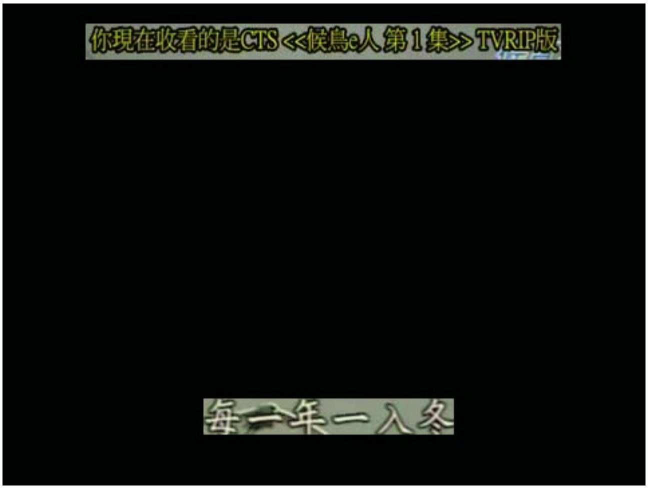
3 Experiment Result

The experimental set is composed of 150 color video frames of size 320×240 and 200 color image of different size, containing 2011 characters.

As shown in Table 1, the proposed algorithm achieved a missed rate of 4.8%.

4 Conclusions

In this paper, we have proposed a method to detect text in images and video frames using SVM-based texture classifier. It works well for the tested dataset with low missed rate.



(g) Location result

Fig. 1. (Continued)

Table 1. Result of text detection

Number of characters	Correctly detected	Missed	False alarm regions	Missed rate
2011	1914	97	191	4.8%

References

1. Jain, A. K., Yu, B.: Automatic Text Location in Images and Video Frames. Pattern Recognition, **31** (1998) 2055-2076
2. Lienhart, R.: Automatic Text Recognition for Video Indexing. Proc. ACM Multimedia 96, Boston MA (1996) 11-20
3. Smith, M. A., Kanade, T.: Video Skimming and Characterization through the Combination of Image and Language Understanding. IEEE International Workshop on Content-Based Access of Image and Video Database (1998) 61-70
4. Wu, V., Manmatha, R., Riseman, E. M.: Finding Text in Images. Proc. 2nd ACM International Conference on Digital Libraries, Philadelphia (1997) 23-26
5. Vapnik, V.: The Nature of Statistical Learning Theory. Springer-Verlag, New York (1997)
6. Jeong, K.Y., Jung, K., Kim, E. Y., Kim, H. J.: Neural Network-based Text Location for News Video Indexing. Proc. IEEE International Conference on Image Processing, Japan (1999) 319-323

Selection of ICA Features for Texture Classification

Xiangyan Zeng[1], Yenwei Chen[2], Deborah van Alphen[1], and Zensho Nakao[3]

[1] California State University Northridge, CA 91325, USA
{xiangyan.zeng,dvanalphen}@csun.edu
[2] Ritsumeikan University, Shiga 525-8577, Japan
chen@is.ritsumeikan.ac.jp
[3] University of the Ryukyus, 903-0213 Okinawa, Japan
nakao@augusta.eee.u-ryukyu.ac.jp

Abstract. In the literature of texture analysis, research has been focused on the issue of feature extraction. Much less attention has been given to the important issue of feature selection, however. Most of the methods rank the features by some criteria, for instance, the eigenvalues and the Fish Criterion, and select some percentage of the top features. In this paper, we propose a feature selection scheme for texture classification. We use the filter bank obtained by independent component analysis (ICA) of nature scenes for multichannel feature extraction and the least squares support vector machine (LS-SVM) for classification. The dimension of the ICA features is first reduced using principal component analysis (PCA). Recursive feature elimination (RFE) is then employed to select the relevant features for LS-SVM classification. Our experimental results show that the proposed method achieves better classification accuracy than the simple PCA and the Fisher Criterion methods.

1 Introduction

Texture analysis is an important part of many computer vision systems. A large number of texture features have been proposed in the literature [1], [2], [3]. One important category of feature extraction is the signal processing methods which use filter banks (multichannel filters), such as Gabor filters and wavelet transforms. The basic idea of multichannel filtering methods is to enhance edges and lines of different orientations and scales to obtain different feature components. In recent years, ICA has been applied to feature extraction of nature image data [4], [5]. The obtained ICA filters exhibit Gabor-like structures and provide an orthogonal basis for image coding. The ICA filter banks have been used in image denoising as an adaptive option of wavelet basis [6]. In this paper, we use the ICA filters to extract texture features. For the multichannel filtering methods, large number of filters leads to expensive computation in both feature extraction and classification. The issue of filter selection is crucial for optimizing the filtering operation [7]. A common method in ICA research is to reduce the dimension by principal component analysis (PCA), and then to extract the ICA filters in a lower dimensional space [8]. PCA is effective in reducing a small number of filters which can be considered as noisy basis. It cannot guarantee a good perform-

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 262–267, 2005.

ance of very few filters, however. A feature selection process is needed for choosing an appropriate subset of ICA features.

Feature selection approaches are categorized into two classes based on whether they are dependent on the learning algorithms used to construct classifiers. Filter methods utilize intrinsic properties of data as criteria, such as Fisher Criterion, and are independent of learning algorithms [9]. Wrapper methods evaluate features by the classification performance of specific classifiers [10]. In general, wrapper methods can achieve higher accuracy than filter methods, but at an expense of training much more classifiers. Different from the wrapper methods which use the classification error as a criterion, recursive feature elimination (RFE) uses the change in the objective function when one feature is removed as a ranking criterion [11]. The features which contribute least to the classification are removed iteratively. Having a good performance with very small training sets, RFE was observed to be robust to data overfitting.

Another important component in texture classification is the classifier being used. In recent years, the support vector machine has emerged as a successful method of pattern recognition and regression [12]. An important advantage of the SVM is that it aims at minimizing a bound on the generalization error instead of minimizing the training error as do many other methods. Therefore, it has good generalization performance even for small training sets. The least square version of SVM (LS-SVM) formulates the problem into a linear system and is computationally efficient for training a large number of classifiers [13]. In this paper, we use LS-SVM and RFE for feature selection and classification of textures. The rest of the paper is organized as follows. Section 2 briefly describes the ICA texture features. The LS-SVM and RFE algorithms are introduced in Section 4. The texture classification experiments and concluding remarks are given in Section 4, and Section 5.

2 Texture Representation Using ICA Features

We start with the assumption that an image is reshaped row-by-row into a column $\mathbf{z} = (z_1, z_2, \cdots z_N)^T$, which can be represented in terms of a linear superposition of basis functions $\mathbf{a}_i$:

$$\mathbf{z} = \sum_i s_i \mathbf{a}_i = \mathbf{As} \qquad (1)$$

where each column of **A** is a basis function $\mathbf{a}_i$, and **s** is a coefficient vector. The goal of efficient coding is to find a set of $\mathbf{a}_i$ that result in the coefficient values being statistically as independent as possible over an ensemble of images. Thus, the linear image analysis process is to find a matrix **W**, so that the resulting vector, **x** =**Wz**, recovers the underlying causes **s**, possibly permuted and rescaled. Each row of **W** is reshaped into a two-dimensional filter. Several ICA algorithms have been proposed. We use the FastICA algorithm proposed by A. Hyvarinen et al [8].

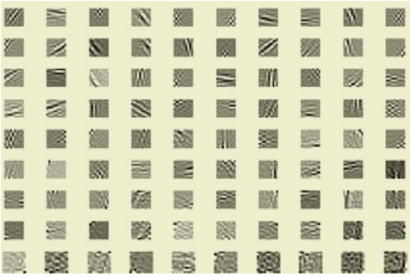


Fig. 1. 90 ICA filters obtained by training an ensemble of 10×10 natural image patches

By applying ICA to image patches of natural scenes instead of specific class images, we obtained the ICA filters that are local and oriented edge filters (Fig.1). Actually, Barlow has proposed earlier that our visual cortical edge detectors might be the end result of a redundancy reduction process, which means the edge detectors can be obtained from the decorrelation process. In the following sections, we use the ICA filters as a Gabor-like filter bank for multichannel texture classification. Denote the *k*-th filtered image as $\mathbf{x}_\mathrm{k}$, the local energy $\sum_{i,j} \mathbf{x}_k (i, j)$ is used as the *k*-th ICA texture feature. If the bank consists of *N* filters, the feature vector has *N* components. PCA is commonly used to reduce the feature dimension in the FastICA algorithm. It is effective in reducing a small portion of filters which can be considered as noisy basis. A very low dimensional space obtained by PCA does not have good classification accuracy as shown in the experiment.

3 Least Squares Support Vector Machines and Recursive Feature Elimination

Given a training set of *N* data points $\{(\mathbf{x}_1, y_1),\cdots,(\mathbf{x}_N, y_N)\}$ where $\mathbf{x}_i \in R^d$ is a feature vector and $y_i \in \{\pm1\}$ is the corresponding target. The data points are mapped into a high dimensional Hilbert space using a nonlinear function $\varphi(\cdot)$. In addition, the dot product in that high dimensional space is equivalent to a kernel function in the input space, i. e. $K(\mathbf{x}_i,\mathbf{x}_j) = \varphi(\mathbf{x}_i)\cdot\varphi(\mathbf{x}_j)$. The LS-SVM classifier is constructed by minimizing

$$\frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{1}{2}C\sum_i {e_i}^2 \tag{2}$$

subject to the equality constraints:

$$y_i - (\mathbf{w}\cdot\varphi(\mathbf{x}_i) - b) = e_i$$

where *C*>0 is a regularization factor, *b* is a bias term, and e_i is the difference between the desired output and the actual output. The Lagrangian for problem (2) is

$$\Re(\mathbf{w}, e_i; \alpha_i) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{1}{2}C\sum_i e_i^2 + \sum_i \alpha_i \left[y_i - \mathbf{w}\cdot\varphi(\mathbf{x}_i) + b - e_i\right] \tag{3}$$

where α_i are Lagrangian multipliers. The Karush-Kuhn-Tucker (KKT) conditions are reduced to a linear system

$$\begin{bmatrix} \mathbf{Q} & 1_n \\ 1_N^T & 0 \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix} \tag{4}$$

where $\mathbf{Q}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) + \sigma_{ij}/C$, and $\sigma_{ij} = 1$ if *i*=*j* and 0 otherwise. Parameters $\boldsymbol{\alpha}$ and b can be efficiently obtained using the conjugate gradient method. LS-SVM avoids solving the quadratic programming problem and greatly simplifies the training of a large number of classifiers in feature selection.

Recursive feature elimination is to iteratively remove the features with least influence on the classification decision. The classification ability of LS-SVM depends on the classifier margin. Therefore, the ranking criterion of feature selection is the margin difference caused by the removal of the feature

$$DW^{-m} = \sum_{i,j} \alpha_i \alpha_j (K(\mathbf{x}_i, \mathbf{x}_j) - K(\mathbf{x}_i^{-m}, \mathbf{x}_j^{-m})) \tag{5}$$

where $\mathbf{x}_i^{-m}, \mathbf{x}_j^{-m}$ are the vectors in which the *m*-th feature has been removed. For the nonlinear LS-SVM which uses Gaussian Kernel,

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{-1}{2\sigma^2}\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2\right) \tag{6}$$

the margin difference can be efficiently computed by

$$DW^{-m} = \sum_{i,j} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \cdot (1 - 1/K(\mathbf{x}_i^m, \mathbf{x}_j^m)) \tag{7}$$

where $\mathbf{x}_i^m, \mathbf{x}_j^m$ are the *m*-th components of $\mathbf{x}_i$ and $\mathbf{x}_j$. For texture classification which involves more than two classes, we adapt the RFE algorithm to the following multi-class algorithm.

```
Repeat until the terminating condition is satisfied
    Train all the binary LS-SVMs and obtain $\boldsymbol{\alpha}^k$ , $k$=1…$C$.
    Remove feature $m^* = \min_m \left(\sum_{k=1}^C \left(DW^{-m}\right)^2\right)$.
End repeat.
```

4 Experimental Results

We have carried out experiments using a dataset of 30 textures selected from the Brodatz Album. Each image is of size 640x640 and is divided into 400 32x32 non-overlapping subimages. 1.25% of the subimages are used in training LS-SVM, 2.5% are used as a validation set as required, and the rest are used for testing. The efficient one-against-all strategy is utilized to combine 30 binary classifiers to implement the 30-class texture classification. In each binary classifier, one texture is assigned as the positive class and the others as the negative class. Nonlinear LS-SVM with Gaussian kernel has the best performance in the experiments. The parameter σ^2 is decided using a validation process.

In the following experiments, we first compare the proposed method with PCA and the Fisher Criterion (FC). Starting from the initial LS-SVM with 90 ICA features, the proposed method iteratively removes one feature at a time and then retrains LS-SVM. The Fish Criterion method selects the top features of large criterion scores [14]. The PCA method simply changes the LastEig parameter in the FastICA algorithm [8]. LS-SVM is used for classification in the PCA and the Fish Criterion methods. Although the proposed method needs to retrain LS-SVM, the computation time is reasonable due to the very small training set. The classification results of test data are shown in Fig. 2. Apparently the Fisher Criterion is inferior to the other two methods in the case of the small training set.

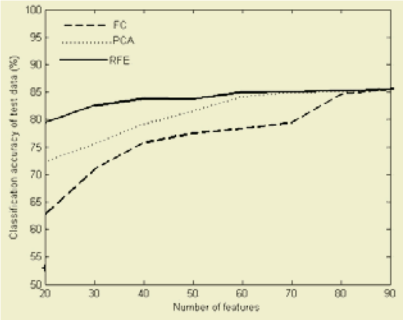


Fig. 2. Classification accuracy of the test data

The proposed method is also compared with the wrapper method which uses genetic algorithms as the search engine. In GA algorithms, each individual in the population represents a candidate solution to the feature selection problem. For the feature space of R^d, the chromosome is represented by a binary vector of dimension *d*. The elements corresponding to the selected features are 1 and otherwise 0. The fitness function is defined as

$$f(x) = err(x) + \lambda \cdot L(x) \qquad (8)$$

where λ is a positive number, *err*(*x*) is the validation error rate of chromosome *x*, and *L*(*x*) is the number of selected features in chromosome *x*. We use a population of 30 individuals, two point crossover, roulette wheel selection and elitism strategy [15]. After 100 iterations, the optimal individual of 41 features reaches 83.55% test data classification accuracy which is close to the 83.89% classification accuracy of the RFE method. The computational complexity of the GA method is very large due to the validation process and the population size.

5 Conclusions

In this paper, we apply the RFE method to selection of ICA features in texture classification. The proposed method was compared with three commonly used feature reduction and selection approaches including PCA, Fish Criterion and genetic algorithms. Our method outperforms the PCA and Fisher Criterion methods in terms of classification accuracy especially in low dimensional spaces. Although GA has close classification accuracy, the high computational cost makes it inefficient in practical application.

Acknowledgements

This material is based upon work supported by the U. S. Army Research Laboratory and the U. S. Army Research Office under contract number DAAD19-03-1-0123.

References

1. Unser, M.: Texture Classification and Segmentation Using Wavelet Frames. IEEE. Trans, Image Processing, **4** (1995) 1549-1560
2. Grigorescu, S.E., Petkov, N. and Kruizinga, P.: Comparison of Texture Features Based on Gabor Filters. IEEE. Trans, Image Processing, **11** (2002) 1160-1167
3. Zeng, X.-Y., Chen, Y.-W., Nakao, Z. and Lu, H.: Texture Representation Based on Pattern Map. Signal Processing, **84** (2004) 589-599
4. Bell, A.J., Sejnowski, T.J.: The Independent Components Nature Scenes Are Edge Filters. Vision research, **37** (1997) 3327-3338
5. Olshausen, B.A., Field, D.J.: Emergence of Simple-cell Receptive Field Properties by Learning a Sparse Code for Natural Images. Nature, **381** (1996) 607-609
6. Hyvarinene, A., Hoyer, P. and Oja, E.: Sparse Code Shrinkage for Image Denoising. Proc. IEEE Int. Conf. Neural Networks (1998) 859-864
7. Randen, T. A., Husoy H.J.: Filtering for Texture Classification: A Comparative Study. IEEE Trans. Pattern Analysis and Machine Intelligence, 21 (1999) 291-310
8. Hyvarinen, A., Oja, E.: A Fast Fixed-point Algorithm for Independent Component Analysis. Neural Computation, **9** (1997) 1483-1492
9. Jenssen, R., Eltoft, T.: ICA Filter Bank for Segmentation of Texture Images. ICA2003 (2003) 827-832
10. Kohavi, R., John, G.H.: Wrappers for Feature Subset Selection. Artificial Intelligence, **97** (1997) 273-324
11. Guyon, I., Weston, J., Barnhill, S. , Vapnik, V.: Gene Selection for Cancer Classification Using Support Vector Machines. Machine Learning, 46 (2002) 389-422
12. Christopher, J.C. Burges: A Tutorial on Support Vector Machines for Patter Recognition. Data Mining and Knowledge Discovery, **2** (1998) 121-167
13. Suykens, A.K., Vandewalle, J.: Least Squares Support Vector Machine Classifiers. Neural Processing Letters, **9** (1999) 293-300
14. Fukunaga, K.: Introduction to Statistical Pattern Recognition. 2nd ed. Academic Press (1991)
15. Davis, L.: Handbook on Genetic Algorithms. Van Nostrand Reinhold (1991)

Feature Selection and Fusion for Texture Classification

Shutao Li[1,2] and Yaonan Wang[1]

[1] College of Electrical and Information Engineering, Hunan University
Changsha, Hunan 410082, China
[2] National Laboratory on Machine Perception, Peking University
Beijing 100871, China
shutao_li@yahoo.com.cn

Abstract. In this paper, a novel texture classification method using selected and combined features from wavelet frame and steerable pyramid decompositions has been proposed. Firstly, wavelet frame and steerable pyramid decompositions are used to extract complementary features from texture regions. Then the number of features is reduced by selection using maximal information compression index. Finally the reduced features are combined and forwarded to SVM classifiers. The experimental results show that the proposed method used selected and fused features can achieve good classification accuracy and have low computational complexity.

1 Introduction

In the past few decades, a large number of texture features have been proposed. Tuceryan and Jain identify these into four major categories, namely, statistical, geometrical, model-based and signal processing features [1]. Recently, one of the major developments in texture classification has been the use of multiresolution and multichannel descriptions. This description provides information about the image contained in time-frequency domain, and thus provides a powerful tool for the description of similar textures. Several multiresolution and multichannel transform algorithms have been used for texture classification [2]. However, most of them were used separately.

In this paper, a novel texture classification method using selected and combined features from wavelet frame and steerable pyramid has been proposed. Firstly, wavelet frame and steerable pyramid are used to extract complementary features. Then the number of features is reduced by selection using maximal information compression index. Finally the reduced features are combined and forwarded to SVM classifiers. The experimental results show that the proposed method used selected and fused features can achieve good classification accuracy and have low computational complexity.

The rest of this paper is organized as follows. The proposed method is described in Section 2. Section 3 describes the experimental setup and results. The last section gives some concluding remarks.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 268–273, 2005.

2 The Proposed Method

The scheme of the proposed method is shown in Fig.1. A texture region is firstly decomposed by wavelet frame and steerable pyramid to extract complementary features. Then the two feature sets are reduced by feature selection phase, which is based on maximal information compression index. Finally the reduced features are combined and forwarded to SVM classifiers. The details of the scheme are described as follows.

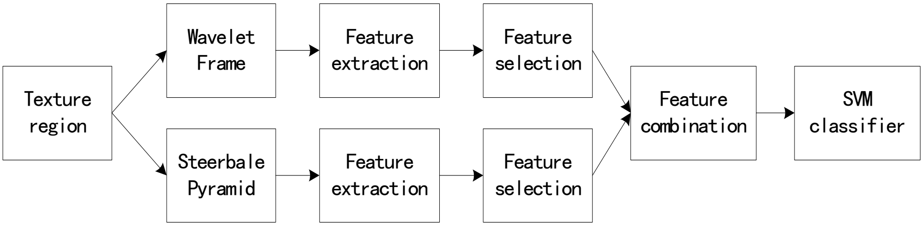


Fig. 1. Scheme of the proposed method

2.1 Wavelet Frame Transform

Unser proposed an overcomplete wavelet representation called a wavelet frame transform (WFT) [3]. The "overcompleteness" is due to the fact that WFT has no dyadic decimation on each decomposition level, shown in Fig.2(a). Avoiding down-sampling guarantees both aliasing free and translation invariant properties. Although the resulting transform is highly redundant from an information theoretic point of view, it is still simple to compute.

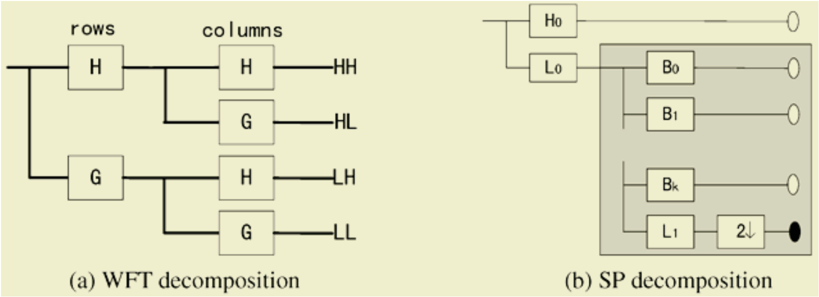


Fig. 2. Decomposition diagram of WFT and SP pyramid

2.2 Steerable Pyramid

The steerable pyramid (SP) is a linear multi-scale, multi-orientation image decomposition method, unlike most discrete wavelet transforms, which is non-orthogonal and overcomplete [4]. Fig.2(b) shows the analysis diagram of steerable pyramid for a single stage. The circles represent the decomposed subband images. The image is first decomposed into lowpass and highpass subbands, using steerable filters L_0 and H_0. The lowpass band continues to be divided into a set of oriented bandpass subbands B_0, ..., B_N and a lower lowpass subband L_1. The lower lowpass subband is subsampled by a factor of 2 along the x and y directions. Repeating the shaded area provides the recursive structure. The steerable pyramid representation is translation-invariant and rotation-invariant.

2.3 Feature Extraction

Based on common belief, the mean and variance of the energy distribution of the multiresolution transform coefficients for each subband at each decomposition level can be used to identify a texture. Let the image subband be $I_i(x, y)$, with *i* denoting the specific subband, the resulting feature vector $f = \{\mu_i, \sigma_i\}$ with,

$$\mu_i = \frac{1}{M \times N} \sum_{x=1}^{M} \sum_{y=1}^{N} | I_i(x, y) | \tag{1}$$

$$\sigma_i = \frac{1}{M \times N} \sum_{x=1}^{M} \sum_{y=1}^{N} | I_i(x, y) - \mu_n |^2 \tag{2}$$

where *M* and *N* is the size of $I_i(x, y)$.

2.4 Feature Selection

Maximal information compression index of random variables x and y is defined as,

$$\begin{aligned}\lambda(x, y) = [&(\mathrm{var}(x)) + (\mathrm{var}(y)) - \\ &\sqrt{(\mathrm{var}(x) + \mathrm{var}(y))^2 - 4\,\mathrm{var}(x)\,\mathrm{var}(y)(1 - \rho(x, y)^2)}\,]/2\end{aligned} \tag{3}$$

where $\rho(x, y) = \frac{\mathrm{cov}(x, y)}{\sqrt{\mathrm{var}(x)\,\mathrm{var}(y)}}$, $\mathrm{var}()$ denotes the variance of a variable and $\mathrm{cov}()$ the covariance between two variables.

If the features are linearly dependent, the value of λ is zero and increases as the amount of dependency decreases.

The feature selection algorithm proposed by Mitra is described as follows [5].

Let the original number of features be D, and the original feature set be $O = \{F_i, i = 1, \cdots, D\}$. Represent the dissimilarity between features F_i and F_j by $S(F_i, F_j)$. Higher the value of is S, the more dissimilar are the features. The maximal information compression index is used to compute S. Let d_i^k represent the dissimilarity between feature F_i and its kth nearest neighbor feature in R. Then

Step 1. Choose an initial value of $k \le D - 1$. Initialize the reduced feature subset R to the original feature set O.

```
Step 2. For each feature $F_i \in R$, compute $d_i^k$.
Step 3. Find feature $F_{i'}$ for which $d_{i'}^k$ is minimum. Retain this feature in $R$ and discard $k$ nearest feature of $F_{i'}$. Let $\varepsilon = d_{i'}^k$
Step 4. If $k > \mathrm{cardinality}(R) - 1$:    $k = \mathrm{cardinality}(R) - 1$.
Step 5. If $k = 1$: Go to step 8.
Step 6. While $d_{i'}^k > \varepsilon$ do
        $k = k - 1,\ d_{i'}^k = \inf_{F_i \in R} d_i^k$
        If $k = 1$: Go to Step 8.
Step 7. Go to step 2.
Step 8. Return feature set as the reduced feature set.
```

2.5 SVM Classifier

The support vector machine (SVM) is a training algorithm for learning classification rules from data [6]. SVMs are based on the structural risk minimization principle. The geometrical interpretation of support vector classification (SVC) is that the algorithm searches for the optimal separating surface, i.e. the hyperplane that is, in a sense, equidistant from the two classes.

Because the number of texture classes is greater than two, we adopt the conventional approach of casting this multi-class classification problem as a number of binary classification problems. In other words, we have one classifier for each texture class, each attempting to separate samples belonging to this class from samples belonging to the other classes. On classifying a new sample, the classifier with the largest output will be selected as the winner, and this new sample is assigned to the winner's corresponding texture class.

3 Test and Results

In this experiment, we use three datasets selected the Brodatz album and the MIT Vision Texture database. The first dataset shown in Fig.3 has 28 textures, which with size of 256×256. This dataset is challenging because there are significant variations within some textures and some of them are very similar to each other. The second and third datasets both have 10 textures, which with size of 128×128, shown in Fig.4 and Fig.5, respectively. For these two groups, due to the inhomogeneity and large variations, texture types in local windows are perceptually close. All the images have been globally equalized prior to being used.

In the experimental setup, we use a complete separation between the training and test sets and repeat the experiment 100 times and compute the average performance. For all the three datasets, the original texture images are subdivided into nonoverlapping samples of size 32×32. 50% of the total samples are used for training and the rest of 50% are used for testing. For WFT, the texture image regions are decomposed with db1 basis and 3 levels. For SP, the decomposition filters are sp5 filters. For the feature selection algorithm, about half of the original features are kept. The numbers of nearest neighbors in selection phase for WFT and SP, k_{WFT} and k_{SP}, are shown in Table 1 too. The Gaussian kernels are used in the SVM, because preliminary results suggested that the Gaussian kernel outperforms the polynomial kernel. The gamma value is set to 1 and C value is 1000.

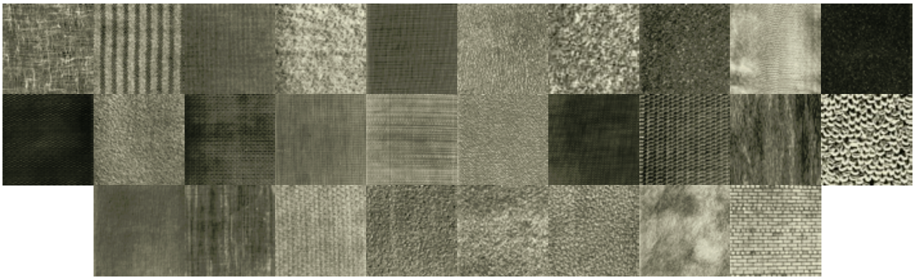


Fig. 3. Texture dataset 1

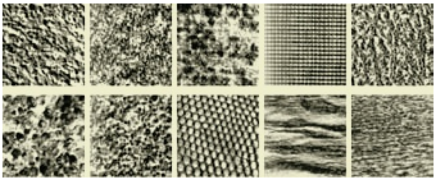


Fig. 4. Texture dataset 2

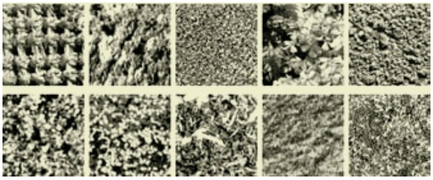


Fig. 5. Texture dataset 3

Table 1. Comparison of different feature sets for the three datasets

	Dataset 1		Dataset 2		Dataset 3	
Feature set	Feature dimension	Accuracy	Feature dimension	Accuracy	Feature dimension	Accuracy
WFT	18	96.20	18	92.49	18	81.25
SP	26	96.58	26	92.63	26	82.64
WFT+SP	18+26=44	98.47	18+26=44	95.25	18+26=44	86.70
WFT+SP with selection	8+12=20 (k_{WFT} =7 k_{SP} =10)	98.62	7+13=20 (k_{WFT} =5 k_{SP} =9)	95.34	7+13=20 (k_{WFT} =7 k_{SP} =10)	88.12

The classification results for the above three datasets using separate features from WFT and SP, the combined features without selection, and selected and combined features shown in Table 1. From Table 1, we can see that for the overcomplete features can achieve significantly better result than those using separated features. The number of selected and fused feature set is less than half of that of the overcomplete feature set, but they give higher accuracy than the overcomplete feature set.

4 Conclusions

In this paper, a novel texture classification method using selected and combined features from wavelet frame transform and steerable pyramid decompositions has been proposed. Firstly, wavelet frame transform and steerable pyramid decompositions are used to extract complementary features from texture regions. Then the number of features is reduced by selection using maximal information compression index. Finally the reduced features are combined and forwarded to SVM classifiers. The experimental results show that the proposed method used selected and fused features can achieve good classification accuracy and have low computational complexity.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No.60402024).

References

1. Tuceryan, M., Jain, A.K.: Texture Analysis. In: Chen C.H., Pau, L.F., and Wang P.S.P. (eds.): Handbook of Pattern Recognition and Computer Vision, 2nd ed., World Scientific, Singapore (1998)
2. Randen, T., Husøy, J.H.: Filtering for Texture Classification: A Comparative Study. IEEE Trans Patt. Anal. Mach. Intell. **21** (1999) 291–310
3. Unser, M.: Texture Classification and Segmentation Using Wavelet Frames. IEEE Trans Image Proc. **11** (1995) 1549–1560
4. Simoncelli, E.P., Freeman, W.T.: The Steerable Pyramid: A Flexible Architecture for Multi-Scale Derivative Computation. Proc. of IEEE 2nd Int. Conf. on Image Proc., Washington DC (1995) 444–447
5. Mitra, P., Murthy, C.A., Pal, S.K.: Unsupervised Feature Selection using Feature Similarity. IEEE Trans. Patt. Anal. Mach. Intell. **24** (2002) 301–312
6. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines. Cambridge University Press, Cambridge (2000)

Scene Classification Using Adaptive Processing of Tree Representation of Rectangular-Shape Partition of Images

Wei Sun, Ken Lo, and Zheru Chi

Department of Electronic and Information Engineering
The Hong Kong Polytechnic University, Hong Kong, China
enweisun@eie.polyu.edu.hk

Abstract. Image classification is very helpful for organizing large image databases and content based image retrieval (CBIR). However, it is very complex and challenging because of lacking effective methods. In this paper, we present a tree representation of images based on rectangular-shape partition. Then an adaptive processing algorithm is adopted to perform the classification task. Experimental results on seven categories of scenery images show that the structural representations are better than the traditional methods and our previous work based on quadtree representation of fixed partition.

1 Introduction

Image content representation has been a popular research area in various image processing applications for the past few years. Some systems attempted to represent image contents in terms of a series of keywords. However, keyword-based taxonomy reduces the system complexity while increasing the requirements of users' knowledge. On the other hand, much research has been conducted on representing image contents with visual features. Most of the applications represent images using low level visual features such as color, texture, shape and spatial layout in a very high dimensional feature space, either globally or locally. However, the most popular distance metrics such as Euclidean distance cannot guarantee that the contents are similar even their visual features are very close in the high dimensional space.

Most recently, region-based representation of images potentially offers an attractive solution to this problem. It involves a number of regions that can be considered as a first level of abstraction with regard to the raw information. Many evidences suggest that image content with region-based representation would be encoded by structuring the regions within a tree representation. In [1] we presented promising results of image classification and image retrieval with a segmentation free quadtree representation due to the challenges in image segmentation. It is simple and straightforward, but there is little meaning attached to each tree node at all, and it may be difficult to improve its performance further.

Both regions or objects and the spatial relationship among them play more important roles in characterizing image contents, because they convey more semantic meaning. By organizing these meaningful regions into tree representation, image contents can be represented more comprehensively at various details, which will be helpful for image classification and image retrieval. Recent approaches to represent images in

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 274–280, 2005.

tree representations can be found in the past decade [2-4]. Salembier and Garrido [5] have also discussed the interest of BSP tree structure as a region-oriented image representation. The back-propagation through structure (BPTS) algorithm can be used in learning tree structure representation or image retrieval with relevance feedback.

This paper is organized as follows: the basic idea of the tree structure representation is described in Section 2. Section 3 presents the adaptive processing by the neural network. Section 4 gives the results of the experiments. Finally, the conclusion is drawn in Section 5.

2 Structural Image Representations

Quadtree is one of the most popular methods decomposing images into blocks. By a pyramidal partition, image contents can be shown from coarse to fine. If each tree node is organized in order, it can be noticed that the quadtree representation presents spatial information implicitly.

There are few semantic attributes attached to each tree node at all because the quadtree decomposition is very simple. It is obvious that regions obtained by segmentation contain more semantic attributes, but one must face the challenging segmentation.

Therefore, we represent an intermediate method to improve the efficiency of the representation. It is to make rectangular-shape partition. Every time we separate the current region to two parts if it is not homogeneous enough. One of the parts will be a rectangle, and we let the upper left corner of the whole region be a corner of the first rectangle. Because it will be a corner of a rectangle eventually, the restriction will not do any harm to the partition. Repeat the split until all parts of the image are homogeneous. Figure 1 shows an example.

We need a parameter of a region, which is named "entropy", to be the homogeneity measure of that region. For example, we calculate the entropy in N gray levels. Assume p(i), i = 0,1,…,N is the number of the pixels whose gray level is i, we define

$$E = \sum_{i=0}^{N} - \frac{p(i)}{n} \log_2 \left[\frac{p(i)}{n} \right]. \qquad (1)$$

where n is the number of pixels in the whole region.

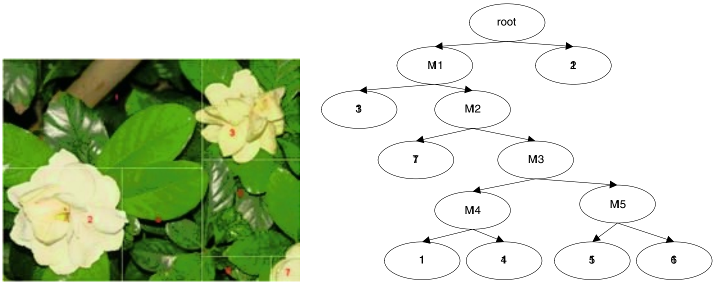


Fig. 1. Binary tree representation of a rectangular-shape partitioned image

Obviously, it means that the region is homogeneous when E is small. Especially if all the pixels in the region are the same, E equals to 0. So the entropy is non-negative. While E is quite a large value, we know that the region is not so homogeneous. When E is less than a predetermined value E_0 in the actual processing, we consider the region to be homogeneous enough and vice versa.

We will find that there are some regions which are very similar but separated, especially when we define the 'homogeneity' quite strongly. For example, regions 5 and 6 in Figure 1 are such regions. Clustering is also utilized to solve this issue. At first, images are separated into small parts to ensure every region is homogeneous. Secondly, these regions are clustered into possible classes such as leaves and flowers. Then they will be merged in next step. Finally binary tree representation is established.

It is well known that visual features, color, texture and shape are very important to characterize the image contents. Each node of the tree can be considered as an image and visual features will be extracted from it. So far, there are many approaches to extract these visual features in the literatures. In our experiments, 12 dimensional visual feature vectors, which are of color, textures, and spatial information, are employed to characterize the attributes of each tree node. The color features include the number of color, the percentages of the most three dominant colors in RGB color space, and the average value of each color component. The average pixel luminance and the standard deviation of pixel luminance are employed to characterize the node texture simply. The spatial information is the center coordinates of each tree node and the normalized node area in terms of image size. The first 9 dimensions only are used for clustering.

3 Adaptive Processing of Data Structure

Connectionlist models have been successfully employed to solve learning tasks characterized by relatively poor representations in data structure such as static pattern or sequence. Most structured information presented in the real world, however, can hardly be represented by simple sequences. Although many early approaches based on syntactic pattern recognition were developed to learn structured information, devising a proper grammar is often a very difficult task because domain knowledge is incomplete or insufficient. On the contrary, the graph representation varies in the size of input units and can organize data flexibly. Its encoding process shown in Figure 3 is independent on the classification task and it is not necessary to pay attention specifically to their nodes. Recently, neural networks for processing data structures have been proposed by Sperduti [3]. It has been shown that they can be used to process data structures using an algorithm namely back-propagation through structure (BPTS). The algorithm extends the time unfolding carried out by back-propagation through time (BPTT) in the case of sequences. A general framework of adaptive processing of data structures was introduced by Tsoi [6] and Frasconi et al. [4].

Considering a generalized formulation of graph encoding shown in Figure 2, we have

$$\mathbf{x} = \mathbf{F}_n\left(\mathbf{A}q^{-1}\mathbf{y} + \mathbf{B}\mathbf{u}\right). \tag{2}$$

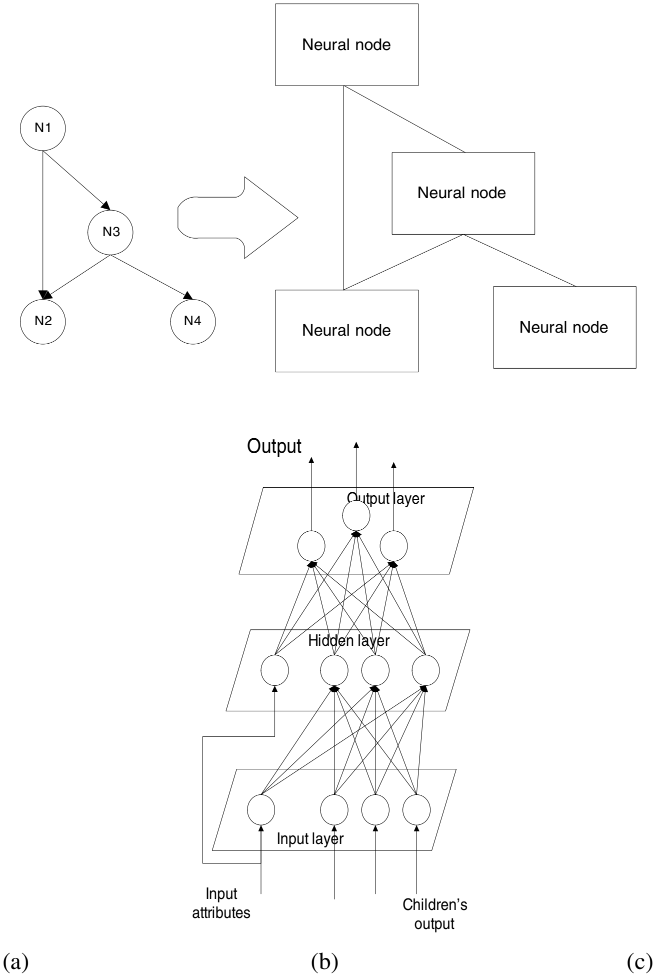


Fig. 2. An illustration of a data structure with its nodes encoded by a single-hidden-layer neural network. (a) A Directed Acyclic Graph (DAG); (b) The encoded DAG; (c) The illustration of the neural network.

$$\mathbf{y} = \mathbf{F}_p\left(\mathbf{Cx} + \mathbf{Du}\right). \tag{3}$$

where **x**, **u** and **y** are respectively the n dimensional output vector of the n hidden layer neurons, the m dimensional inputs to the neurons, and the p dimensional outputs of the neurons. q^{-1} is merely a notation to indicate that the input to the node is taken from its children. The matrix A is defined as following:

$$\mathbf{A} = \left[\mathbf{A}^1 \quad \mathbf{A}^2 \quad \cdots \quad \mathbf{A}^c\right]. \tag{4}$$

where c is the maximal out degree of the graph. $\mathbf{A}^i$, $i = 1,2,...,c$ is an $n \times p$ matrix, and is formed from the vector a_j^i, $j = 1,2,...,n$. **A** is a $c \times (n \times p)$ matrix. And **B**, **C**, and **D** are respectively matrices of dimensions $n \times m$, $p \times n$ and $p \times m$. $\mathbf{F}_n(\bullet)$ is an *n* dimensional vector given as:

$$\mathbf{F}_n(\alpha) = [f(\alpha) \quad f(\alpha) \quad \cdots \quad f(\alpha)]^T \,. \tag{5}$$

where $f(.)$ is a nonlinear function such as a Sigmodal function.

Note that we have assumed only one hidden layer in the formulation, because a single hidden layer with sufficient number of neurons is a universal approximator.

The training process is to estimate the parameters **A**, **B**, **C**, and **D** from a set of input/output samples by minimizing the cost criterion:

$$J = \frac{1}{2} \sum_{i=1} N_T \| \mathbf{d}_i - \mathbf{y}_i \|^2 \,. \tag{6}$$

where $\mathbf{d}_i$ denotes the output of the root of the *i*-th sample, $\mathbf{y}_i$ denotes the desired output of the *i*-th sample, and N_T is the number of the samples. The derivation of the training algorithm minimizing the cost criterion will follow a fashion similar to gradient learning by computing the partial derivation of the cost *J* with respect to **A**, **B**, **C**, and **D**. The deep tree structure will introduce long dependence problem during this gradient learning method [7].

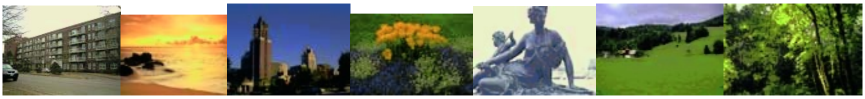


Fig. 3. Sample images of the 7 categories used in our experiments: apartment, beach, campus, flower, statue, grass and tree

4 Experiments and Discussions

Our study was conducted on our collected scenery images of seven categories: apartment, beach, campus, flower, statue, grass and tree as shown in figure 3. 40 sample images were collected for each category. Half of the images were used for training and the other images were for test. We tried 3 methods to do this work. The first one is mentioned in Section 2 and 3. The second is the same to the first one except that the fixed line-partition and quadtree are used instead of rectangular-shape partition and binary tree representation [1]. For comparing to the traditional methods, we test the classification method 3 based on the histogram similarities between the test image and each category. The similarity is calculated by follows:

$$D(k) = \sqrt{\sum_{i=0}^{N}\left[h(i) - \frac{1}{m_k}\sum_{j=1}^{m_k} h_k(j,i)\right]^2} \;. \tag{7}$$

where N is the number of gray levels, h(*i*) is the *i*-th histogram element of test image, $h_k(j,i)$ is the *i*-th histogram element of the *j*-th image in category k, and m_k denotes the image counts in category k. D(k) means the distance between the test image and the *k*-th category, it is the Euclidean distance. We arrange the image into category k when the corresponding D(k) is a minimum.

Results are listed in Table 1.

Table 1. Classification results by different methods

Methods	**Training Set(%)**	**Testing Set(%)**
Rectangle-partition	97.39	84.03
Fixed line-partition	96.08	79.38
Histogram	N/A	71.25

It is observed that first two methods based on tree representation achieves better classification performance, and the first approach (rectangle partition) performs best. It indicates that the rectangular-shape partition conveys more semantic information.

5 Conclusions and Future Work

We present a scenery image classification method using a rectangle-partition binary tree representation and a Backpropagation Through Structure (BPTS) training algorithm. The binary tree representation is constructed by merging the regions of some rectangles in a bottom up way. Only seven simple visual features are used to characterize each rectangle. It is shown that the rectangle region-based binary tree representation contains much more semantic content than the fixed partition quadtree representation and the histogram, which is observed in our experiments of classification of scenery images.

Since it is a region-based representation, it must depend on the partition methods. It seems that segmentation-based method will achieve better performance, but there is not effective general segmentation approach now. In the other hand, it is quite time-consuming. However, in our future study, the effects of different partition or segmentation approaches will be investigated. Better methods to establish the tree structure will also be studied in the future. For instance, the tree will not be restricted in a binary tree.

Acknowledgements

The work described in this paper was fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project no. PolyU B-Q705).

References

1. Wang, Z., Chi, Z., Feng, D., Cho, S.Y.: Adaptive Processing of Tree-Structure Image Representation. IEEE Pacific-Rim Conference on Multimedia (PCM2001), Beijing (2001) 989-995
2. Goller, C., Kuchler, A.: Learning Task-Dependent Distributed Representations by Back-Propagation through Structure. In IEEE International Conference on Neural Networks (1996) 347-352
3. Sperduti, A., Starita, A.: Supervised Neural Networks for The Classification of Structures. IEEE Trans. on Neural Networks, **8** (1997) 714-735
4. Frasconi, P., Gori, M., Sperduti, A.: A General Framework for Adaptive Processing of Data Structures. IEEE Trans. on Neural Networks, **9** (1998) 768-785
5. Salembier, P. , Garrido, L.: Binary Partition Tree as an Efficient Representation for Image Processing, Segmentation, and Information Retrieval. IEEE Trans. on Image Processing, **9** (2000) 561-576
6. Tsoi, A.C.: Adaptive Processing of Data Structures: an Expository Overview and Comments. Technical Report, Faculty of Informatics, University of Wollongong, Australia (1998)
7. Cho, S., Chi, Z., Wang, Z., Siu, W.: An Efficient Learning Algorithm for Adaptive Processing of Data Structure. Neural Processing Letters, **17** (2003) 175-190

Shape Recognition Based on Radial Basis Probabilistic Neural Network and Application to Plant Species Identification

Jixiang Du[1,2], Deshuang Huang[1], Xiaofeng Wang[1], and Xiao Gu[1]

[1] Intelligent Computing Lab, Institute of Intelligent Machines, Chinese Academy of Sciences
Hefei, Anhui 230000, China
{du_jx,dshuang,xfwang,xgu}@iim.ac.cn
[2] Department of Automation, University of Science and Technology of China

Abstract. In this paper, a novel shape recognition method based on radial basis probabilistic neural network (RBPNN) is proposed. The orthogonal least square algorithm (OLSA) is used to train the RBPNN and the recursive OLSA is adopted to optimize the structure of the RBPNN. A leaf image database is used to test the proposed method. And a modified Fourier method is applied to descript the shape of the plant leaf. The experimental result shows that the RBPNN achieves higher recognition rate and better classification efficiency with respect to radial basis function neural network (RBFNN), BP neural network (BPNN) and multi-Layer perceptron network (MLPN) for the plant species identification.

1 Introduction

The shape recognition is an important part of machine intelligence that is useful for both decision-making and data processing. More importantly, the recognition based on shape feature is also a central problem in those fields such as pattern recognition, image technology and computer vision, etc., which have been received considerable attention recent years. In addition, Plant is a greatly significant thing for human's living and development. It is necessary to correctly evaluate intra- and inter-specific variations for the efficient collection and preservation of genetic resources. Leaf shape is one of the characteristics to be evaluated and that plays an important role in the classification process. Therefore, it is necessary to develop a method that can easily measure and quantitatively evaluate leaf shape. This paper focuses on using a novel radial basis probabilistic neural network (RBPNN) model to perform plant recognition task through modified Fourier descriptors of leaf shapes.

2 Radial Basis Probabilistic Neural Network (RBPNN) Model

The RBPNN model [1] as shown in Fig.1 was derived from the radial basis function neural network (RBFNN) and the probabilistic neural network (PNN). Hence it possesses the advantages of the above two networks while lowers their demerits.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 281–285, 2005.

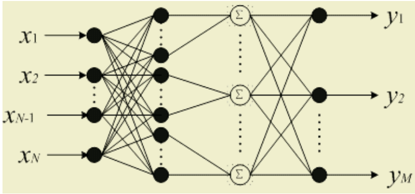


Fig. 1. The topology scheme of radial basis probabilistic neural network

In mathematics, for input vector x, the actual output value of the ith output neuron of RBPNN, y_i^α, can be expressed as the following equation:

$$y_i^\alpha = \sum_{k=1}^{M} w_{ik} h_k(x). \tag{1}$$

$$h_k(x) = \sum_{i=1}^{n_k} \phi_i\left(\|x - c_{ki}\|_2\right) \quad k = 1,2,\cdots M. \tag{2}$$

Here $h_k(x)$is the kth output value of the second hidden layer of RBPNN; $\phi_i(\bullet)$ is the kernel function, which is generally Gaussian kernel function and can be written as

$$\phi_i\left(\|x - c_{ki}\|_2\right) = \exp\left(-\frac{\|x - c_{ki}\|_2^2}{\sigma_i^2}\right). \tag{3}$$

The purpose of training the network is to assure the synaptic weight to change along with the direction of minimizing the squared error between the teacher signal and the actual output. Generally, the training algorithms for the RBPNN include orthogonal least square algorithms (OLSA), recursive least square algorithms (RLSA), etc. These two methods have the common advantages of fast convergence and good convergent accuracy. The RLSA, which requires good initial conditions, however, is to fit for those problems with the large training samples set. As the OLSA make full use of matrix computation, such as orthogonal decomposition algorithm of matrices, its training speed and convergent accuracy is faster and higher than the ones of the RLSA. So the OLSA is preferred in this paper and the details of it can be found in literature [3].

Two methods of the structure optimization of RBPNN can be found in literature [2],[3]. An optimization method based on genetic algorithms (GA) was proposed on recursive orthogonal least square algorithm (ROLS) was proposed in literature [5]. Compared with ROLS, GA is a global search method, and it can usually obtain a more computations and takes a longer training time. On the other hand, the ROLS is a backward selection algorithm. The philosophy of this method is to sequentially remove from the network, one hidden node at a time, i.e., the center that causes the smallest increase in training error. The details of ROLS method have been described in literature [3]. In this paper the ROLS is preferred.

The key point of optimizing the RBPNN is to select the hidden center of the first hidden layer of the RBPNN is not only involved in how many the number of the hidden centers being selected, but also in what space locations the hidden centers being located at. Usually, we wish the number of selected centers to be as small as possible for the fewer hidden center will not only simplify the training and the testing of the network but also improve the generalization capability of the network. On the other hand, the locations of the hidden centers in space are of utmost importance to the performance of the network, In the case of the number of the hidden centers being fixed, different locations for the hidden centers can lead to different network performance. In this paper, the ROLS is used to select the hidden centers of the first hidden layer of the RBPNN and to optimize the structure of the RBPNN.

3 Image Analyses and Fourier Transform

Image processing and analysis has been developed quickly since 60s. Now it has been widely used in biomedical, remote sensing, documents, etc. Our present image processing and analysis is another new application: plant species identification.

It is well known that Fourier transform is a right candidate to describe smooth change and sudden changes in its frequency domain. Theoretically the FD [6] are invariant to translation, rotation, scaling and starting point, actually it is sensitive to rotation, scaling due to the complicated calculating error accumulation and the image changes in the rotation and scaling transforms. Based on our practical issue, we propose a modified way to apply Fourier transform to the leaf shape analysis. The modification is as follow:

$$u^{'}(n)=d(n) \qquad n=0,1,2,\ldots,N-1. \tag{4}$$

$d(n)$ is the normalized distance between the shape center and the point at the contour. Its Fourier transform is:

$$\begin{aligned} u^{'}(n)&=\frac{1}{N}\sum_{k=0}^{N-1}a^{'}(k)\exp(\frac{j2\pi kn}{N}) \qquad 0\le n\le N-1,\\ a^{'}(k)&=\sum_{k=0}^{N-1}u^{'}(n)\exp(-\frac{j2\pi kn}{N}) \qquad 0\le k\le N-1. \end{aligned} \tag{5}$$

$$v(k)=\frac{1}{1+\sqrt[3]{k^{2}}\,\exp\left(-k\left|a^{'}(k)\right|/a^{'}(0)\right)} \qquad 1\le k\le N-1. \tag{6}$$

$v(k)$ is a nonlinear transformation of $a^{'}(k)$ and is called Modified Fourier Descriptors (MFD) of the contour. The original Fourier Descriptors (FD) [6] are much more sensitive to scaling and rotation than the Modified Fourier Descriptors. But they both are not sensitive to translation. So the Modified Fourier Descriptors should be are more suitable to our leaf shape analysis.

4 Experimental Results

The leaf image database used in the following experiment includes 20 species of different plants. Each species includes at least 40 leaves images, 20 of which are used as training samples. There are totally 1540 images with the database.

First, we select 430 training samples as the hidden centers of the first hidden layer. The number of the second hidden neurons is set as 20. The number of output layer neurons is also set as 20. The OLSA is used to train the RBPNN. Consequently, the recognition rate of the testing samples is 96.2% (as shown in Table.1).

In order to optimize and prune the RBPNN, likewise, by using the parameter similar to the one mentioned above we use the recursive orthogonal least squares algorithm to optimize the structure of RBPNN. As a result, the selected hidden centers number of the first hidden layer is reduced from 400 to 132 and the recognition rate of testing samples is 96.1%.

Compared with the RBPNN, with the same training and testing samples, by selecting all the training samples as the hidden centers, the maximum recognition rate of RBFNN is 94.9%. The recognition rate of BP neural network (BPNN) [4] to this data is 94.3%, in which the number of the hidden neurons is set as 40. The recognition rate of multi-Layer perceptron network (MLPN) [4] to this data is 94.4%, in which the number of the first hidden neurons is also set as 40 and the ones of the second is set to 20. Thus, it can be seen that the classification performance of RBPNN is higher than that of the RBFNN, BPNN and MLPN. The k-NN (k=1and 4) classifier and MMC classifier [5] is also used to compare with the RBPNN with the same training and testing samples. The recognition rates of these classifiers are lower than that of RBPNN.

On the other hand, it was found that the training speed and testing speed with the RBPNN are also very fast in experiment. The algorithm was programming with Matlab 6.5, and was run on Pentium 4 with the clock of 2.6GHz and the RAM of 256M under Microsoft Windows XP environment. The result is shown in Fig.4 and Table.1. The CPU time recognizing one image is about 0.006 seconds and the training CPU time is about 48 seconds. For other neural network classifiers, the training time and classification time are longer than the ones of the RBPNN.

From the above experimental results, it can be observed that our shape recognition method based on the RBPNN not only achieved higher statistical recognition rate but also are of faster training speed and testing speed.

Table 1. Performance comparison between RBPNN and other classifiers

Classifiers	Recognition Rate (%)		CPU time (Second)	
	MFD	FD	Training	Classification
RBPNN(50-400-20-20)	96.2	94.1	48	0.006
RBFNN(50-400-20)	94.9	93.1	53	0.008
BPNN(50-40-20)	94.3	92.5	198	0.082
MLPN(50-40-20-20)	94.4	92.8	272	0.012
1-NN	92.6	90.7	\	23
4-NN	90.9	89.6	\	136
MMC	91.6	90.2	25	0.015

5 Conclusions

This paper proposed a shape recognition method based on radial basis probabilistic neural networks (RBPNN) and applied to plant identification successfully. The orthogonal least square algorithm (OLSA) is used to train the RBPNN and the recursive OLSA is adopted to optimize the structure of RBPNN. Our method achieves higher recognition rate and faster training speed through the modified Fourier descriptors. The experimental results obtained show that our approaches are effective, efficient and feasible, which greatly support the claim that RBPNN is a very promising neural network model in practical applications. Future research works will focus on how to use some effective feature extraction methods together with our RBPNN classifier to achieve better recognition performance in shape recognition and other practical applications.

References

1. Huang, D.S.: Radial Basis Probabilistic Neural Networks: Model and Application. International Journal of Pattern Recognition and Artificial Intelligence, **13** (1999) 1083-1101
2. Zhao, W.B., Huang, D.S.: The Structure Optimization of Radial Basis Probabilistic Networks Based on Genetic Algorithm. IJCNN2002, Hilton Hawaiian Village Hotel, Honolulu, Hawaii (2002) 1086-1091
3. Zhao, W.B., Huang, D.S.: Application of Recursive Orthogonal Least Squares Algorithm to The Structure Optimization of Radial Basis Probabilistic Neural Networks. ICSP 2002, Beijing, China (2002) 1211-1214
4. Huang, D.S.: Systematic Theory of Neural Networks for Pattern Recognition. Publishing House of Electronic Industry of China, Beijing (1996)
5. Wan, Y. Y., Du, J. X., Huang, D. S., Chi, Z., Cheung, Y. M., Wang, X. F., Zhang, G. J.: Bark Texture Feature Extraction Based on Statistical Texture Analysis. ISIMP2004, Hong Kong, China (2004) 482-485
6. Rui, Y., She, A.C. Huang, T.S.: Modified Fourier Descriptors for Shape Representation-A Practical Approach. First International Workshop on Image Databases and Multi Media Search, Amsterdam, The Netherlands (1996)

Image Recognition Using Synergetic Neural Network

Shuiping Gou and Licheng Jiao

Institute of Intelligent Information Processing Xidian University
Xi'an, Shaanxi 710071, China
shpgou@mail.xidian.edu.cn

Abstract. A method for texture image recognition using Synergetic Neural Network (SNN) [1] technique is presented. The method combines Immune Clonal Strategy (ICS) [2] with fuzzy clustering to train the prototype vectors in SNN, which is used to classify object images into groups. Simulation results show the proposed algorithm not only reduces complexity of computing but also improves the image recognition performance of SNN. Moreover, the discussion has been made of multi-class recognition using SNN in this paper.

1 Introduction

Image recognition is the one of the most important subject in the image processing field and it is the significant embranchment of artificial intelligence. In this paper synergetic neural network (SNN) is acted as image object recognition classifier. In Synergetics, high dimension and nonlinear problem can be described as a set of low-dimension nonlinear equations, which focuses on the research of the qualitative change of macro-feature in complex system. Dominated principle and order parameter is introduced to describe order macro action in the macro-evolvement process of the system. In the late 1980s, Haken extended synergetic theory to recognition and computer science [1]. Later, he generalized the theory to the area of pattern recognition and put forward the new theory of neural network, synergetic neural network (SNN), used in pattern recognition in 1990s. The SNN was constructed from up to down, whose merit did not produce the pseudo-state. In addition, NN based on evolutionary learning has become hotspot gradually in the field of intelligent computation [3]. Furthermore, In order to enhance the diversity of the population in GA and avoid prematurity, based on the antibody clonal select mechanism of artificial immune system, Immunity Clonal Strategy [2], a new artificial intelligence method, is proposed by Du, H.F. and jiao, L.C. etc. On the basis of the evolutionary algorithms, the method introduces avidity, clonal and mnemon mechanism while adopts corresponding operator to make it convergence quickly to the global optimal value.

Recently, real image recognition based on evolutionarily synergetic neural network has already gained more and more attention by many researchers. An improved learning algorithm of synergetic neural network based on clustering was proposed for more accurate image recognition [4]. But the clustering algorithm is very sensitive to initial value and it gets into easily local minimum. So, we use the immune clonal strategy (ICS) and fuzzy clustering to train SNN, which improves classifying accuracy of the object image.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 286–291, 2005.

2 Synergetic Neural Network

The basic principle of SNN is that the pattern recognition procedure is consistent with the dynamic process [2]. q , a pattern remained to be recognized, is constructed by a dynamic process which translates q into one of prototype pattern vectors v_k through status $q(t)$, namely, this prototype pattern is closest to $q(0)$. The process is described in the following dynamic equation:

$$\dot{q} = \sum_{k=1}^{M} \lambda_k (v_k^+ q) v_k - B \sum_{k \neq k'} v_k (v_{k'}^+ q)^2 (v_k^+ q) - Cq(q^+ q) + F(t) . \tag{1}$$

Where q is the status vector of input pattern with initial value q_0, λ_k is attention parameter. If and only if λ_k is positive, patterns can be recognized. v_k is prototype vector and v_k^+ is its adjoint vector that satisfies $(v_k^+, v_{k'}) = v_k^+ v_{k'} = \delta_{kk'}$. The second term on the right of equation (1) plays a patterns recognition role in defining the potential surface. The second term defines the competition among prototypes and controls the location of ridges in the potential surface. B and C are the constant. Based on the idea of synergetics the pattern recognition procedure can be viewed as the competition process of many order parameters, whose dynamic equation is as follows:

$$\dot{\xi} = \lambda_k \xi_k - B \sum_{k' \neq k} \xi_{k'}^2 \xi_k - C(\sum_{k'=1}^{M} \xi_{k'}^2) \xi_k . \tag{2}$$

Where ξ_k satisfies initial condition:

$$\xi_k(0) = v_k^+ \cdot q(0) . \tag{3}$$

Haken has proved that when $\lambda_k = c, c > 0$, the largest initial order parameter will win and the network will then converge. In this case, the SNN can meet some real-time needs for it avoids iteration [2].

3 Immunity Clonal Strategy Algorithm

In order to enhance the diversity of the population in GA and avoid prematurity, Immunity Clonal Selection is proposed. The clonal operator is an antibody random map induced by the affinity. The state transfer of antibody population is denoted as follows:

$$C_{MA} : A(k) \xrightarrow{clone} A'(k) \xrightarrow{mutation} A''(k) \xrightarrow{selection} A(k+1)$$

According to the affinity function f(*), a point $a_i(k) \in A(k)$ in the solution space will be divided into qi different points $a_i'(k) \in A'(k)$, by using clonal operator, a new antibody population is attained after performing clonal mutation and clonal selection. In fact, the process includes three operating steps: clone, clonal mutation, and clonal selection. Here antibody, antigen, the avidity between antibody and antigen are similar to the definition of the objective function and restrictive condition, the possible solution, match between solution and the fitting function In Artificial Intelligent Computation respectively [5].

It can be realized that the essence of immune clone, which is that clonal operator can produce a variation population around the optimal of parents according to their affinity, and then the searching area is enlarged which is helpful for avoiding prematurity in course of evolution Style information.

4 Learning Algorithm of SNN Based on Immunity Clonal Clustering

In SNN, learning is also the training and modification process on weight matrix. Its learning problem can be reduced to how to get prototype pattern vector v_k and adjoint vector v_k^+ [4]. In this paper, we adopt Immunity Clonal Fuzzy *C*-mean Clustering (*ICFCM*) to obtain prototype pattern vector, where the FCM is the typical fuzzy clustering algorithm based on object function and its essential is to get minimum of objective function, namely, $J_m(U,V)$ [6]. Where *U* is the partition matrix and V is clustering center matrix. The FCM adopts to iterate climbed method and its search is based on grads descend, and FCM is very sensitive to initial value and it get into easily local minimum. However, in the immune clone strategy, clonal operator can produce a variation population around the optimal of parents according to their affinity, and then the searching area is enlarged which is helpful for avoiding prematurity in course of evolution. Then, in this paper, to avoid the singleness of initial prototype of the FCM, we introduce Immune Clonal Strategy into FCM, namely, CFCM. The CFCM not only can overcome FCM depending greatly on initial value of clustering parameters but also can slow down local search.

And the clustering center from CFCM is used as prototype pattern vector of training samples in SNN, which classify data sets. In addition, feature vectors extracted from images are treated as samples of the FCM and the optimal process is based on Immune Clonal Strategy, which gets prototype vector in SNN and identifies the image. Image recognition algorithm using SNN based on immune clonal clustering is presented:

Step 1. Initialize the parameters

Set a small positive number, ε, the number of clustering c and the number of iteration L of FCM in every generation. Randomly generate individual, initializing *c*-partition *U* (0), and get the population with size of pop_size. Then given degree of fuzzy m and suppose generation *t* is 1.

Step 2. Repeat and produce population

For every individual *U* (0) using the FCM algorithm calculating alternately the partition matrix U and clustering center V, i.e., prototype pattern vector, to ob tain objective function J_m of every individual, where the number of iteration is L, the sample data is feature vectors extracted from image.

```
        Set affinity $f_k = \frac{1}{J_m(k)+1}$
          $k = k+1$
       Until $k$=pop_size
```

Step 3. Compute the affinity according to $f(t) = \max_{k=1}^{n} f_k$.

```
       If $|f(t) - f(t-1)| < \varepsilon$
          The best individual, the partition matrix U, of obtained is the optimum of this algorithm. Go to step9 and then terminate the algorithm.
       Else
          t=t+1
       End
```

Step 4. Clone population in generation t and select better individual in sequence to form new population.

$$U'(K) = \Theta(U(K)) = [\Theta(\mu_1(k)), \Theta(\mu_2(k)), \cdots, \Theta(\mu_N(k))]^T$$

Where N is clonal scale.

Step 5. Clonal mutation $U''(K) = T_m^c(U'(k))$

Step 6. Calculate the affinity: $U''(k)$: $\{f(U''(K))\}$

Step 7. Clonal selection: $U''(K+1) = T_s^c(U''(k))$

Select the best individual with better affinity into new parent population

Step 8. Update population and return step2

Step 9. Use the result of step7 to get prototype pattern vector. Then solving its pseudo-inverse to get adjoint vector, and image pattern is recognized by equation (3).

5 Experiment Results and Analysis

Experiment 1. Image collection is composed of 16 texture images from Brodatz texture database, as shown in Fig.1. Each of the images is cropped into 25 non-overlap sub-images with size of 128×128. The 400 images are decomposed into 3 levels by undecimated wavelet transform. The wavelet coefficient energy at different levels and different subbands is treated as texture features. The dimension of the feature is 10:

$$\{e_{LL-1,}e_{LH-1,}e_{HL-1,}e_{HH-1}, e_{LH-2,}e_{HL-2,}e_{HH-2,}e_{LH-3,}e_{HL-3,}e_{HH-3}\}$$

We select 2 training data and 23 test data from each class respectively. One-Against-One (OAO): [7], is adopted in this experiment. And the OAO can divide multi-class problem into the number of k∗(k-1)/2 two-class problem, where k is the number of classes of the data sets. The parameters of the experiment are defined as: the size of initial population is 10, mutation probability 0.1 and the clonal scale 5.

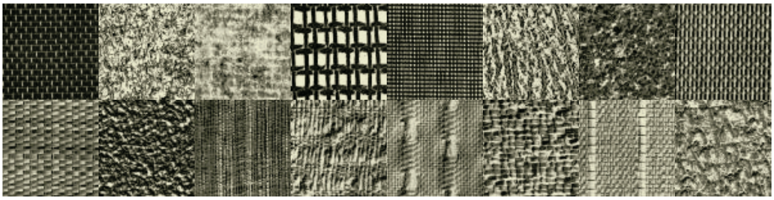


Fig. 1. 16 texture images from Brodatz texture library

L, the iteration generation of the FCM is specified 6. And the termination condition is the iterative generation up to 50. The average image recognition results of 20 times are shown in Table 1, in which substitutes SVM+OAO for Support Vector Machines classification method with one-against-one strategy, the rest three method are used for training prototype vector in SNN then image is recognized by equation (3), where image is original image feature vectors, ICFCM is the presented algorithm in this paper and ICFCM+OAO supplement the one-against-one strategy. In addition, the string 'ARR' replaces the average recognition rate.

Table 1. Comparison of recognition results by several methods

	image	ICFCM	ICFCM+OAO	SVM+OAO
Training time (s)	0.037	0.063	0.219	0.015
test time (s)	0.078	0.078	0.078	0.016
ARR of training sets	87.5%	91.2%	100%	100%
ARR of test sets (%)	66.25	81.25	99.77	97.5

From Table 1 it is obvious that the ICFCM has much higher recognition accuracy rate than the original image pixel and the SVM. But training time is slight long, the reason that is the one-against-one strategy adds computation times. Fortunately, the scale of training samples requires be small and recognition speed of SNN is quick. On the other hand, the recognition accuracy of SNN is unsatisfactory without the strategy of OAO, since this experiment belongs to the recognition of multi-class objects. For recognition with SNN based on equation (3), requires correlation between prototype pattern vectors be very small. In fact, feature dimension of each prototype pattern vector is 10 and pattern remained to be recognized 16, which cannot satisfy well the condition of the equation (3). Therefore, here we introduce the strategy of OAO, as a result, the recognition rate of SNN is improved remarkably.

Experiment 2. Another method for extracting image texture features, brushlet decomposition [8], is introduced to deal with the multi-class recognition problem in SNN. Here, to compare under the same condition, previous images are decomposed into 2-level and 3 levels by brushlet transform, which gets 8-dimension and 32-dimension feature vectors, respectively. Then the pattern discrimination is conducted using the same method as experiment 1, i.e. three methods in Table 2, where substitutes 8-D for 8-dimension, 32-D for 32-dimension and mean for the mean value of the original image feature vectors. The corresponding recognition results different features dimension are shown in Table 2.

Table 2. Comparison of recognition results using different dimension of feature

	SNN(image)		SNN(mean)		SNN(ICFCM)	
	8-D	32-D	8-D	32-D	8-D	32-D
test time (s)	0.015	0.016	0.016	0.016	0.047	0.063
ARR of training sets(%)	82.2	100	84.3	99.46	89.76	100
ARR of test sets (%)	61.41	85.75	67.75	86.09	78.89	89.77

From Table 2, we can see that recognition result is also improved greatly only by increasing feature dimension without resorting to the strategy of OAO. In other words, for multi-class recognition problem in SNN, increasing the feature dimension to describe prototype pattern can decrease the correlation between prototype pattern vectors, which improves the classifying ability of the SNN. However, it is difficulty to get high-dimension feature vectors to denote the image. This problem can be solved either by developing new method of feature extraction or by feature fusion.

6 Conclusions and Discussion

In this paper, we have presented an approach to optimize prototype vectors in SNN based on immune clonal clustering, by which texture image recognition is implemented. In addition, we argue that the existing SNN algorithm do not provide better multi-class recognition performance. Further, optimization of attention parameters and construction of order parameters deserve more deep research in SNN.

References

1. Haken, H.: Synergetic Computers and Recognition-A Top-down Approach to Neural Nets. Springer-Verlag Berlin (1991)
2. Du, H.F., Jiao, L.C.: Clonal Operator and Antibody Clone Algorithms. Proceedings of the First International Conference on Machine Learning and Cybernetics, Beijing (2002) 506–510
3. Yao, X., Liu, Y.: Evolving Neural Network Ensembles by Minimization of Mutual Information. International Journal of Hybrid Intelligent Systems, 1 (2004) 1221
4. Wang, H.L., Q, F.H.: Learning Algorithm of Synergetic Neural Network Based on Clustering Algorithm. Journal of Shanghai Jiaotong University, 32 (1998) 39–41
5. Liu, R.C., Du, H.F., Jiao, L.C.: Immunity Clonal Strategies. In Proceedings of Fifth International Conference on Computational Intelligence and Multimedia Applications, Xi'an (2003) 290–295
6. Bezdek, J.C., Hathaway, R.J.: Optimization of Fuzzy Clustering Criteria Using Genetic Algorithm. FUZZY-IEEE, (1994) 589–594
7. Weston, J., Watkins, C.: Multi-class Support Vector Machines. Technical Report CSD-TR–98–04, Royal olloway University of London (1998)
8. Francois, G.M., Ronald, R.C.: Brushlets: A Tool for Directional Image Analysis and Image Compression. APPLIED AND COMPUTATIONAL HARMONIC ANALYSIS, 4 (1997) 147–187

Content Based Retrieval and Classification of Cultural Relic Images*

Na Wei[1], M. Emre Celebi[2], and Guohua Geng[1]

[1] Dept. of Computer Science, Northwest University at Xi'an
Shaanxi 710069 ,China
weinaok@126.com, ghgeng@nwu.edu.cn
[2] Dept. of Computer Science and Engineering, University of Texas at Arlington
Arlington, TX 76019 USA
celebi@cse.uta.edu

Abstract. In this paper we present a novel system for content-based retrieval and classification of cultural relic images. First, the images are normalized to achieve rotation, translation and scaling invariant similarity retrieval. After normalization, a combination of color and shape features is extracted from the images. In order to improve the retrieval efficiency, a modified version of principal component analysis is used to reduce the dimensionality of the feature space. Retrieval performance of the system is evaluated for three different distance functions using the normalized recall measure. A multi-class support vector machine (SVM) classifier is used for classification. The results demonstrate that the system is both effective and efficient.

1 Introduction

Content-based image retrieval (CBIR) has been a topic of considerable interest in various applications domains such as trademark retrieval, fashion design, journalism, digital museum browsing, etc. [1],[2]. Numerous CBIR systems have been developed to address these problems both in the industry and academia [8],[11],[12].

We have developed a digital archaeology museum, which contains 2740 cultural relic images of high quality. Our project aims at advancing information technologies related to the preservation, retrieval, and dissemination of digital imagery of cultural heritages. This paper focuses on content-based retrieval and classification of cultural relic images in our digital archeology museum.

The rest of the paper is organized as follows: Section 2 presents the image preprocessing phase. Section 3 describes the feature computation, dimensionality reduction, and retrieval performance evaluation procedures. Section 4 describes the support vector machine classification. Finally, Section 5 concludes the paper and provides future research directions.

* Na Wei and Guo-Hua Geng are supported by the *Remote Resource Construction* project funded by the Ministry of Education of China.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 292–297, 2005.

2 Preprocessing

During data acquisition, the camera orientation or camera-to-object distance may change. Hence the image fed to the system may contain a pattern that is rotated, scaled, or translated compared to its original form. In such cases, either the system should use features that are invariant to such transformations or there should be a preprocessor to maintain the rotational, scaling, and translation invariancy [3]. This study uses the latter approach.

Figure 1 shows a sample of images from our database. First, the images are binarized using dynamic thresholding method described in [13]. Let $f(x,y)$ and $c=[C_x,C_y]^T$ denote the binary image function and the center of gravity of the object, respectively. To achieve translation invariance, we translate the object so that the center of gravity coincides with the origin:

$$f_T(x_i,y_j)=f(x_i-C_x,y-C_y) \tag{1}$$

The term radius for a pixel is defined to be the length of the straight line connecting the pixel and the origin. The scaling process will bring a radial boundary to the pattern in the image while adjusting the number of on-pixels. It thus alleviates any possible pattern deformation caused by scaling. The average radius is computed as:

$$\bar{r}=\frac{\sum\limits_{i=1}^{N}\sum\limits_{j=1}^{N} f_T(x_i,y_j)\cdot\sqrt{x_i^2+y_j^2}}{\sum_{i=1}^{N}\sum_{j=1}^{N} f_T(x_i,y_j)} \tag{2}$$

In order to achieve scale invariance we scale the object with a scaling factor $s=\bar{r}/R$, where R is equal to one-fourth of the grid size:

$$f_{TS}(x_i,y_j)=f_T(s\cdot x_i,s\cdot y_j) \tag{3}$$

To achieve rotational invariance we determine the major axis orientation. Thus, the rotation angle in terms of the central moments of the object is given by:

$$\theta=\tan^{-1}\left(\frac{\mu_{20}-\mu_{02}+\sqrt{(\mu_{20}-\mu_{02})^2+4\mu_{11}^2}}{2\mu_{11}}\right) \tag{4}$$

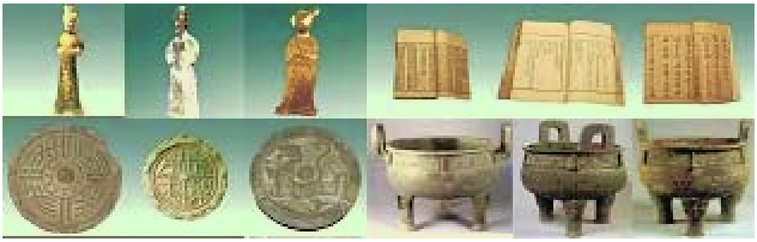


Fig. 1. Sample cultural relic images from four different classes

3 Content Based Retrieval

3.1 Feature Computation and Dimensionality Reduction

For each image in the database, we compute the following 19 features: Circularity [4], bending energy [4], 7 moment invariants $\{\Phi_1, \Phi_2, \ldots, \Phi_7\}$ [4], contour sequence moments $\{F_1, F_2, F_3, F_4\}$ [4], and 1st & 2nd order color moments [5] computed from individual channels of the HSV color space. Since these descriptors are widely used in the literature, we omit their formulae.

As a result of the feature computation step, we have feature space with 19 dimensions. Well known problems associated with high dimensionality include (a) high computational cost, (b) classifier accuracy degradation, and (c) difficulty in visualization [10]. In this study we use a modified version of the Principal Component Analysis (PCA) for dimensionality reduction. In the classical PCA method, the data is represented by N principal components (*PCs*), where $N < K$ = the number of features. However, we still need all of the K original features to compute the N *PCs* since each *PC* is a linear combination of the original features. This modified version of PCA differs from the classical one in that instead of using N *PCs*, we select N original features to represent the data. We use Jolliffe's B4 Method [6] for selecting N representative features from K original features. In this method, starting from the largest *PC*, we select the feature with the highest coefficient on the relevant *PC* to represent that component, unless it has been chosen to represent a larger *PC*. In this way, a total of N features are retained.

The eigenvalue and the percentage of explained variance for each *PC* are given in Table 1.

In our analysis we decided to keep 9 *PCs* that account for 96.9% of the variation in the original data. Using Jolliffe's B4 method we decide to keep the following original features: Circularity, F_4, Φ_1, Φ_5, Φ_7, the 1st order color moments for *H* and *V*, and the 2nd order color moments for *H* and *S*. The remaining 10 features are discarded.

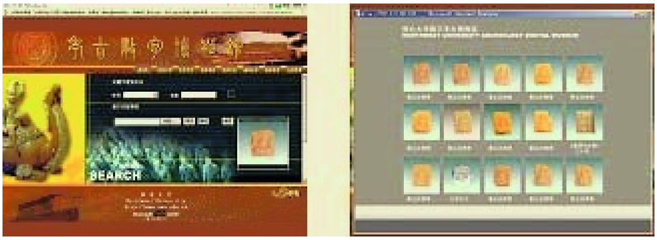
Table 1. Eigenvalue and percentage of explained variance for each $PC/Eigenvector$

$PC/Eigenvector$	*Eigenvalue*	*Variance%*	*Cumulative Var.%*
1	7.5273	39.617	39.617
2	3.7591	19.785	59.402
3	2.0371	10.722	70.124
4	1.6328	8.594	78.718
5	1.3688	7.204	85.922
6	0.8469	4.457	90.379
7	0.5656	2.977	93.356
8	0.3837	2.020	95.376
9	0.2911	1.532	96.908

3.2 Retrieval Performance Evaluation

An important step between feature computation and similarity measurement is feature normalization. In this study, we use z-score normalization.

Retrieval performance of the system is evaluated using the normalized recall (*NR*) measure [8]. *NR* is formulated as $AVRR/IAVRR$, where *AVRR* is the average rank of all relevant, displayed images and *IAVRR* is the ideal average rank which is the maximum when all relevant images are retrieved on the top. *NR* is equal to 1 for perfect retrieval and is greater than 1 for all other cases. For the experiments we selected 98 query images from 5 different classes. The relevant images in the database for each query image are determined manually. A list of 32 best matches is displayed for each query. Figure 2 (a) and (b) show snapshots of the query interface before and after query execution, respectively.



(a) (b)

Fig. 2. Snapshots of the query interface (a) before query execution (b) after query execution

The choice of distance function can drastically influence the retrieval performance [7]. For this reason, we compare the performance of three different distance functions: Euclidean, chi-square, and *K-L* divergence. Figure 3 shows the *NR* values for each combination of distance function and query image class. It can be seen that Euclidean distance function outperforms both chi-square and *K-L* divergence distance functions.

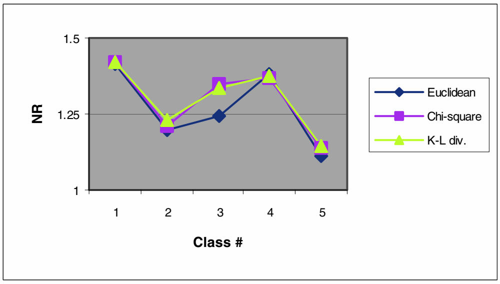


Fig. 3. Retrieval performance comparison using the *NR* measure

4 Support Vector Machine Classification

Support vector machines (SVMs) are non-parametric supervised classifiers with excellent generalization capabilities. They have been successfully used in numerous applications such as hand-written digit recognition, object recognition, speaker identification, face detection, etc. [9].

SVMs are originally designed for binary classification. There are two main approaches for extending the basic SVM to handle multi-class classification: (1) Modifying the design of the SVM in order to incorporate the multi-class learning directly in the quadratic solving algorithm (2) Combining several binary classifiers: "One against one" applies pairwise comparisons between classes, while "One against the others" compares a given class with all the others put together. In this work we experimented with the second approach. In our experiments, the "one against the others" approach often produced quite unbalanced two-class problems. Therefore, we chose the "one against one" method to perform multi-class classification.

In a $k-class$ classification problem, the "one against one" method requires $k(k-1)/2$ binary classifiers, one for each pair of classes. In the training phase, each binary classifier is trained to classify two classes. In the testing phase, the output from each binary classifier in the form of a class label is obtained. The class label that occurs the most is assigned to the data.

The general purpose Gaussian RBF is chosen as the kernel:

$$K(x,y) = \exp(-\gamma(x-y)^2) \tag{5}$$

For $k=2$ and $k=5$, i.e. $2-class$ and $5-class$ problems, we set the kernel parameter γ as *0.5* and *0.2*, respectively and estimate the classification error for each cost parameter (C) value in the set $\{2^{11}, 2^{10}, \cdots, 2^{-2}\}$. For each *(C, γ)* combination, the validation performance is calculated using 70% of the data as the training set and 30% as the test set. Then, we calculate the average classification error rate using the leave-one-out method with the *(C, γ)* combination that achieves the highest validation rate. Table 2 shows the optimal *(C, γ)* value and the corresponding average classification error rate for the $2-class$ and $5-class$ cases.

Table 2. Average classification error rates

Problem	(C,γ)	*Error rate*
$2-class$	$(2^4, 0.5)$	2 %
$5-class$	$(2^7, 0.2)$	4%

5 Conclusions

In this paper we presented a novel system for content-based retrieval and classification of cultural relic images. First, the images are normalized to achieve rotation, translation, and scaling invariant similarity retrieval. Following image normalization, a combination of color and shape descriptors is extracted from the images. A modified version of principal component analysis is employed to reduce the dimensionality of the feature space. To test the retrieval performance, three distance functions are evaluated. Finally, a multi-class support vector machine is used for classification.

Future work will be directed towards incorporation of texture features and relevance feedback to the system in order to improve the retrieval performance.

References

1. Zhou, M.Q, Wei, N., Geng, G.H.: Content Based Image and Model Retrieval Techniques. The Third Asian Workshop on Foundations of Software, Xi'an, China (2004)
2. Wei, N., Geng, G.H., Zhou, M.Q.: 3D Model Retrieval Using Distance Histogram. Proc. of the 14th National Conference on Artificial Neural Networks and Natural Computing., He-Fei, China (2004) 537-539
3. Yuceer, C., Oflazer, K.: A Rotation, Scaling, And Translation Invariant Pattern Classification System. Pattern Recognition. **26** (1993) 687-710
4. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis, and Machine Vision. Brooks/Cole Pub. CA (1998)
5. Stricker, M., Dimai, A.: Color Indexing with Weak Spatial Constraints. Proc. of the SPIE. **2670** (1996) 29-40
6. Dunteman, G.H.: Principal Component Analysis. Sage Publications (1989)
7. Zhang D.S., Lu, G.: Evaluation of Similarity Measurement For Image Retrieval. Proc. of IEEE ICNNSP'**03** (2003) 928-931
8. Faloutsos, C., Equitz, W., Flickner, M. et al.: Efficient and Effective Querying by Image Content. Journal of Intelligent Information Systems, **3** (1994) 231-262
9. Chapelle, O., Haffner, P., Vapnik, V.: Support Vector Machines for Histogram-Based Image Classification. IEEE Trans. on Neural Networks, **10** (1999) 1055-1064
10. Celebi, M.E., Aslandogan, Y.A.: Content-Based Image Retrieval Incorporating Models of Human Perception. Proc. of IEEE ITCC Conf. (2004), Las Vegas, NV. 241-245
11. Pentland, A., Picard, R.W., Sclaroff, S.: Photobook: Content-based manipulation of image databases. International Journal of Computer Vision, **18** (1996) 233-254
12. Smith, J.R., Chang, S.-F.: VisualSEEK: A Fully Automated Content Based Image Query System. Proc. of the ACM Multimedia Conf., Boston, MA. (1996) 87-98
13. Bernsen, J.: Dynamic Thresholding of Grey Level Images. Proc. of the 8th Int. Conf. on Pattern Recognition., Paris, France (1986)1251-1255

Obscene Image Recognition Based on Model Matching and BWFNN

Xiaohua Liu, Zhezhou Yu, Libiao Zhang, Miao Liu, Chunguang Zhou,
Chunxia Li, Catitang Sun, and Li Zhang

College of Computer Science and Technology
Key Laboratory of Symbol Computation and Knowledge Engineering of Ministry of Education
Jilin University, Changchun 130012, China
{cgzhou,xiaohua}@jlu.edu.cn

Abstract. In this paper the obscene images first primarily recognizes through the human skin color detection and key point model matching. The other images that are not confirmed extract characteristic of obscene images through edge detection, posture estimation and wavelet compression, and then recognized using the optimizing broaden weighted fuzzy neural network, which is called two-phase recognizing method. The experiment indicates the method that this paper present can recognize the obscene images effectively.

1 Introduction

There are many teenagers commit crime related to the poison of the obscene picture and obscene website. In this paper, the pictures that appear in Internet, the hard disk of the computer and CD are processed based on the image's content to judge whether they are healthy. It will provide a healthy, green machine environment after obscene image detecting. Many scholars are devoted to the research of the rejecting obscene software. Image recognition software "photo no no" appeared in U.S.A. "Photo no no" is developed with java. Through testing, the efficiency is not very high. Protecting the teenager's edition at home is developed with VC, with the large databases. The algorithm judges the image based on the analysis of the image content. Multi resolution wavelet analysis and multi layer neural network are used in it. In this paper two-phase recognition method is proposed, namely the key point model matching and BWFNN. After K-L transforming, Sobel transforming, model matching and EZW encoding, we extract the feature vector, which will be used to recognize.

The second part of the paper is obscene image modeling consist of skin color detection, posture and key position characteristic models and wavelet compression; the broadened weighted fuzzy neural network is introduced in the third part; the forth part shows and analyzes the experiment results; and the fifth part includes the conclusion and anticipant.

2 Obscene Image Modeling

We extract the feature data of two-phase recognition by preprocessing the obscene image. The detailed process is explained as follow.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 298–303, 2005.

2.1 Skin Color Detection

As the same as [1][2], we use KL transform on the image. The transform matrix is:

$$\begin{bmatrix} K_1 \\ K_2 \\ K_3 \end{bmatrix} = \begin{bmatrix} 0.661 & 0.547 & 0.507 \\ -0.71 & 0.254 & 0.657 \\ 0.232 & -0.80 & 0.560 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

By statistic, confirm the skin color value domain of K1, K2, and K3 as following:

$$108.6<K1<379.4;\ -60.8<K2<31.8;\ -17.9<K3<19.1$$

Regarding this value range as KL threshold that skin color filters, and transform R, G and B into the space of pure color, namely

$$I=R+G+B;\ Rc=R/I;\ Gc=G/I;\ Bc=B/I;\ Rc+Gc+Bc=1$$

Rc and Gc can be used as the additive judging value,

$$0.31\leq Rc\leq 0.465;\ 0.28\leq Gc\leq 0.363$$

The color of the input picture is transformed with above-mentioned matrix. The result is set to 255 if within the range of the value, otherwise set to 0. The skin area is confirmed in two-value picture.

2.2 Posture and Key Position Characteristic Models

1) Posture Models.

In this paper 100 obscene pictures with different posture are used as the initial samples. The obscene part of the picture is located manually. Using Sobel edge detector, we get edge picture. By clockwise rotation, for every 45 degrees each sample is turned into a new one. In this way, 200 primitive samples produce 800 ones after transforming, 800 healthy pictures are chosen at the same time as negative samples.

2) Matching of Key Point.

In this paper key point body model is presented besides posture model. In the obscene picture, the key position of chest and waist are located manually. The key point rectangles are located according to the position in each picture respectively. Then the scale of each rectangle is normalized, and the edges are from Sobel edge detector. By some rules synthesizing all the rectangles, the key point model is obtained. There are four models that are grouped into two classes according to the difference of male and female.

2.3 Embedded Zero Wavelet Coding

Embedded zero wavelet (EZW) [3] encoded method is based on the following 3 main parts, namely:

(1) Use the inherent similarity of wavelet transform to predict the important information position.

(2) Gradually quantize the wavelet coefficients.

(3) Adaptive arithmetic encoding is used to implement data lossless compression.

EZW encoded method is showed in fig.1

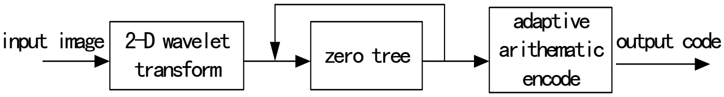


Fig. 1. The flow of EZW encoding

EZW encoding is used to encode the image after the input image is transformed by bior-orthorhombic wavelet for 4 times. Edge detecting the above-mentioned posture image, 256-grey level picture is obtained, and then the size scale is normalized to 120*160. 50-dimension feature vector is formed after EZW encoded on the data.

3 Broadened Weighted Fuzzy Neural Network

The broadened weighted fuzzy neural network is used to obscene image recognition. Its result is good by the experiment. It will be introduced through two parts.

3.1 Broaden Weighted Fuzzy Reasoning Method

We definite the fuzzy set number of the input vector x_i as k_i, and the number of output vector Y is l. Let m=$\prod_{i=1}^{n} k_i$, then the maximum number of the fuzzy rules is m*l. We attempt to enumerate out various kinds of regular fuzzy possible forms, and express various kinds of possibilities of the reasoning result to the weights of different results with every rule. The fuzzy rule is collected to show:

$$\underset{j=1}{\overset{m}{OR}}\ (\mathrm{IF}\ (\underset{i=1}{\overset{n}{AND}}\ (x_i\ \mathrm{IS}\ A_{ij})\ )\ \mathrm{THEN}\ Y\ \mathrm{IS}\ w_{j1}\ /\ z_1\ ,\ w_{j2}\ /\ z_2\ ,\cdots\ w_{j1}/\ z_1\ )$$

The m is counted for the fuzzy rule, n is the number of input vector, l is the number of fuzzy subset of the output vector, z_i is a constant. W_{ji} expresses the weight. If the fitness value is defined to μ_j ,the final conclusion can be obtained by the below(See formula 1) improved method of weighted averages.

$$z_0 = \frac{\sum_{i=1}^{l} (f(\sum_{j=1}^{m} u_j w_{ji}) * z_i)}{\sum_{i=1}^{l} (f(\sum_{j=1}^{m} u_j w_{ji}))} \qquad (1)$$

3.2 The Broaden Weighted Fuzzy Network (BWFNN)

The structure of broaden weighted fuzzy neural network [4] shows as Fig 2

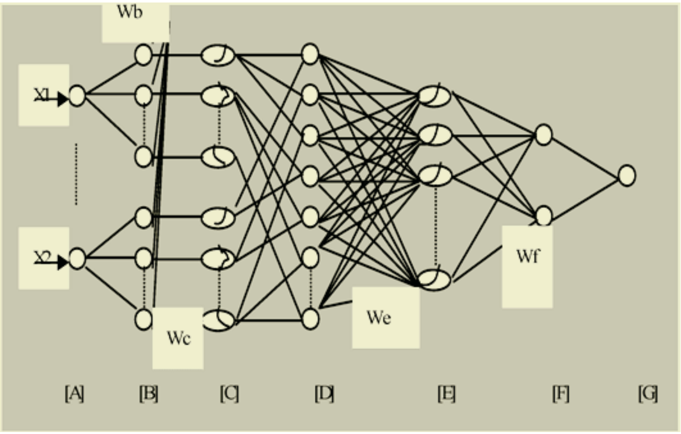


Fig. 2. Structure of the weighted fuzzy neural network

(A): $O_i=I_i=x_i$

(B): $O_i=I_i- w_{bi}=x_i- w_{bi}$

(C): $O_i=Sigmoid(w_{ci}*I_i)$ or$=Exp(-(I_i/ w_{ci})^2)$ or$=1 - Sigmoid(w_{ci}*I_i)$

(D): $O_i=\prod I_i$

(E): $O_i =Sigmoid(\Sigma(I_i*w_{ei}))$

(F): $O_1=\Sigma w_{fi}*I_i$; $O_2 =\Sigma I_i$

(G): $O=I_1 / I_2$

Among them w_b is the variable threshold value, w_c, w_e and w_f are the variable weight value. And A layer is input layer, B and C layers are made to be fuzzy to the input vector. D, E layers realize the fuzzy reasoning rules, and D layer realizes ‘and operation’ of fuzzy rules’ prediction, E layer used to finish the sum function and get output, F ,G layers realize operating fuzzily with “center law”.

For above-mentioned networks, the essential of the rule obtaining and the member function improving is to look for the fit w_b, w_c, w_e and w_f. It uses the HGA-FNN[5] to optimize the structure and parameter of the fuzzy system. The balance of accuracy and complexity can be realized through adjusting to α, β value of fitness function.

4 Obscene Image Recognition

We can get the rectangle that needs to judge by the proportion of the skin pixels in the area after detecting skin color. They can be recognized by the key point model matching and the fuzzy neural network.

4.1 Key Point Recognition

It is mainly by key point model matching [6] to recognize the obscene part. As the second part showing, there are four models made up of the man and woman's chest and the waist part. It is obscene when the similarity is higher than the threshold after modeling matching, then the flow is finished. Otherwise it is recognized by fuzzy neural network. The threshold value is set by the results of many times calculating.

4.2 Recognition by the Broaden Weighted Fuzzy Neural Network

According to the feature extracting method offered by the second part, 50-dimension vector obtained after EZW encoding working on the edge image is sent to the network, see fig.3. All of network parameters are fit for obscene image recognition after optimizing the network structure and parameters by HGA-FNN.

4.3 Experimental Result Analysis

The result of the network recognition is correct for the 200 obscene images of the above. As for other images, the correct rate is 50% after the model matching and 90% after the network recognizing when the image is clarified and the light is balanced. But the accurate rate will fast drop when the illumination and shooting angle and color are different. See the table 1.

Table 1. The results with different conditions

Picture quality	Correct rate
Clarity & Balance	85%
Common & Balance	74%
Common & Unbalance	68%
Dimness & Balance	53%
Dimness & Unbalance	28%

To a healthy picture, if its color is same to the skins', the wrong rate(that the healthy picture is recognized an obscene one) is 13% and if the color is far different from the skins', the wrong rate is 4%.

5 Conclusions and Anticipation

The algorithm offered in the paper can make the effective judgment to a lot of obscene pictures. The correct recognition rate still needs to improve further because of the influence of many factors such as illumination, race, skin color and shooting angle; moreover, the posture models can not contain all the obscene images. In the future, all kinds of obscene images can be collected to increase the network samples, and the impact of skin color and illumination can be reduced and a lot of performance parameters can be self-adapted or manually regulated.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. 60 175024 and 60433020, and the Key Science-Technology Project of the National Education Ministry of China under Grant No. 02090.

References

1. Liu, X.H., Cui, Z., Duan, J., Zhou, C.G.: Face Detection in Complex Background. Minimicro System, (2003) 1105-1109
2. Zhang, H.M., Zhao, D.B., Gao, W.: Face Detection Under Rotation in Image Plane Using Skin Color Model. Neural Network and Feature-based Face Model, Chinese Journal Computers. (2002) 1250-1256
3. Shapiro, J.M.: Embedded Image Coding Using Zerotree of Wavelet Coefficients. IEEE Trans. Signal Processing, 41 (1993) 3445-3462
4. Zhou, C.G., Liang, Y.C.: Computation Intelligent. Jilin University Publication (2001)
5. Ma, M., Zhou, C.G., Zhang, L.B., Dou, Q.S.: Optimization of Fuzzy System Based on Hierarchical Genetic Algorithm. Journal of Jilin University (Science Edition), (2004) 559-564
6. Liang, L.H., Ai, H.Z., Xiao, X.P., etc.: Face Detection Based on Template Matching and Support Vector Machines, Chinese Journal of Computers, (2002) 22-29.

Classification of SAR Imagery Using Multiscale Self-organizing Network

Xianbin Wen

School of Math. and Information Science, Shandong University of Technology
255049, Zibo, China
xb_wen@yahoo.com.cn

Abstract. A multiscale self-organizing mixture network (MSOMN) is proposed for learning mixture multiscale autoregressive model of synthetic aperture radar (SAR) imagery. The MSOMN combines the multiscale method, the Kullback-Leibler information metric, the stochastic approximation method, and the self-organizing map structure. Updating of the parameters is limited to a small neighborhood around the winner that is based on maximum posterior probability. The network possesses a simple structure, and yields fast convergence, which is confirmed by experimental results of SAR images.

1 Introduction

The presence of speckle on SAR images not only reduces the interpreter's ability to resolve fine detail, but also makes automatic segmentation and pattern recognition of such images difficult. To fully exploit the coherent nature and complexity of SAR image formation, we employ a recently introduced class of mixture multiscale autoregressive (MMAR) model evolving on dyadic trees [1]. The model provides a powerful semi-parametric framework for describing complex random fields that evolve in scale. Meantime, the expectation-maximization (EM) algorithm to the MMAR model is applied [1]. However, the EM algorithm and standard ML based algorithm maximize the sample joint-likelihood. This process can over-emphasize or favor these observations, so may lead to over-fitting problems, especially when the sample size is small. In addition, due to its use of deterministic gradient descent and batch operation nature, the EM algorithm has a high possibility of being trapped in local optima and is also slow to the converge [2]. There have been several networks or neural networks approaches to solving Gaussian mixture problems, such as a self-organizing mixture network (SOMN) [3] and probabilistic decision-based neural network [4]. These networks all have shown additional advantages over the EM algorithm.

In this paper, we further extend and generalize the SOMN to the MMAR model of SAR imagery. The resulting network, namely multiscale self-organizing mixture network (MSOMN), combines the multiscale method [5], the Kullback-Leibler information metric [6], the stochastic approximation method [7], and the self-organizing map structure [8]. The MSOMN not only has a advantages of SOMN but also utilizes information of multiscale imagery.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 304–309, 2005.

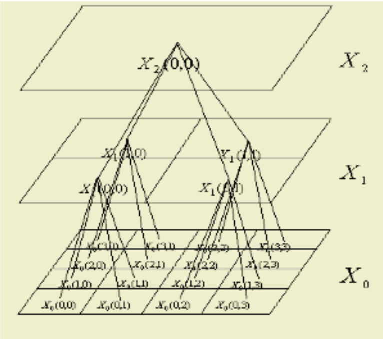


Fig. 1. Sequence of three multiresolution SAR images mapped onto a quadtree

2 MMAR Model of SAR Image and EM Algorithm

The starting point for our model development is a multiscale sequence X_L, $X_{L-1}, \dots, X_0$ of SAR images, where X_L and X_0 correspond to the coarsest and finest resolution images, respectively. The resolution varies dyadically between images at successive scales. More precisely, we assume that the finest scale image X_0 has a resolution of $\delta \times \delta$ and consists of an $N \times N$ array of pixels (with $N = 2^M$ for some M). Hence, each coarser resolution image X_m has $2^{-m}N \times 2^{-m}N$ pixels and resolution $2^m \delta \times 2^m \delta$. Each pixel $X_m(k,l)$ is obtained by taking the coherent sum of complex fine-scale imagery over $2^m \times 2^m$ blocks, performing log-detection (computing 20 times the log-magnitude), and correcting for zero frequency gain variations by subtracting the mean value. According, each pixel in image X_m corresponds to four "child" pixels in image, X_{m-1}. This indicates that quadtree is natural for the mapping. Each node s on the tree is associated with one of the pixels $X_m(k,l)$ corresponding to pixel (k,l) of SAR image X_m. Fig. 1 illustrates a multiscale sequence of three SAR images, together with the quadtree mapping. We use the notation $X(s)$ to indicate the pixel mapped to node s. The scale of node s is denoted by $m(s)$.

For complex SAR image, we define the MMAR model of SAR imagery as:

$$f(X(s) \mid \Theta, \Im_s) = \sum_{k=1}^{K} \frac{\pi_k}{\sigma_k} \varphi\left(\frac{X(s) - a_{k,0} - a_{k,1} X(s\bar{\gamma}) - \cdots - a_{k,p_k}(s\bar{\gamma}^{p_k})}{\sigma_k} \right).$$

$$\triangleq \sum_{k=1}^{K} \frac{\pi_k}{\sigma_k} \varphi\left(X(s) \mid \theta_k, \Im_s \right). \tag{1}$$

where $\Theta = (\pi_1, \dots, \pi_K, \theta)$, $\theta = (\theta_1, \dots, \theta_K)$, $\theta_k = (a_{k,0}, \dots, a_{k,p_k}, \sigma_k)$, $\sum_{k=1}^{K} \pi_k = 1$, $\pi_k > 0$, f is the probability density function. K is the number of classes, $\Im_s$ is the set of $X(s\overline{\gamma}),\ldots,\ X(s\overline{\gamma}^{p})$ ($p=\max_{k} p_k$), and $\varphi(.)$ be the probability density function of a standard normal distribution.

For most unsupervised learning applications, only forms of the component-conditional densities are known. Given SAR image X , the MMAR model parameters can be estimated. And, likelihood function is given as follows:

$$f(X,\theta)=\prod_{\{s|m(s)=m\}}\{f(X(s)\,|\,\Theta,\Im_s\}. \tag{2}$$

The iterative EM algorithm for estimating parameters by (2) consists of an E-step and an M-step, which have been described in [1]. But, its slow convergence and high computational costs need to be addressed for practical application.

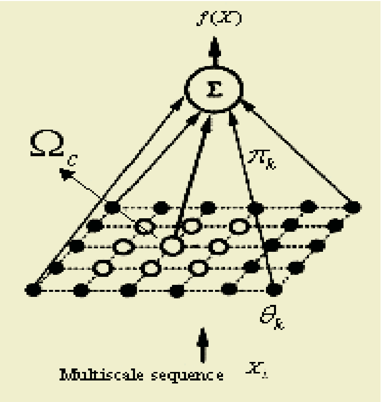


Fig. 2. The structure of the multiscale self-organizing mixture network

3 The Multiscale Self-organizing Mixture Network

3.1 The MSOMN Structure

Based on the MMAR model (1), the MSOMN structure can be illustrated as in Fig. 2. For a mixture of finite components, the network Ξ place M nodes in the input SAR imagery sequence, X_L . The kernel parameters, e.g., θ_k , are the learning weights. The output of a kernel is the conditional density of that component in the mixture. The network output sums the responses of these kernel weighted by the prior probabilities or mixing parameters, π_k , which are also the learning weights. At each step, j , a sample point, denoted by $X(s)$, is randomly taken from the finest scale imagery X_0 . A winner is chosen according to its kernel output multiplied by its mixing parameter, i.e., estimated posterior probability

$$P(k\,|\,X(s))=\frac{\hat{\pi}_k\varphi(X(s)\,|\,\theta_k,\Im_s)}{\hat{\sigma}_k f(X(s)\,|\,\hat{\Theta},\Im_s)}\,. \tag{3}$$

Within a neighborhood of the winner, Ω_c , the weights are updated. Thus MSOMN is similar to the SOMN in the sense of local learning properties.

3.2 The MSOMN Updating Algorithm

Suppose that the true SAR imagery data density function and estimated one are $f(X)$ and $\hat{f}(X)$, respectively. The Kullbac-Leibler information metric [6] measures the divergence between these two, and is defined as

$$I = -\int \log \frac{\hat{f}(X)}{f(X)} f(X)dX \ . \tag{4}$$

When the estimated density is modeled as a MMAR model, i.e., a function of various sub-densities and their parameters, one can seek the optimal estimate of these parameters by minimizing I via its partial differential in respect to every model parameter, i.e., for $k = 1,\dots K$, $i = 0,\dots, p_k$,

$$\frac{\partial I}{\partial a_{ik}} = -\int \left[\frac{1}{\hat{f}(X \mid \hat{\Theta}, \Im_s)} \frac{\partial \hat{f}(X \mid \hat{\Theta}, \Im_s)}{\partial a_{ik}} \right] f(X)dX \equiv 0 \, . \tag{5}$$

$$\frac{\partial I}{\partial \sigma_k} = -\int \left[\frac{1}{\hat{f}(X \mid \hat{\Theta}, \Im_s)} \frac{\partial \hat{f}(X \mid \hat{\Theta}, \Im_s)}{\partial \sigma_k} \right] f(X)dX \equiv 0 \, . \tag{6}$$

$$\frac{\partial I}{\partial \pi_k} = -\int \left[\frac{1}{\hat{f}(X \mid \hat{\Theta}, \Im_s)} \frac{\partial \hat{f}(X \mid \hat{\Theta}, \Im_s)}{\partial \pi_k} \right] f(X)dX + \lambda \frac{\partial}{\partial \pi_k} \left[\sum_{k=1}^{K} \pi_k - 1 \right] = 0 \, . \tag{7}$$

As the true SAR imagery data density is not known, the Robbins-Monro stochastic approximation method [7] can be used for solving these nondirectly solvable equations. This results in the following adaptive updating algorithm with the constraint parameter λ set to one:

$$\hat{a}_{ik}^{(j+1)} = \hat{a}_{ik}^{(j)} + \alpha(j)\hat{P}(k \mid X) \left[X(s) - \sum_{l=0}^{p} \hat{a}_{lk}^{(j)} \mu(X(s),l) \mu(X(s),i) \right]. \tag{8}$$

$$\sigma_k^{(j+1)} = \sigma_k^{(j)} + \alpha(j)\hat{P}(k \mid X) \left\{ \left[X(s) - \sum_{l=0}^{p} \hat{a}_{lk}^{(j)} \mu(X(s),l) \right] - \hat{\sigma}_k^{2(j)} \right\}. \tag{9}$$

$$\hat{\pi}_k^{(j+1)} = \hat{\pi}_k^{(j)} - \alpha(j) \left[\hat{P}(k \mid X) - \hat{\pi}_k^{(j)} \right]. \tag{10}$$

where $\alpha(j)$ is the learning rate at time step j , and $0 < \alpha(j) < 1$, and decreases monotonically. $\mu(X(s),i) = 1$ for $i = 0$ and $\mu(X(s),i) = X(s\bar{\gamma}^i)$ for $i > 0$.

The updating of the above parameters can be limited to a small neighborhood of the winning node, due to the diminishing spreading properties of the most conditional densities. That is, the density can be approximated by a mixture of a small number of nodes at one time, i.e.,

$$\hat{f}(X\,|\,\Theta,\Im_s) \approx \sum_{k\in\Omega_c} \frac{\hat{\pi}_k}{\hat{\sigma}_k}\varphi(X(s)\,|\,\hat{\theta}_k,\Im_s) \ . \tag{11}$$

where c is the winning node index and Ω_c is a neighborhood of the winner. Shrinking the neighborhood during the process provides a subjective and ad hoc adjustment to the variance of the Gaussians in order to aid the fitting and to allow them to settle to the data distribution as the learning progresses.

4 Experimental Results for SAR Imagery

To demonstrate the proposed algorithm for MMAR model, we apply it to two complex SAR images in Fig. 3(a), which consists of two classes of homogeneous regions. From the complex images, we generate an above-mentioned quadtree representation and use a two orders regression. Because it is found that by increasing the regression order to $p=2$ for both images, we can achieve a lower probability of misclassification and a good trade-off between modeling accuracy and computational efficiency. The initial value of parameters a_{ik} were set to small random values around zero, the initial weights π_k and the initial variance σ_k were set equally to $1/2$ and large value, respectively. Learning rates in the MSOMN were linearly decreasing from 0.5 and 0.05 for the parameters a_{ik} and for σ_k and a_{ik} , respectively. When the total of iterations is reached, then we can conclude with the segmentation step which consists on the application of the Bayesian decision rule [1].

Fig. 3 shows results from applying MSOMN to two SAR images, as well as results from EM algorithm for comparison. From Fig. 3, the MSOMN algorithm and EM algorithm perform similarly. But, the MSOMN algorithm converges much faster than the EM algorithm (the time of MSOMN and EM is 683(s) and 4682(s) under P4 computer, respectively.), and considerably reduces the segmentation time.

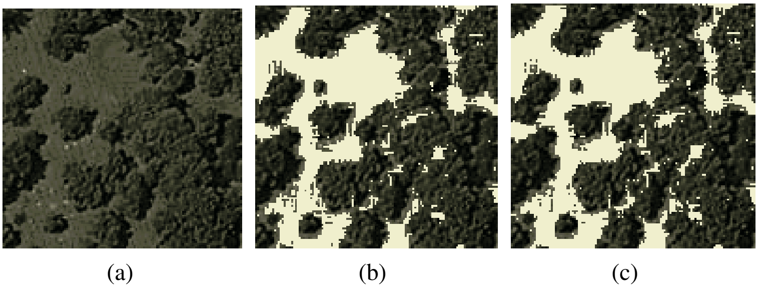


Fig. 3. (a) Original SAR image composed of woodland and cornfield. (b) Segmented image obtained using EM algorithm. (c) Segmented image obtained using MSOMN algorithm

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China (No. 60375003), the Science Foundation of Shandong University of Technology (No. 2004KJM32).

References

1. Wen, X.B., and Tian, Z.: Mixture Multiscale Autoregressive Modeling of SAR Imagery for Segmentation. Electronics Letters. 39 (2003) 1272-1274
2. Redener, R.A. and Walker, H.F.: Mixture Densities, Maximum Likelihood and the EM Algorithm. SIAM Rev.. 26 (1984) 195-239
3. Yin, H. and Allinson, N.M.: Self-Organizing Mixture Networks for Probability Density Estimation. IEEE Trans. Neural Networks. 12 (2001) 405-411
4. Yiu, K.K., Mak, M.W., and Li, C.K.: Gaussian Mixture Models and Probabilistic Decision-Based Neural Networks for Pattern Classification: A Comparative Study. Neural Comput. & Applic. 8 (1999) 235-245
5. Irving, W.W., Novak, L.M. and Willsky, A.S.: A Multiresolution Approach to Discrimination in SAR Imagery, IEEE Tran. Aerosp. Electron. Syst., 33 (1997) 1157-1169
6. Kullback, S. and Leibler, R.A.: On Information and Sufficiency. Ann. Math. Statist. 22 (1951) 79-86
7. Robbins, H. and Monro, S.: A Stochastic Approximation Method. Ann. Math. Statist., 22 (1951) 400-407
8. Kohonen, T.: Self-organized Formation of Topologically Correct Feature Maps. Biol. Cybern, 43 (1982) 56-69

Mixture of Experts for Stellar Data Classification

Yugang Jiang and Ping Guo

Department of Computer Science
Beijing Normal University, Beijing 100875, China
yugangjiang@sohu.com, pguo@ieee.org

Abstract. In this paper, mixture of experts model is first applied to stellar data classification. In order to obtain input patterns of mixture of experts model, we present a feature extraction method for stellar data based on wavelet packet transformation. Then a mixture of experts model is built for classifying the feature vectors. A comparative study of different classification methods such as a single radial basis function neural network is given. Real world data experimental results show that the mixture of experts has a good generalization ability and the obtained correct classification rate is higher than that of using a single neural network.

1 Introduction

Evolutionary studies are an essential part of astrophysics, because they allow us to discover and follow temporal changes of the physical and chemical conditions of the stars [1]. The study of distribution of spectral types and analysis of spectral data can help us to learn about the temporal change from a statistical point of view, which is an important part of evolutionary studies.

The traditional manual classification by experts is rather subjective and time-consuming, especially when the number of spectra is very high and a large number of human resources is required. It would therefore be advisable to optimize the procedure by means of an automatic and efficient computational technique.

Artificial neural network (ANN) developed from the 1940s proves to be a very effective method for automatic classification. The supervised learning networks, such as the back propagation (BP) neural network was adopted to classify astronomical objects by Odewahn *et al* in 1992 [2]. However, if back propagation is used to train a single, multilayer network to perform subtasks on different occasions, there will generally be strong interference effects that lead to slow learning and poor generalization. These drawbacks can be overcome by using a Mixture of Experts (ME) model. ME model is composed of several different expert networks plus a gating network that decides which of the experts should be used for each training case [3]. The principle of this system is to divide a large complex set of data into smaller subsets via the gating network that allows each expert to better represent a specific subset. Because it is easier to learn multiple

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 310–315, 2005.

small subsets than to learn a very large complex set, the ME model learns more quickly and is more appropriate to the stellar spectra.

Astronomical observations are often photon starved. Consequently many astronomical spectra have a poor signal-to-noise ratio (SNR) and are significantly corrupted by photon noise. Moreover, the spectral data are extremely voluminous. If we use the spectral lines directly as input of a neural network without pre-processing. The classification process is very time consuming and the classification results are very poor. In order to obtain a high correct classification rate, we have to reduce the data dimensions and suppress the noise. Because wavelet packet technique is able to de-noise both the high frequency and low frequency simultaneously, we adopt it to implement the de-noising process. Then a feature extraction method is proposed based on distribution of the noise. After calculating the feature vectors, a ME model is built to complete the classification.

The organization of this paper is as follows. In Section 2 we briefly introduce the ME model and wavelet packet technique. Section 3 illustrates the experimental strategy and gives the results and comparison. The last section offers concluding remarks.

2 Background

ME was developed by Jacobs *et al* [3] in 1991, which is one of the ways of combining multiple learners. It seems like voting, but the mechanism, which gives votes to each expert, is not stable over all the patterns in mixture of experts and it considers the input pattern while giving decisions [4]. The ME model consists of two types of neural networks, the experts and the gating network. The experts work in parallel using the same input pattern and the gating network determines which expert is best suited for the specific region of input space and assigns that portion to the expert.

Different experts are allowed to specialize on local regions. Each local expert learns training patterns from different regions and these are combined by the gating network. The architecture of ME is shown in Fig. 1. The output of the ME model looks like:

$$\mathbf{y} = \sum_{k=1}^{K} g_k \mathbf{y}_k, \tag{1}$$

where $\mathbf{y}_k$ represents output of the kth local expert, g_k is the proportion that the kth expert contributed to the output, which is determined by the gating network. For each input vector $\mathbf{x}$, there exists a probability vector $(g_1, g_2, \ldots, g_K)^T$. A softmax function is used to generate the probability vector:

$$g_k = \frac{\exp(\mathbf{w}_k \mathbf{x})}{\sum_{j=1}^{K} \exp(\mathbf{w}_j \mathbf{x})}, \tag{2}$$

where $\mathbf{w}_j$ is weight vector in the gating network.

Wavelet transform (WT) is a very powerful and versatile tool for astronomical signal processing. One of its drawbacks is that the frequency resolution is

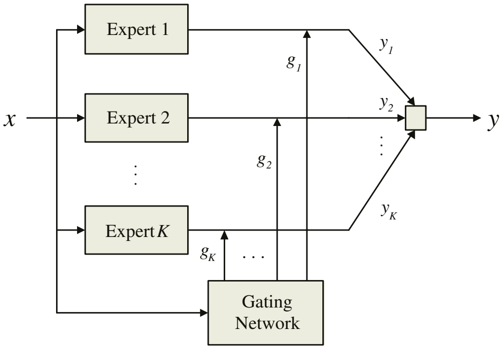


Fig. 1. Architecture of the mixture of experts

poor in the high frequency region. In certain applications, the wavelet transform may not generate a spectral resolution fine enough to meet the problem requirement. Wavelet packet transform is invited to solve this problem. The wavelet packet transform procedure can be considered as an extension of WT, in which both high and low frequency region are analyzed in the same way as the low frequency region in WT. This feature opens new and fruitful ways of processing and analyzing data of various kinds.

3 Experiments

In general terms, a stellar spectrum consists of a black body continuum light distribution, distorted by the interstellar absorption and emission of light and by the presence or absence of absorption and emission lines and molecular bands. In MK classification system, stellar spectra are catalogued into seven main spectral types in the order of decreasing temperature, namely: O – He II absorption; B – He I absorption; A – H absorption; F – Ca II absorption; G – strong metallic lines; K – bands developing; and M – very red lines [5]. The stellar spectra used in our experiments are selected from Astronomical Data Center (ADC). We use 161 stellar spectra contributed by Jacoby *et al* [6]. The spectral samples cover all the seven classes. The wavelength is from 351.0 to 742.7 nm and resolution is 0.14 nm. Before analyzing the spectra, all of them are scaled to flux 1 at wavelength 545.0 nm in order to normalize the flux values.

3.1 Pre-processing Based on Wavelet Packet Transformation

First, three-level wavelet packet decomposition is performed to each spectrum. Then the hard threshold technique proposed by Donoho [7] is adopted to identify and zero out the wavelet coefficients which are likely to arise from noise. Donoho proposes to use the following threshold T, which is called a universal threshold,

$$T = \sqrt{(2\log_2(\alpha))}, \tag{3}$$

where $\alpha = l * \log_2(l)$, l is dimension of the original spectrum.

The basic policy for hard threshold is keep or kill. The absolute values of all coefficients are compared to the fixed threshold T. If the magnitude of all transform coefficients is less than T, the coefficients is replaced by zero.

After the de-nosing process, in order to extract the emissions and absorptions accurately, we have to determinate the shape of the local continuum in the spectra first. The smoothed low frequency coefficients in wavelet packet decomposition is a good representation of the continuum, because it contains all the information at a very low spectral resolution [8]. The scale of wavelet packet decomposition has to be decided, which requires *a priori* knowledge of the spectra. In experiments, we adopt a total seven scales wavelet packet decomposition. Then, by truncating the smoothed wavelet coefficients and reconstructing the spectra, the feature spectra F can be obtained, which is shown by the solid line in Fig. 2 (b). As shown in the figure, the feature spectra is not very satisfying. Here we use the following method to optimize the feature extraction process.

The approximate distribution of the noise at each wavelength can be estimated by:

$$N(\lambda) = D(\lambda) - S(\lambda), \tag{4}$$

where λ represents the wavelength, $D(\lambda)$ indicates the de-noised spectrum by using threshold technique, $S(\lambda)$ is the original spectrum.

We define STD as the noise's local standard deviation around wavelength λ. Total i wavelengths are used to calculate $STD(\lambda)$:

$$STD(\lambda) = std[N(\lambda - i/2), \ldots, N(\lambda), \ldots, N(\lambda + i/2)]. \tag{5}$$

Then an emission band can be detected if $F(\lambda) > \delta STD(\lambda)$, and an absorption is detected if $F(\lambda) < -\delta STD(\lambda)$. Because only absorptions exist in the most of stellar spectra, so we just need to extract the absorptions. As is the case in more conventional methods, a value of $\delta = 3$ implies a confidence level for the detection greater than 99.9% [8]. Considering uncertainties in the noise estimation and calibration problem, we adopt $\delta = 4$ in the experiments. The $-4STD(\lambda)$ line is given in Fig. 2 (b) by dashed line. An example of feature spectra extracted using this method is shown in Fig. 2 (c).

3.2 Classification

In experiments, four expert networks are employed and they are chosen as radial basis function (RBF) neural network model. Different training set and various number of hidden layer neurons are chosen for these experts. The gating network is trained by the gradient descending algorithm according the input pattern and outputs of each local expert.

In order to obtain input pattern of the ME model, principle component analysis (PCA) technique is adopted to reduce dimension of the feature spectra to 20. For the local experts target output vector, we use one-of-k encoding method. The stellar data set is divided into two parts: training set and testing set which have 87 and 74 spectra respectively. The ME model is built using the training set. Fig. 3 shows the outputs of the trained gating network. For each sample,

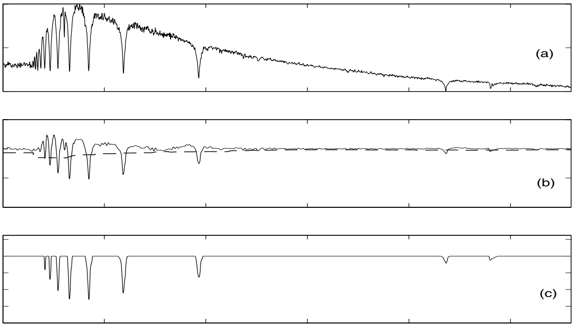


Fig. 2. (a) Original spectra. (b) De-noised and subtracted the continuum (*solid line*), $-4STD(\lambda)$ (*dashed line*). (c) Feature spectra

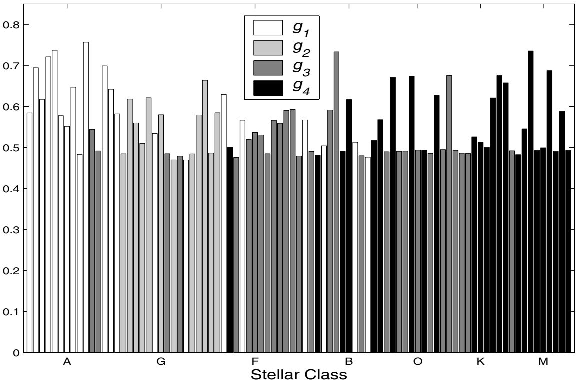


Fig. 3. Outputs of the gating network

Table 1. CCR comparison of ME, RBF and BP network model

	ME	RBF	BP
CCR	93%	91%	79%

the maximal g_k is drawn, which indicates the kth expert is very suitable for the sample. After building the model, the testing set is adopted to calculate correct classification rate (CCR), which is shown in Table 1.

As a comparison, we also adopt a single RBF neural network and single BP network to do the final classification [9]. The inputs of the RBF/BP network is same as the ME model. Classification results are also shown in Table 1, which proves the advantages of ME model introduced in Section 2.

In order to evaluate the effectiveness of the proposed wavelet packet feature extraction method, we employ PCA for reducing the dimension of the original spectra to 20 directly. Then the feature vectors are used as input of the ME model. With the same training and testing samples, the CCR is only 77%, which is much lower than that of using the proposed feature extraction method.

4 Conclusion

In this study, ME model is adopted to classify stellar spectral data. The ME can divides a large, difficult task into appropriate simple and each of subtasks can be solved by a very simple expert network, which is easier to implement in application than other methods. In order to obtain the input patterns of the ME model, a feature extraction method based on wavelet packet transformation is presented. Experimental results show that the CCR of ME model is higher than that of using either a single RBF or BP neural network, which demonstrate that it is a very promising technique for spectral classification.

Acknowledgements

This work was fully supported by a grant from the National Natural Science Foundation of China (Project No. 60275002) and the project sponsored by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry.

References

1. Rodriguez, A., Acray, B., Dafonte, C., Manteiga, M., Carricajo, I.: Automated Knowledge-based Analysis and Classification of Stellar Spectra Using Fuzzy Reasoning. Expert Systems with Applications, **27** (2004) 237–244
2. Odewahn, S. C., Stockwell, E. B., Pennington, R. L., Humphreys, R. M., Zumach, W. A.: Automated Star/Galaxy Discrimination with Neural Networks. Astronomical Journal, **103** (1992) 318–331
3. Jacobs, R. A., Jordan, M. I., Norlan, S. J., Hinton, G. E.: Adaptive Mixtures of Local Experts. Neural Computation, **3** (1991) 79–87
4. Yumlu, M. S., Gurgen, F. S., Okay, N.: Financial Time Series Prediction Using Mixture of Experts. Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg New York **2869** (2003) 553–560
5. Kurtz, M. J.: The MK Process and Stellar Classification. David Dunlap Observatory, Toronto (1984)
6. Jacoby, G. H., Hunter, D. A., Christian, C. A.: A Library of Stellar Spectra. The Astrophysical Journal Supplement Series, **56** (1984) 257–281
7. Donoho, D. L.: De-noising by Soft-Thresholding. IEEE Trans. on Information Theory, **41** (1995) 613–627
8. Starck, J. L., Siebenmorgen, R., Gredel, R.: Spectral Analysis Using the Wavelet Transform. The Astrophysical Journal, **482** (1997) 1011–1020
9. Jiang, Y. G., Guo, P.: Spectral Analysis and Recognition Using Multi-scale Features and Neural Networks. Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg New York **3174** (2004) 369–374

A Neural Network Model for Extraction of Salient Contours

Qiling Tang[1], Nong Sang[1,*], and Tianxu Zhang[2]

[1] Institute for Pattern Recognition and Artificial Intelligence
Huazhong University of Science and Technology, Wuhan 430074, China
tqlinn@sohu.com, nsang@hust.edu.cn
[2] Key Laboratory of Ministry of Education for Image Processing and Intelligent Control
Huazhong University of Science and Technology, Wuhan 430074, China
txzhang@hust.edu.cn

Abstract. In this paper, we construct a neural network structure to model the mechanisms of visual perception for salient contour – for some stimulus conditions, the response of a visual central stimulus is suppressed and for other conditions the response is enhanced. The proposed method which distinguishes between contours and texture edges can effectively eliminates surround textures, and at the same time, enables preservation of smooth contour. In particular, while some contours embedded in cluttered background are destructed due to surround disturbance, our approach can restore them better.

1 Introduction

Extraction of contours is a critical problem for object recognition tasks. A higher-level processing must include methods for discriminating nontexture edges from texture edges and for grouping edges into bounding contours. In neurophysiological view, a salient contour integration is achieved in the primary visual cortex (V1) by means of "context-dependent" influences [1], that is, the integration process is affected by the pattern coherent and spatial configuration [2].

Parallel physiological and anatomical studies show that the responses of V1 cells are significantly influenced by stimuli nearby and beyond their classical receptive fields (CRF) [3]. It is observed experimentally that the response to stimulus in the CRF is suppressed significantly by similarly oriented stimuli in the surround – iso-orientation inhibition, and this inhibition is converted to strong facilitation with the addition of collinear stimuli in the surround [3],[4],[5]. Furthermore, the enhancement depends only on the immediately adjacent surround (i.e., increasing the surround width beyond this adjacent region had no effect on the enhancement), whereas the suppression depended on a much wider surround region [6]. Accordingly, the contour detection model based on visual mechanism is iso-orientation surround (2D) inhibition and collinear (1D) facilitation.

Contour saliency is determined by Gestalt properties such as smoothness and proximity which make the smooth contour more salient than a jagged one [2]. This provides the psychological basis for extraction salient contours from cluttered background.

[*] Corresponding author.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 316–320, 2005.

2 Computational Models

2.1 Neural Network Structure

We construct a neural network to model the mechanisms of visual perception for salient contour, as shown in Fig.1. This structure is similar to Kohonen network, but their learning rules are not identical.

Psychophysical and physiological evidence suggested that the visual input is first decomposed by local mechanisms or channels tuned to specific properties such as orientation, spatial frequency and direction of motion [2]. The input layer consisting of twelve units is used to decompose the original image into different orientations. The orientation decomposition process is performed by Gabor energy filters. Each unit represents a preferred orientation. In the output layer, each neuron corresponds to each pixel in original image, and is connected to all units in the input layer. Thus, all connecting weights of each output neuron represent twelve different orientation responses of the corresponding pixel. This is analogous to "hypercolumns" proposed by Hubel and Wiesel [7], which consist of cells which respond to the same spatial position in the retina, but with different preferred orientations [1].

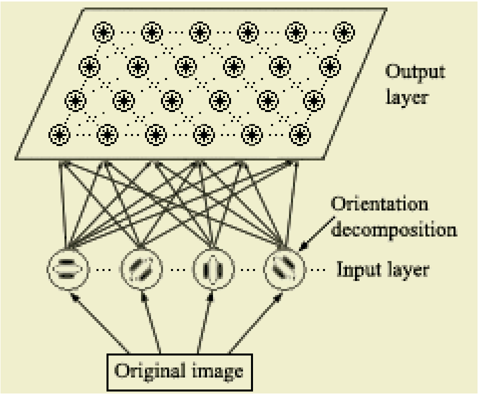


Fig. 1. The structure of the neural network

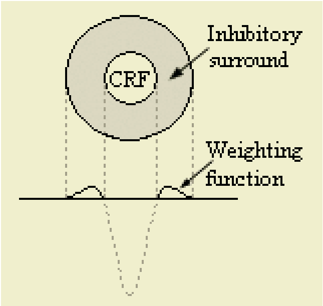


Fig. 2. The inhibitory weighting function

2.2 Inhibitory and Facilitatory Behavior

The response in CRF is subject to iso-orientation inhibition from its surround, on the other hand, receives collinearity excitation from its neighbours. We imitate the visual mechanisms by the above network model to extract smooth contour from complex background.

In the output layer, each neuron interacts with surrounding neurons. For the inhibition behavior, the interactions between neurons depend on distances and orientations. As observed experimentally, the more similar their orientations are and the closer their distances are, the stronger the inhibitory function is.

We use the following operator to simulate the fashion of the inhibitory strength varying with the distance, as shown in Fig.2.

$$w_\sigma(\Delta j) = \frac{1}{\left\| H(LoG_\sigma) \right\|_1} H(LoG_\sigma(\Delta j)) \tag{1}$$

$$H(z) = \begin{cases} 0, & z < 0 \\ z, & else \end{cases} \tag{2}$$

where Δj denotes the Euclidean distance of the stimuli outside CRF and the center, $\|\cdot\|_1$ denotes L_1 norm, σ is the size of CRF, the function $H(z)$ ensures that inhibitory action only occur outside CRF, and $LoG_\sigma(\Delta j)$ is the following Laplacian of Gaussian operator,

$$LoG_\sigma(\Delta j) = \left(\frac{\Delta j^2 - \sigma^2}{\sigma^4} \right) e^{-\frac{\Delta j^2}{2\sigma^2}} \tag{3}$$

To describe the influence of orientation contrast, the following operator is used,

$$w(\Delta i) = e^{-8(\Delta i/12)^{1.5}} \tag{4}$$

where Δi denotes orientation contrast between the centre and surrounding stimuli.

So far as enhancement of smooth aligned contours is concerned, we consider the pixels in the smooth contour should have orientation continuity with its context. Accordingly, the angle between the line connecting the center with the neighbours and the orientation of the center is no more than $\pi/4$, and only the nearest neighbours outside CRF from the center are considered due to the dependence of the enhancement only on the immediately adjacent surround. We still adopt the same $w(\Delta i)$ as in (4) to characterize the influence of orientation contrast in the enhancement behavior.

2.3 Algorithm Implementation

First, the original image is decomposed into twelve orientations by the input layer. Each pixel has a response in per orientation of Gabor energy, and the value of the response is taken as the initial value of the weight connecting the neuron in the input layer with the corresponding ones in the output. Thus the physical meaning of weight W_{ji} denotes the Gabor energy of the j th pixel in the i th orientation.

On account of neurons in the output layer interacting with surrounding ones, connecting weights are dynamically tuned according to their local features. The corresponding weights are decreasing as the cluttered textures are gradually suppressed; on the other hand, the corresponding weights are increasing as smooth contours are progressively enhanced.

The i th orientation response of the j th neuron receives the inhibition and the enhancement from its surround, respectively, as follows:

$$S_{ji}(t) = \sum_{j^{'}} \sum_{i^{'}} w_\sigma(\Delta j) w(\Delta i) W_{j^{'}i^{'}}(t) \tag{5}$$

$$E_{ji}(t) = \sum_{j^{'} \in \Omega_{ji}} \sum_{i^{'}} w(\Delta i) W_{j^{'}i^{'}}(t) \tag{6}$$

where $S_{ji}(t), E_{ji}(t)$ are inhibition and enhancement terms, respectively, Ω_{ji} denotes the set of the context meeting the enhancement conditions.

The weight updates are illustrated as the following dynamic equation:

$$W_{ji}(t+1)=W_{ji}(t)+\eta(t)\left[w_e E_{ji}(t)-w_s S_{ji}(t)\right] \tag{7}$$

$$\eta(t)=e^{-0.02(t-1)} \tag{8}$$

where t indicates the number of iterations. w_e, w_s determine the degree of enhancement and inhibition, respectively. That the factor $\eta(t)$ gradually decreases with the iterations process. This shows that the algorithm has the course of adjustment from coarse to fine. For every iteration, $W_{ji}(t)$ is normalized to a fixed dynamic range (between 0 and 1) to eliminate amplitude differences. To terminate the iteration, we can define the maximum number of iterations. Finally, the winner-take-all strategy is conducted in the competitive layer, i.e., the maximum of all weighs of each neuron treat as the output of the neuron.

3 Experimental Results

We use two natural images to verify the performance of the proposal, as showm in Fig.3. Contours are extracted through using the strategies of nonmaxima suppression and hysteresis thresholding by Canny.

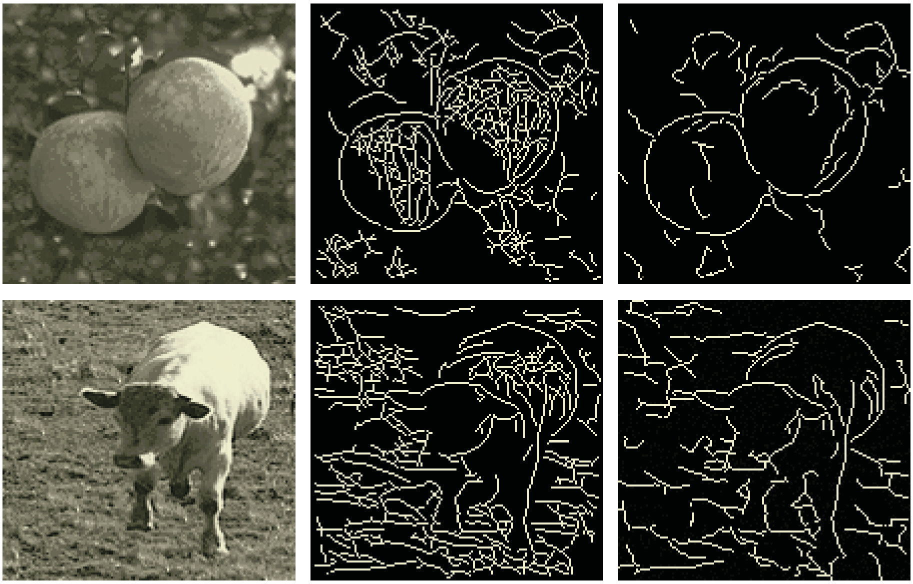


Fig. 3. The results of extraction of salient contours. From left to right: original image, Gabor energy operator and our proposal

The size of the receptive field σ is determined by the standard deviation of the Gaussian function in Gabor functions. In this experiment σ is 2.0 (see [8] for details) and the parameters w_e, w_s are 0.5 and 3.0, respectively.

Compared to the result of edge detection using Gabor energy operator without inhibitory and facilitatory functions, the approach is able to eliminate more unwanted textures, and extracts contours more complete. This advantage is particularly useful for object recognition based on the contour because of reducing the interference of the extraneous background textures.

4 Conclusions

So far, many useful algorithms have been presented in the region of computer vision. But there still exist many puzzles which visual mechanisms can easily do with, but these algorithms are hard to obtain satisfying results. Considering object detection is the visual behavior, we try to simulate vision versus smooth contour detection capability. Although the proposed models are as yet very primitive compared to the actual visual mechanism, our results show the approach is feasible to detect smooth contour from cluttered surround that cannot be easily detected by other methods.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Contract 60135020.

References

1. La Cara, G.E., Bettini, M., Ursino, M.: Extraction of Salient Contours in Primary Visual Cortex: A Neural Network Model Based on Physiological Knowledge. Proc. IEEE Conf. EMBS (2003) 17-21
2. Bonneh, Y., Sagi D.: Configuration Saliency Revealed in Short Duration Binocular Rivalry. Vis. Res, **39** (1999) 271-281
3. Li, Z.P.: Visual Segmentation by Contextual Influences via Intracortical Interactions in Primary Visual Cortex. Comput. Neural Syst, **10** (1999) 187-212
4. Levitt, J.B., Lund, J.S.: Contrast Dependence of Contextual Effects in Primate Visual Cortex. Nature, **387** (1997) 73-76
5. Yen, S.C., Finkel, L.H.: Extraction of Perceptually Salient Contours by Striate Cortical Networks. Vis. Res, **38** (1998) 719-741
6. Xing, J., Heeger, D.J.: Measurement and Modeling of Center-surround Suppression and Enhancement. Vis. Res, **41** (2001) 571-583
7. Hubel, D.H., Wiesel, T.N.: Functional Architecture of Macaque Monkey Visual Cortex. Proc. R. Soc. Lond, **198** (1977) 1-59
8. Grigorescu, C., Petkov, N., Westenberg, M.A.: Contour Detection Based on Nonclassical Receptive Field Inhibition. IEEE Trans. Image Processing, **12** (2003) 729-739

A Mechanism for Extracting Optical Virtual Contours of Discrete Dot Stimuli

Eunhwa Jeong and Keongho Hong

Department of Information and Communication, Cheonan University
115, Anseo, Cheonan, Chungnam 330-704, Korea
{khhong,ehjeong}@infocom.cheonan.ac.kr

Abstract. This study proposed a novel mechanism that can extract optical virtual contours from dot stimuli referenced on visual system. In this experiment we deal with optical virtual contours from simple arrays of discrete dot figures as well as virtual regular polygon contours that include dots positioned on the vertices of imaginary regular polygon. It is well known that some simple arrays of discrete dots can perceive as successive virtual lines. In this paper we proposed a visual mechanism that predicts the shape of virtual polygon contours. This system employs an important six-stage process: image acquisition, contrast extraction, simple virtual stimuli extraction, virtual stimuli restoration, virtual contour extraction and image smoothing process. The proposed system is demonstrated on a variety of array-type and polygon-type dot stimuli displays. The results of the experiment showed that the proposed model was successful not only in extracting the successive virtual lines between dots, but also in extracting the virtual polygon contours from dot stimuli.

1 Introduction

The perceptual phenomenon relating to dot stimuli was investigated that some simple arrays of discrete dots appear as a continuous curve, whereas others are perceived as an angular contour or as consisting of separate groups of dots (Fig.1, Fig.2). These results suggest the psychological existence and importance of virtual lines in the visual processing of dot stimuli. Yet the objective conditions that must be fulfilled for the unambiguous perception of a line of any shape are poorly understood. Although it is possible to describe the phenomena shown in Fig. 1, such descriptions provide no quantitative measure for predictions. In experiment, or objective is to deal with dot figures of dots positioned on both triple dot arrays (Fig. 2) and the vertices of an imaginary regular polygon (Fig. 3). In Fig.3 it can be perceived ambiguously, either as a polygon or as a circle. It is not clear what makes some of these figures almost naturally seen as polygons and others as circles. From the Gestalt point of view, both polygon and circle are "good" forms.

A number of points can be made relating the present results to suggestions made about the underlying visual processes, and in connection with a number of well-known perceptual illusions. Cealli and Umansky (1976) proposed that the recognition

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 321–326, 2005.

of shape in dot stimuli is based on an interpolation process. It can be stated, in relation to the visual processes involved, that the contours need to be created somewhere in the visual process, since the dots, in themselves, only provide the visual system witch represents the pair-wise relation between (neighboring) dots as virtual lines (Juliez 1971; Marr 1982; Smits and Vos 1987; Stevens 1978) and that the perceived form results from an analysis of these virtual lines. Most papers regarding with dot stimuli deal with the perceptual phenomenon using psychological experiments. Unlike these types of studies, we proposed a visual mechanism for recognizing virtual contours betweens dot stimuli and forming virtual polygon contours from dot stimuli based on the feature extraction of the visual cells. This system consists of six layers: image acquisition, contraction extraction, simple virtual feature extraction, virtual feature restoration, virtual contours extraction and image smoothing process.

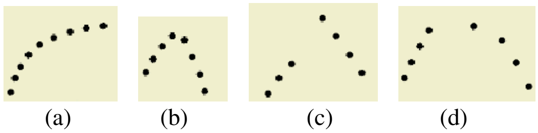


Fig. 1. The Patterns of dot array (a) are immediately perceived as a continuous curve, whereas array (b) has an angular discontinuity, and (c) and (d) have a discontinuity in the form of a separation into two sub-arrays. Or is (d) perceived as an array of type (a) or (b)

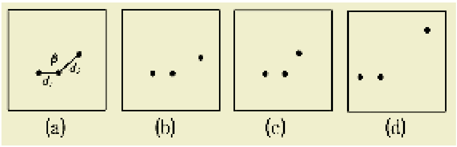


Fig. 2. The Patterns of triple dot array (a) The stimulus parameters β, d_1, and d_2 of triplets. (b)-(d) Examples of the three response categories: curve, angle/triangle, and separation, respectively

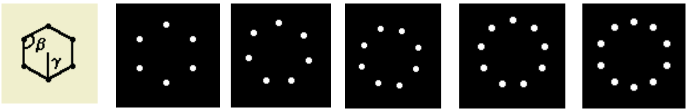


Fig 3. Virtual regular dot polygons with increasing number of dots(n) n= 6, ... , 10. The first figure shows the stimulus parameters, angle β and size

2 Visual System

The mammalian visual system receives input in the form of visible light. Photoreceptors in the retina absorb this light, emitting neural signals on the process, which stimulates bipolar and retinal ganglion cells. Information from each retina passes along the optic nerves, though the optic chasm and into the lateral geniculate nucleus (LGN). The cells within the LGN do not appear to exert any profound transformation upon the neural information they receive other than organizing retinal signals into right and left visual field components. From the LGN, information is topographically mapped onto the visual cortex where it is further processed by simple and complex cells. The terms "simple", "complex" etc. refer to the types of stimuli that elicit responses from these from these cells. In a serial model of vision processing, the visual input required to activate a cell within the cortex becomes progressively more complicated further along the visual pathway. The architecture of the proposed system is still developed referencing features of receptive fields found in visual pathway[4].

3 Extraction of Virtual Contours

3.1 Image Acquisition (Layer U_0)

In this case of input from color image, the mammalian visual system classifies color information. Since the virtual contour figure, which is seen as lying on top of the remaining parts of patterns, is represented with white, then the others usually are different colors. The color classification in virtual contour figures is not of great significance. Thus, for simplification, we remove the process of color classification, and simply convert a color image to a binary one.

3.2 Contrast Extraction by LGN (Layer U_1)

The system first detects low-level feature, such as contrast in the binary image (layer U_0). This work performs by spatial filtering. A set of two-dimensional DOG (Difference Of two Gaussians) functions with an adequate space constant is used for the spatial filters. The function is defined by

$$DOG(x,y) = \frac{1}{2\pi\sigma_e^2} e^{-r^2/2\sigma_e^2} - \frac{1}{2\pi\sigma_i^2} e^{-r^2/2\sigma_i^2} \tag{1}$$

Where r represents distance from the origin, σ_e represents the space constant of excitatory region and σ_i represents that of inhibitory region. This filter corresponds to an on-center and off-surround receptive field found in mammal's retina[2]. The ratio of space constants σ_i / σ_e is set to 1.6 for the filter. This ratio gives a close approximation to the idea Laplacian operator[6].

3.3 Simple Virtual Feature Extraction (Layer U_2)

Simple cells are excellent at detecting the presence of *simple* visual virtual features, such as lines and edges of *a particular orientation*. In consideration of the orientation selectivity property of simple cells, a set of asymmetrical two-dimensional three Gaussian filters for *eight preferred orientations* is used. The filter with vertical preferred orientation is defined by

$$g(\sigma, x) = e^{-x^2/2\sigma^2} \tag{2}$$

$$d_7(x,y) = \left(g(\sigma_e, x-l) - g(\sigma_i, x) + g(\sigma_e, x+l)\right) \cdot g(\sigma_{en}, y) \tag{3}$$

where l represents the space constant between excitatory and inhibitory region. The filter is detecting simple visual responses between the dot stimuli. This filter corresponds to a simple cell receptive field found in mammal's visual system[4].

3.4 Virtual Feature Restoration (Layer U_3)

As simple cells, complex cells respond to stimuli such as lines and edges of a particular orientation. However, the exact location of the stimulus is of no concern to a complex cell, as long as it is within that cell's receptive field. A set of two-dimensional Gaussian filters for eight orientations is used. The filter with vertical preferred orientation is defined by

$$d_3(x,y) = e^{-x^2/2\sigma_x^2} \cdot e^{-y^2/2\sigma_{en}^2} \tag{4}$$

This filter corresponds to a complex cell receptive field found in mammal's visual system[4].

3.5 Virtual Contour Extraction (Layer U_4)

The process forms polygon contour from the extracted stimuli using image operation and feedback process. The process combines the extracted simple visual feature stimuli of the previous visual feature restoration using image operation and does feedback process to the simple visual feature extraction. The output $u_4(x,y)$ of layer U_4, can be expressed as following;

$$d_4(x,y) = \prod_{\alpha=0}^{7} 1[\iint_{A_4} d_4(\xi,\eta,\alpha)\cdot u_3(x+\xi, y+\eta, \alpha)d\xi d\eta - \theta_9] \tag{5}$$

$$u_4(x,y) = \left[d_4(x,y) \vee u_0(x,y)\right] \tag{6}$$

where $\vee$ represents logical OR operation. The output of this layer feedbacks to the simple visual feature extraction process until extracting the features between the extracted stimuli of the previous stage and forming virtual lines as well as polygon contours.

3.6 Image Smoothing Process (Layer U_5)

This is a smoothing process for image improvement. The filter of layer U_5 recovers the weaken or reduced stimuli. A set of two dimentional Gaussian filters is used.

4 Experimental Results

In order to show the performance of the system, experiments carried out using the proposed dot stimuli figures consisting of triple dot array figures and regular virtual polygon contour figures. Examples of the experiments are shown in Fig.4, Fig.5 and Fig.6. The results of Fig.4 and Fig.5 present optical virtul lines from triple dot array figures. The results of regular virtul polygon contours from dot stimuli figures are given in Fig.6 . These results show that the performance of the system is sufficiently general.

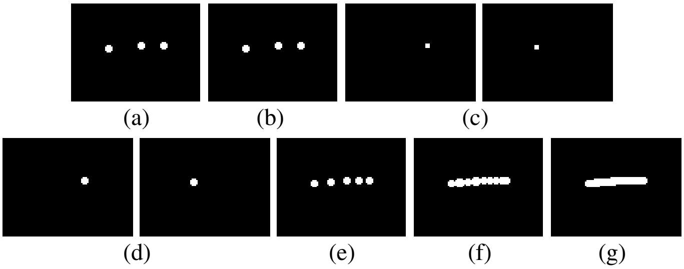


Fig. 4. The results of virtual continuous curved contour from triple dot stimuli (a) input image (b) image after contrast extraction (c) images after simple visual virtual feature extraction (d) images after virtual feature restoration (e) image after first iteration for virtual contour (f) virtual contour image after the second iteration (g) output image

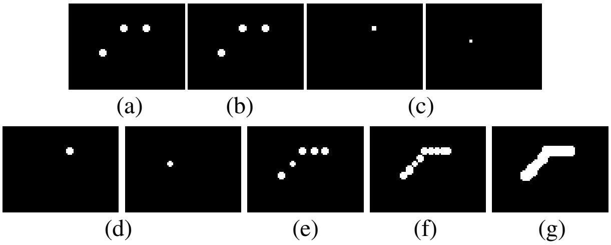


Fig. 5. The results of virtual angular contour from triple dot stimuli (a) input image (b) image after contrast extraction (c) images after simple visual virtual feature extraction (d) images after virtual feature restoration (e) image after first iteration for virtual contour (f) virtual contour image after t second iteration (g) output image

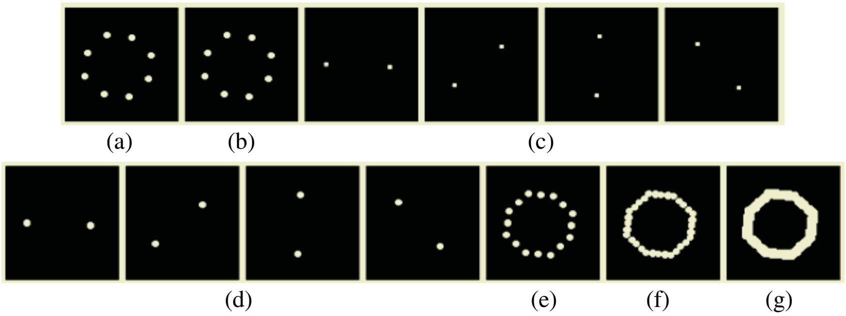


Fig. 6. The results of virtual polygon contour (a) input image (b) image after contrast extraction (c) images after simple visual virtual feature extraction (d) images after virtual feature restoration (e) image after first iteration for virtual contour (f) virtual contour at second iteration (g) output image

5 Conclusion

A novel approach for the perception of virtual polygon contour from discrete dot stimuli is proposed using the mechanism of feature extraction in visual system. Our model consists of six layers: image acquisition, contraction extraction, simple visual virtual features extraction, virtual feature restoration, virtual contours extraction and image smoothing. This proposed model is demonstrated on a variety of dot stimuli displays. The results of the experiment showed that the proposed mechanism was successful not only in extracting the successive optical virtual lines between dots, but also in perceiving virtual polygon contours from dot stimuli.

References

1. Marr, D.: Vision: A Computational Investigation into the Human Representation and Processing of Visual Information. W. H. Freedom & Company (1982)
2. Smits, J.T.S., Vos, P.G.: A Model for the Perception of Curves in Dot Figures: The Role of Local Sailence of 'virtual' lines. Biological Cybernetics, **54** (1986) 407-416
3. Smits, J.T.S., Vos, P.G.: The Perception of Continuous Curves in Dot Stimuli. Perception, **16** (1987) 121-131
4. Hubel. D.H. and Wiesel. T.N.: Receptive Fields Binocular Interaction and Functional Architecture in the Cat's Visual Cortex. J Physiology, (1962) 106 -154
5. Marr, D. and Hildreth, E.: A Theory of Edge Detection. Proc. R. Soc. London, **207** (1980) 187-217

Using Self-organizing Map for Mental Tasks Classification in Brain-Computer Interface

Hailong Liu, Jue Wang, and Chongxun Zheng

Key Laboratory of Biomedical Information Engineering of Education Ministry
Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China
lhl2001@tom.com

Abstract. One problem in Brain-Computer Interface (BCI) is the requirement of online training of classifiers, since EEG patterns vary greatly at two separate time with long period. In this paper, the use of Self-Organizing Map (SOM) as an adaptive classifier for mental tasks classification was proposed. As for SOM, there are two cases about the labeling of map units, which correspond to semi-supervised and unsupervised algorithm respectively. In one case, the map units are labeled according to the labels of training patterns. In the other case, the map structure information, e.g., the U-matrix, is used to cluster map units. The ability of SOM to recognize mental task was analyzed for both cases. The organized SOM is tested on testing patterns. The averaged classification accuracy of 96.2% and 90.8% across 10 task pairs was obtained for both cases respectively. This result indicates the feasibility of online training of SOM for mental tasks classification.

1 Introduction

A Brain-Computer Interface (BCI) can extract people's intents by recognizing their brain states, which are characterized by the corresponding EEG patterns. Up to now, many research laboratories around the world have been devoted to this technology (reviewed in [1]). BCI technology provides brain with a new communication and control channel that does not depend on brain's normal output pathways of peripheral nerves and muscles. The recognizing process of a BCI system is just the classification of EEG patterns related to different mental states. BCI systems detect and extract characteristic features from EEG signals and translate them into control signals for external devices.

A major problem encountered in the implementation of a BCI system is online training of classifiers. There are mainly two reasons for using online training process. One is that subjects can not maintain exactly the same mental states for a long time [2]. EEG patterns recorded at two separate time varies greatly even though subjects perform the same one mental task. Thus, an online training process is necessary to capture the characteristics of EEG signals. Another reason is that offline analysis is hard to provide sufficiently large data set for training classifiers. Online training process can solve the problems mentioned above. In order to achieve this goal, some clustering algorithms should be used.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 327–332, 2005.

The Self-Organizing Map (SOM) neural network is an unsupervised algorithm that clusters similar input pattern vectors. In the unsupervised learning process, no prior knowledge like labels is needed and the information required is just the input patterns. Besides reducing the number of data points by clustering, it also projects the data nonlinearly onto a lower-dimensional display map with each map unit's weighting vector being a clustering center. SOM presents the structure of the input data on the lower-dimensional display map by preserving the relations among the original data points. The characteristic of needing no prior knowledge makes SOM appropriate for online learning and data processing.

SOM has been successfully used on mental workload classification [3] and detection of visually apparent EEG phenomena [4]. SOM can be able to be applied to EEG data for recognition of mental states. SOM is a convenient tool for EEG data visualization and analysis, and this can be implemented in real time. In this paper, we will test SOM's ability of distinguishing EEG patterns corresponding to complex mental tasks to see its feasibility in BCI.

2 The Method of Self-organizing Map

SOM creates a low-dimensional topological map with each map unit's weight vector being a clustering center of input vectors. The map units are usually arranged in a regular or hexagonal grid. During the unsupervised learning process, the best-matching weight vector of input pattern and its topological neighbors on the map are updated together. So, the neighboring map units have similar weight vectors. The relations among weight vectors are well preserved on the topological map. The maintenance of similarity relations makes the visualization of the structure of EEG patterns more easily understandable. It can be seen that besides the clustering process, SOM also projects input data nonlinearly onto a low-dimensional map. For any input vector, a corresponding unit on the map will be specified such that the map unit's weight vector and the input vector have the closest distance. Similar inputs project near to each other onto the map.

The SOM training algorithm is iterative. At the beginning, the weight vectors can be initialized linearly. For input vector x, the map unit whose weight vector is closest to x is called the Best-Matching Unit (BMU) of x. The weight vector of BMU is denoted by m_{c}. During each learning step t, the weight vectors of both BMU and its topological neighbors on the map are updated:

$$m_i(t+1) = m_i(t) + \alpha(t)h_{ci}(t)[x(t) - m_i(t)] \tag{1}$$

where $x(t)$ is the input vector, $h_{ci}(t)$ is the neighborhood function around the BMU, and $\alpha(t)$ is the learning rate. A Gaussian $h_{ci}(t) = \exp\left(-\frac{\|r_c - r_i\|^2}{2\delta(t)^2}\right)$ is usually used as the neighboring function.

There are two ways in which SOM can be utilized for mental tasks classification. One is that the map units are labeled according to those training patterns that most frequently project onto them after the clustering process. New patterns are classified into the class of the map unit to whose weight vector they are closest. Another way is to cluster the weight vectors into different classes using the information of map structure, e.g., the U-matrix, instead of using the labels of input patterns after the clustering process. For the later case, prior knowledge like labels is not needed. This paper uses both methods to see whether SOM is able to be used for online training. The SOM is implemented by using the SOM Toolbox [5].

3 EEG Data and Features Extraction

3.1 Experiment Data

The data used in this study was originally obtained by Keirn and Aunon (detailed information about the experiment can be seen in [2]). EEG signals were recorded from six channels (C3, C4, P3, P4, O1 and O2). Eye blinks was detected by a separate channel. The signals were band-pass filtered at 0.1-100Hz and sampled at 250Hz. Seven subjects participated in the experiment and each subject performed five mental tasks (i.e., baseline measurement, mental multiplication, mental letter composing, geometric figure rotation, and visual counting). An EEG recording of ten seconds constituted a trial and a session comprised five such trials. Subjects performed one session for each task on one day, and different sessions were recorded on separate weeks. Subject 2 and subject 7 finished one session, subject 5 finished three sessions, and each of the remaining subjects finished two sessions.

3.2 Preprocessing and Features Computation

To remove the ocular artifacts of EEG signals, a time-domain regression method was used, which subtracted weighted EOG from the EEG signals (described by Girton [6] and reviewed in [7]). The weight values used in our study were 0.1 for C3 and C4, 0.05 for P3 and P4, and 0.025 for O1 and O2.

EEG time series from each channel were divided into 1-second segments within each session of task for each subject. The neighboring segments overlapped each other by 0.9s. The power spectrum of each segment that was estimated by using FFT algorithm was reduced to 10 features by first equally dividing frequency band into 10 sub-bands and then computing the weighted sum of power values within each sub-band. Features from six channels were concatenated to form sixty dimensional feature vectors.

Five tasks formed 10 task pairs. In our study, classification was performed on each task pair within sessions for 7 subjects. Each task pair classification was repeated 1000 times, and the classification accuracy for a task pair was the averaged result over the 1000 runs. In each run of classification, SOM was trained using randomly chosen 80 percent feature vectors and tested with the remaining 20 percent.

4 Results

We first visually analyzed the map structure after SOM's clustering of the input pattern vectors. Take the task pair of Rest and Multiplication of subject 7 for an example. The topological map with labeled units and the distance matrix between units are illustrated in Fig.1. Figure 1(a) presents the map units that are labeled according to the labels of the training set vectors. A map unit has the same label as those input vectors that most frequently project onto it. The map units presented in Fig.1(c), however, are clustered by using the distance information of weighting vectors between map units. In this case, the prior information of labels is not needed. It can be seen that the result is quite similar to that presented in Fig.1(a). The distance information is reflected by the U-matrix (unified distance matrix). The U-matrix is presented in Fig.1(b), in which great values marked in darker color indicate the clustering border. Those small values marked in lighter color mean small distances between map units, and so indicate the clusters themselves. This also is visually interpreted clearly by the distance matrix presented in Fig.1(d). Element of the distance matrix illustrated by Fig.1(d) is the average distance of map unit from all its neighbors. It can be seen that the cluster border is visually distinct from the clusters. Figure 1 shows that using either the labels information of the training vectors (identified as 1st case) or the distance information of map units (identified as 2nd case) are able to obtain almost similar clustering border.

In the way of the first case, i.e., first clustering the training patterns, and then labeling the map units according to the labels of training patterns, and further testing the trained SOM on test set, task pair classification was made for all 7 subjects within each session. The averaged classification results across all ten task pairs are presented in Table 1. It can be seen that high classification accuracy is obtained by using SOM as classifiers.

Table 1. Average Percent Classification Accuracy Across All Ten Task Pairs. The symbol of "-" indicates no EEG recording.

	Subject #						
	1	2	3	4	5	6	7
1st Session	97.5	92.1	91.6	97.8	97.2	95.2	96.2
2nd Session	98.1	-	97.0	91.6	96.6	96.4	-
3rd Session	-	-	-	-	97.8	-	-

For the second case, i.e., the map units are clustered into two classes according to the map structure—the U-matrix, classification of ten task pairs within the first session of subject 7 was made. The comparison with the first case are presented in Table 2. The 2nd case is purely unsupervised leaning process. The 1st case is comparatively semi-supervised. Seen from Talbe 2, acceptable classification accuracy can be achieved using the unsupervised algorithm.

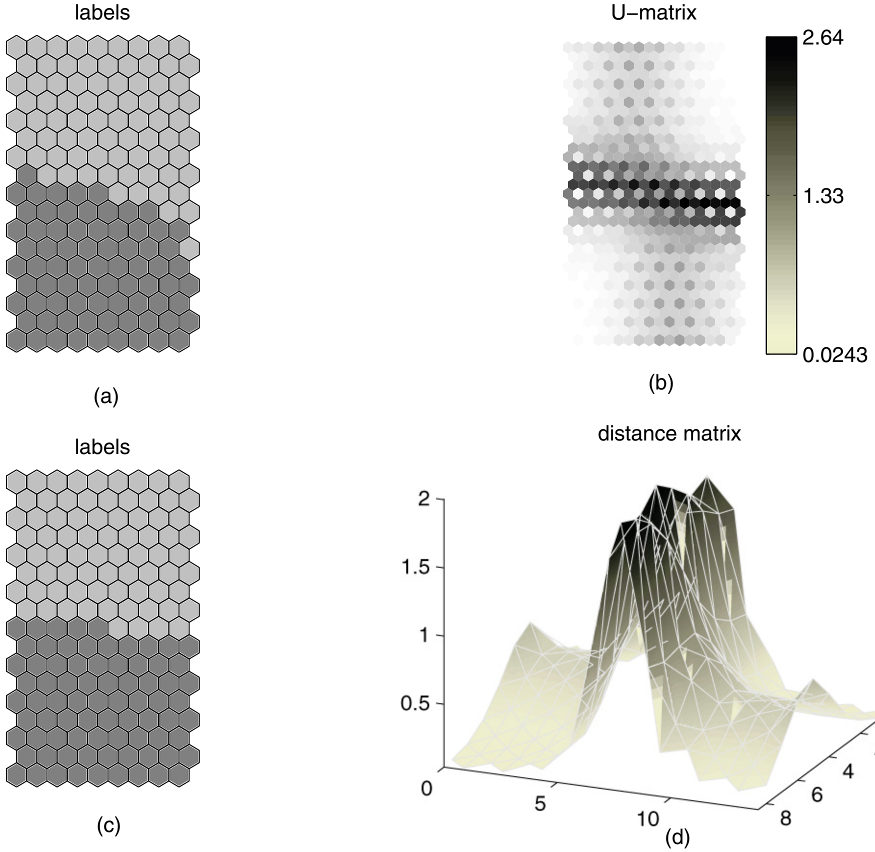


Fig. 1. Map structure for the task pair of Rest and Multiplication of subject 7. In (a), map units are labeled according to the labels of the training patterns. Map units in darker color correspond to Rest task, and the units in lighter color represent Multiplication task. (b) is the U-matrix of the map, which indicates a distinct separation of two clusters. In (c), the map units are labeled according to the map structure indicated by the U-matrix. In this case, the prior information of labels is not used. (d) is the distance matrix, each element of which is the average distance of the corresponding map unit from all its neighbors.

Table 2. Classification Accuracy in Percent for 1st Session of Subject 7. In 1st case, units of the trained SOM are labeled by using the prior label information. In 2nd case, map units are divided into two classes according to the map structure, i.e., the U-matrix.

	Task Pair #										
	1	2	3	4	5	6	7	8	9	10	Average
1st case	98.9	90.9	99.8	89.7	99.4	91.4	99.5	99.4	93.7	99.6	96.2
2nd case	93.7	76.9	96.0	89.5	96.8	73.6	93.9	97.9	93.0	97.0	90.8

5 Discussion

Under the condition of the prior knowledge of labels being known, SOM's map units can be labeled by using the prior information after its clustering process, and then, the trained SOM can be used as a normal classifier to classify new patterns online or offline. This case is appropriate for offline analysis, or first offline training and then online testing. The results presented in Table 1 indicate that SOM works well for mental tasks classification.

SOM can be used for mental tasks classification with wholly unsupervised learning. In this case, the prior knowledge of labels is not needed, and the SOM divides map units into different classes according to the map structure information, e.g., the U-matrix. The results presented above indicate that the map structure reflects right classification information, and acceptable classification accuracy can be achieved by using the map structure information. Since no prior knowledge is needed, SOM can be used as a tool for online training and data processing. Thus, SOM can be used potentially as an adaptive classifier, which can be able to solve the problem of greatly varying of patterns corresponding to a mental task after a long period.

Acknowledgment

The authors would like to thank Anderson and Keirn to provide the experiment data for analysis. This work was supported in part by the National Natural Science Foundation of China under Grant 60271025 and 30370395 and in part by the Science and Technology Department of Shaanxi Province under Grant 2003K10-G24.

References

1. Wolpaw, J.R., Birbaumer, N., McFarland, D.J., Pfurtscheller, G., Vaughan, T.M.: Brain-computer Interfaces for Communication and Control. Clinical Neurophysiology, **113** (2002) 767–791
2. Keirn, Z.A., Aunon, J.I.: A New Mode of Communication between Man and His Surroundings. IEEE Trans. Biomed. Eng., **37** (1990) 1209–1214
3. Mazaeva, N., Ntuen, C., Lebby, G.: Self-organizing Map (SOM) Model for Mental Workload Classification. In: IFSA World Congress and 20th NAFIPS International Conference, Vancouver, BC Canada (2001) 1822–1825
4. Joutsiniemi, S.L., Kaski, S., Larsen, T.A.: Self-organizing Map in Recognition of Yopographic Patterns of EEG Spectra. IEEE Trans. Biomed. Eng., **42** (1995) 1062–1068
5. Vesanto, J., Himberg, J., Alhoniemi, E., Parhankangas, J.: Self-organizing Map in Matlab: the SOM Toolbox. In: Proceedings of the Matlab DSP Conference, Espoo, Finland (1999) 35–40
6. Girton, D.G., Kamiya, J.: A Simple On-line Technique for Removing Eye Movement Artifacts from the EEG. Electroencephalogr. Clin. Neurophysiol, **34** (1973) 212–217
7. Croft, R., Barry, R.: Removal of Ocular Artifact from the EEG: A Review. Neurophysiologie Clinique / Clinical Neurophysiology, **30** (2000) 5–19

Speech Recognition Using Stereo Vision Neural Networks with Competition and Cooperation*

Sung-III Kim

Division of Electronic and Electrical Engineering, Kyungnam University
449 Wolyoung-dong, Masan City 631-701, Korea
kimstar@kyungnam.ac.kr

Abstract. This paper describes the speech recognition based on stereoscopic vision neural networks(SVNN) that has a dynamic process of self-organization that has been proved to be successful in recognizing a depth perception in stereoscopic vision. This study has shown that the process has also been useful in recognizing human speech. In the stereoscopic vision neural networks, the similarities are first obtained by comparing input vocal signals with standard models. They are then given to a dynamic process in which both competitive and cooperative processes are conducted among neighboring similarities. Finally, only one winner neuron is finally detected through the dynamic process. In a comparative study, the average phoneme recognition accuracies on the SVNN was 6.6 % higher than the existing recognizer based on Hidden Markov Models(HMM) with the structures of a single mixture and three states. From the results, therefore, it was noticed that the speech recognizer using SVNN outperformed the conventional recognizer in phoneme recognition under the same conditions.

1 Introduction

Hitherto, many studies have been conducted on the basis of Hidden Markov Models(HMM)[1],[2] and several kinds of artificial neural networks (ANNs) [3],[4],[5]. Though HMM has been regarded as a useful recognizer by producing relatively accurate probabilistic acoustic models, it still has a weakness in the viewpoint of the modeling with human-like speech understanding. As the alternative approach, therefore, ANNs such as multi-layer perceptron [3], time-delay neural network [4], or hidden control neural network [5], etc., have been introduced by modeling an information processing mechanism of physiological human brain. The one of major strength of them is in the fact that there is no need for any mathematical assumptions about statistical distributions or independence among input frames. However, there are still demerits of dealing with too many parameters in both training and recognition processes as well as structural complexity.

In the neural networks for stereoscopic vision, there are two beneficial features compared with the above-mentioned neural networks. The one thing is that it has much more simple architecture because the network parameters are always fixed and

[*] This work is supported by the Kyungnam University Research Fund, 2005.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 333–338, 2005.

not revised at any time. The other is that it has a powerful information processing capability of identifying the most likely neuron among confusable candidates. The process is made by both cooperative and competitive process among their similarities. These stereo vision neural networks(SVNN)[6],[7] process the input visual data, yielding a depth perception of a specific object in stereoscopic vision.

In the same way, it is assumed that speech recognition can be performed by the same process between vocal features as input data and memorized ones as standard models in human brain. In the processing, SVNN triggers not only competition among similarities in all possible speech candidates but cooperation among ones in temporal frames of the candidates, and finally so-called winner-take-all process plucks only one neuron from the candidates. Though it has not been found if a visual processing mechanism for depth perception is compatible with an actual hearing system for speech recognition, it is worth to apply the cognitive architecture in stereoscopic vision to speech recognition, on the viewpoint of information processing based on the neural networks. In this new approach, the recently modified algorithms of SVNN, which have been optimized through preliminary investigations [8],[9], were successful in stereoscopic depth perception. We will describe the SVNN with a dynamic process of competitive and cooperative coupling among input similarities. It would be then explored if the process works well in recognizing human speech.

2 Stereo Vision Neural Networks(SVNN)

The equations for SVNN are given as

$$\tau_1 \dot{\xi}_u^a(t) = -\xi_u^a(t) + f(\beta_u^a) \quad . \tag{1}$$

where ξ_u^a (t) is a time-dependent neural activity and f(x) is the sigmoid function. β_u^a (t) is a coefficient represented as a middle layer, that is

$$\tau_2 \dot{\beta}_u^a(t) = -\beta_u^a(t) + g(\alpha_u^a(t)) + g(\xi_u^a(t)) \quad . \tag{2}$$

where g(u) is a linear function given by

$$g(u) = u^+ = \begin{cases} u, u > 0 \\ 0, u \le 0 \end{cases} \quad . \tag{3}$$

α_u^a (t) is given as following

$$\tau_3 \dot{\alpha}_u^a(t) = -\alpha_u^a + A\lambda_u^a - B \sum_{\substack{a'=a-a_s \\ a' \neq a}}^{a+a_s} g(\xi_u^{a'}(t)) + D \sum_{\substack{u'=n-l \\ u' \neq u}}^{n+l} g(\xi_{u'}^a(t)) \quad . \tag{4}$$

A, B, D, and, τ_1, τ_2, τ_3, used in the above equations are all positive constants.

The equilibrium solution can be considered by assuming that $\dot{\xi}_u^a = \dot{\alpha}_u^a = \dot{\beta}_u^a = 0$. Therefore, the equations of (1), (2), (4) can be represented as following

$$\xi_u^a(t) = f(g(\alpha_u^a) + g(\xi_u^a)) \quad . \tag{5}$$

The solution of the equation can be an intersection between y = ξ and y = $f(g(\alpha)+g(\xi))$. It is noticed that the value of intersection point between two equations rises in proportion to an increase of α_u^a (t). As a result of the dynamic process, ξ_u^a (t) will approach a certain stable point, so that only one neuron would be determined regardless of initial conditions of SVNN equations.

3 Mechanism of Speech Recognition Based on SVNN

In speech recognition based on SVNN, the similarities are first obtained by comparing the input vocal signals with the trained standard models. The similarity map is then given to the dynamic process with competitive and cooperative coupling. Figure 1 shows the dynamic process among input similarities.

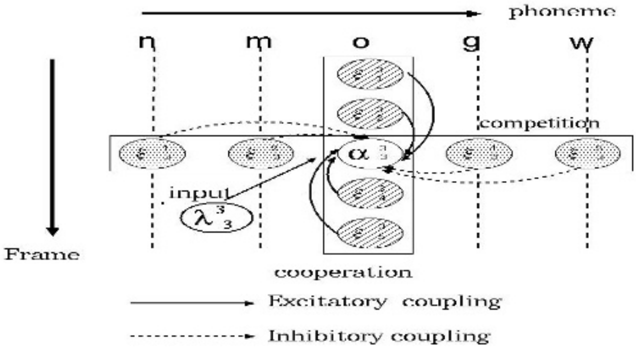
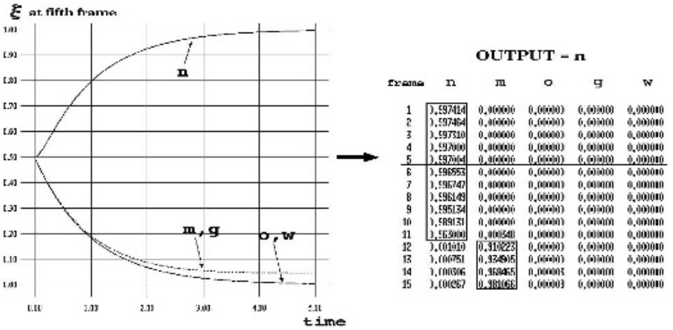


Fig.1. Competitive coupling among similarities in all possible candidates(for example, 5 different phonemes) and cooperative coupling in temporal frames

As shown in this figure, the first layer, α_u^a (t), is influenced by not only input similarities but neighboring neural activities. Namely, it is activated by an inhibitory coupling among candidates and by an excitatory coupling among neighboring frames as well. The dynamic process ultimately makes each neuron to converge to a certain final value independent of the initial conditions of parameters in neural net equations.

Figure 2 shows an example of time-dependent behaviors of ξ_u^a (t) influenced by α_u^a (t). It starts with the initial value preset in the neural net equations. α_u^a (t) first takes values corresponding to λ_u^a (t). ξ_u^a (t) then updates its value through competition-cooperation process among neighboring neural activities. In this figure, for example, the value of ξ_u^a (t) in /n/ grows to converge to a maximum point, while others fall down to approach minimum values.

The binocular neurons compete over the inhibitory coupling area and simultaneously cooperate over the excitatory area. Through the dynamic process, therefore, only one specific neuron wins over the other neurons whose activities are damped to minimum points. As shown in these figures, the dynamic process ultimately makes both α_u^a (t) and ξ_u^a (t) to move toward a stable point, namely 0 or 1, regardless of the initial conditions of parameters. This figure also shows an example of output values through dynamic process in which parameters were determined experimentally. The neuron with value of near 1 is called a winner neuron, whereas one with value of 0 is called a loser neuron. Since /n/ has more winner neurons than others, it is finally recognized as the most likely candidate to input speech.



OUTPUT - n

frame	n	m	o	g	w
1	),597414	0,000000	0,000000	0,000000	0,000000
2	),597464	0,000000	0,000000	0,000000	0,000000
3	),597310	0,000000	0,000000	0,000000	0,000000
4	),597000	0,000000	0,000000	0,000000	0,000000
5	),597004	0,000000	0,000000	0,000000	0,000000
6	),596553	0,000000	0,000000	0,000000	0,000000
7	),596747	0,000000	0,000000	0,000000	0,000000
8	),596149	0,000000	0,000000	0,000000	0,000000
9	),595134	0,000000	0,000000	0,000000	0,000000
10	),589131	0,000000	0,000000	0,000000	0,000000
11	),563000	0,000548	0,000000	0,000000	0,000000
12	),001010	0,910223	0,000000	0,000000	0,000000
13	),000751	0,934905	0,000000	0,000000	0,000000
14	),000306	0,969465	0,000003	0,000000	0,000000
15	),000267	0,981066	0,000003	0,000000	0,000000

Fig. 2. Time-dependent behaviors of ξ_u^a (t) at the fifth frames of 5 different candidate phonemes and the result values as a result of the dynamic process using SVNN with A=3.0, B=3.5, D=1.5, w=2.5, h=0.5

4 Experiments and Discussion

The Japanese phoneme recognition based on SVNN was conducted, which was also compared with the performance of HMM speech recognizer with a structure of a single mixture and three states. For training standard models, fist of all, each of recognition systems used two kinds of the phoneme-labeled training database. The labeled phonemes were extracted from ATR Japanese word speech database which was composed of 4000 words spoken by 10 male speakers, and from ASJ Japanese continuous speech database which was composed of 500 sentences by 6 male speakers. For evaluation, test data consisted of two kinds, one from database of 216 words set and the other from 240 words set, each of which is the phoneme balanced ATR database spoken by 3 male speakers, respectively.

For acoustic feature parameters, 10 dimensional Mel-frequency Cepstrum coefficients(MFCC) and their derivatives were used. The input data is compared with the corresponding part of the Gaussian probability density functions and a similarity map is then obtained for a dynamic process of SVNN.

The speaker independent recognition accuracies based on SVNN, which were compared with HMM, are shown in figure 3 and 4 respectively. When using SVNN, the average recognition accuracies of 3 speakers were 78.0% and 78.9% for 216 and 240 test sets, which were compared with 71.6% and 72.4% by HMM, respectively. As a result, performance based on SVNN was 6.6% higher in average.

Therefore, it was found in the results that SVNN outperformed the existing HMM recognizer. However, as shown in phonemes 'P', 'R' 'U' (figure 3) for example, the accuracies based on SVNN do not show always better performance in every phoneme than HMM. In order to reduce the error rate in performance, the first thing to be considered is an exact modeling of the inner change of phonetic features. Since this study is restricted to phoneme recognition, in addition, we should make further experiments to word or continuous speech recognition as future works.

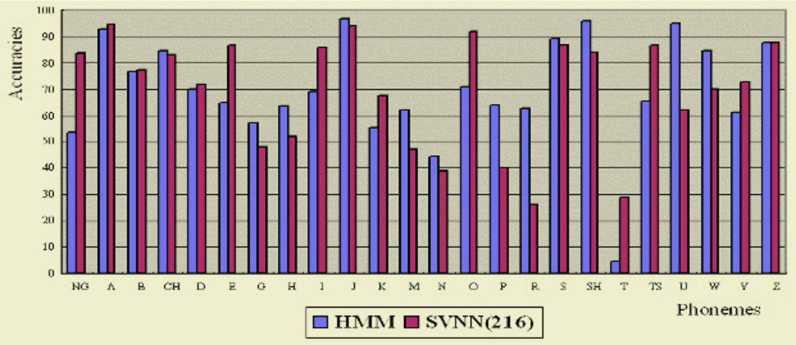


Fig.3. Comparison of SVNN with HMM on 216 test sets

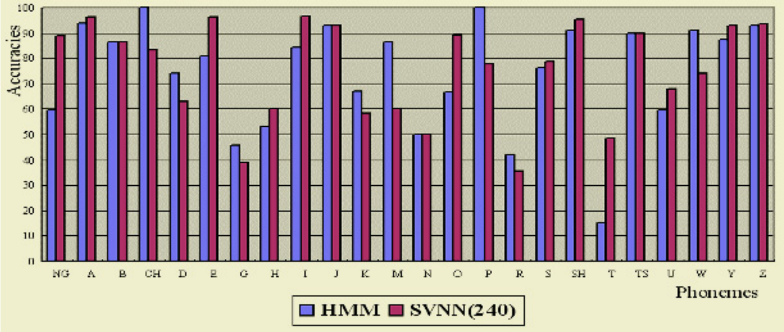


Fig.4. Comparison of SVNN with HMM on 240 test sets

5 Conclusion

This study focuses on enhancing the discriminative capability in detecting the most likely candidate out of confused sounds. The proposed system using SVNN were proved to be successful in performing them in this respect. Particularly, it was revealed that the mechanism of dynamic process for stereoscopic vision, which played a crucial role in selecting the best candidate as winner neuron, might be compatible with the underlying principles of identifying different sounds. In addition, we could see that the totally new types of the neural networks for speech recognition were able to yield much simpler architecture than the other ordinary artificial neural networks. From the experimental results, moreover, it was shown that the proposed approach with the unique characteristics in recognizing speech had better performance than the existing HMM recognizer.

References

1. Woodland, P.C., Leggestter, C.J., Odell, J.J., et. al.: The 1994 HTK Large Vocabulary Speech Recognition System. Proc. IEEE Int. Con. on Acoustics, Speech, and Signal Processing, **1** (1995) 73–76
2. Lee, K.F., Hon, H.W.: Speaker-Independent Phone Recognition Using Hidden Markov Models. IEEE Tran. on Acoustic, Speech and Signal Processing, **37** (1989) 1641–1648
3. Bourlard, H., Wellekens, C.J.: Links between Markov Models and Multi-layer Perceptrons. IEEE Tran. Patt. Anal. Machine Intell., **12** (1990) 1167–1178
4. Lang, J., Waibel, A., Hinton, G.E.: A Time-Delay Neural Network Architecture for Isolated Word Recognition. Artificial Neural Networks, Paradigms, Applications and Hardware Implementations, IEEE press, New York (1992) 388–408
5. Martinelli, G.: Hidden Control Neural Network. IEEE Tran. on Circuits and Systems, Analog and Signal Processing, 41 (1994) 245–247
6. Reinmann, D., Haken, H.: Stereo Vision by Self-organization. Biol. Cybern., **71** (1994) 17–26
7. Amari, S., Arbib, M.A.: Competition and Cooperation in Neural Nets. Systems Neuroscience, Academic Press (1977) 119–165
8. Yoshitomi, Y., Kanda, T., Kitazoe, T.: Neural Nets Pattern Recognition Equation for Stereo Vision. Trans. IPS (1998) 29–38
9. Yoshitomi, Y., Kitazoe, T., Tomiyama, J., Tatebe, Y.: Sequential Stereo Vision and Phase Transition. Proc. Int. Sym. on Artificial Life and Robotics (1998) 318–323

Speech Recognition of Finite Words Based on Multi-weight Neural Network

Yan Wu[1], Hongbo Wang[1], Mingxi Jin[1], and Shoujue Wang[2]

[1] Dept. of Computer Science and Engineering, Tongji University
1239# Siping Road, Shanghai 200092, China
yanwu@mail.tongji.edu.cn
{axlear,jinmingxi}@hotmail.com
[2] Lab of Artificial Neural Networks, Institute of Semiconductors, CAS
Beijing 100083, China
wsjue@red.semi.ac.cn

Abstract. Under the guide of the novel biomimetics pattern recognition theory that is based on the character of human's recognition, combining the character of traditional neural network, a new multi-weight neural network is constructed to realize the idea of that theory. Extraordinary results are obtained with the first use of the new multi-weight neural network in speech recognition of finite words. The experiment results show that the multi-weight neural network can not only recognize finite words correctly and promptly, but also keep quite high correct recognition rate under the circumstance of small-number samples.

1 Introduction

As the most natural and convenient means of communication, speech includes various information. People tried to process human speech signals through computer in order to make machines recognize speech in the way used by human from 1950s. The successful application of Hidden Marcov Model (HMM) [1] in speech recognition in 1980s has become a milestone in speech recognition field. Some systems such as ViaVoice of IBM and Xunfei of USTC (University of Science and Technology of China) have been used in commercial application. Meanwhile, the use of HMM model in speech recognition has a lot of shortcomings. HMM is a static model based on probability and it pays too much dependence on training samples, needs large quantity of calculation, and lacks of dynamic. In order to make up for these shortcomings, some scientists have made lots of improvements on it [2]. At the same time, HMM classifies speech signals into one or the other, which belongs to the traditional pattern recognition [3] and has great difference from the way human to recognize objects. The biomimetics pattern recognition theory [3] simulates the way that human cognize objects, that is "cognition of all sample classes one by one" rather than "the classification of many kinds of samples".

Although the development of artificial neural network [4] has encountered some difficulties, it is widely used to solve various practical problems because of its excellence in nonlinear approximation, learning capability, robust and dynamic performance. From the mathematical point of view, a single neuron can be regarded as a hy-

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 339–344, 2005.

per plane or a hyper curve in the high dimensional space and numbers of neurons form the crosses of different hyper planes or hyper curves, which divide the high dimensional space into different areas. In point of fact, these areas stand for projections of a certain kind of objects in high dimensional space. The basic theory of biomimetics pattern recognition shows that different kinds of objects can be projected to different areas and not be influenced by each other because of the infinity of the high dimensional space. On the other hand, an input of a traditional neuron corresponds to a single weight, which is the key factor to determine the quality of neural network, so a single-weight neural network limits its performance in some sense.

With a novel angle of view, this paper creates a kind of multi-weight neural network that differs from traditional neural network and owns great capability of covering space. It can realize the "cognition" ability described in the biomimetics pattern recognition theory, and has been firstly successfully applied in speech recognition of finite words. It can preferably construct the covering area of speech signal distribution in the feature space and obtain ideal experiment results.

2 Biomimetics Pattern Recognition

Different from traditional pattern recognition, biomimetics pattern recognition takes the best coverage of a kind of samples as target [3] rather than the best partition of different kinds of samples in the feature space. Fig. 1 [3] explains this theory in 2-D space.

In Fig. 1, the triangles represent samples to be recognized, the little circles and crosses represent samples to be distinguished from triangles. Polygonal lines represent the classification manner of the traditional Back Propagation (BP) network, the large circles represent the classification manner of Radial Basis Function (RBF) network, and the slight ellipses represent the "cognition" manner of the biomimetics pattern recognition.

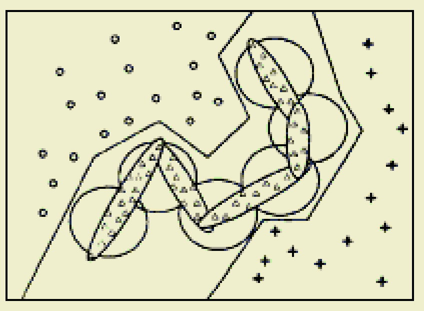


Fig. 1. Schematic diagram of bionic pattern recognition in 2-D space

The biomimetics pattern recognition adopts the recognition method of "covering the high dimensional geometrical distribution", which means to "cognize" certain kinds of samples by analyzing the relationships of training samples and coving the distribution of certain samples in feature space.

3 Multi-weight Neuron

The theory of biomimetics pattern recognition can be realized by multi-weight neural network. A multi-weight neuron can be generally expressed as:

$$Y = f[\Phi(W_1, W_2, \cdots, W_m, X)] . \tag{1}$$

in this formula, $W_1, W_2, \cdots, W_m$ are m weight vectors, X is an input vector, Y is an output vector, Φ is a calculation function decided by multi-weight neuron (input a number of vectors and output a scalar), f is a nonlinear transfer function.

Given the feature space is an n-dimensional real number space R^n , then the calculation function is

$$\Phi(W_1, W_2, \cdots, W_m, X) = k . \tag{2}$$

where k is a constant that could be regarded as a track of vector X in the feature space R^n decided by m weight vectors $W_1, W_2, \cdots, W_m$. This track divides R^n into two parts, in one part, $\Phi < k$, in the other, $\Phi \geq k$.

If formula (2) denotes a close hyper geometrical object, a finite covering area is formed in the feature space.

4 Speech Recognition Based on Multi-weight Neural Network

In speech recognition[5], the feature vectors gained from speech pre-processing are projected into the high dimensional feature space as sample points. The distribution of several sample points from a certain pattern constructs a kind of covering, which represents the distribution status of this pattern in the feature space to some extent. In this paper, the covering of speech in high dimensional space is completed by a hyper triangle constructed with samples of one word, as shown in Fig. 2(b). Namely, a hyper triangle was constructed with 3 samples in the beginning stage of training. The covering area was expanded when a new coming sample was out of the primary area. The trained multi weight neural network estimates whether a sample is in the covering area by calculating the distance between the sample and the covering area.

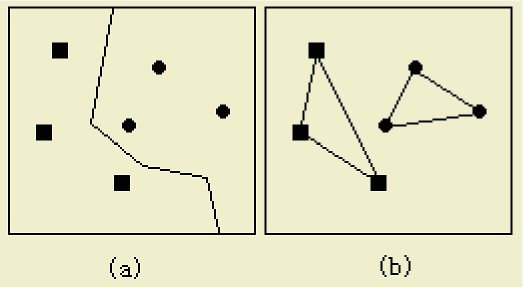


Fig. 2. Partition and covering of speech signal samples

Meanwhile, Fig. 2 shows the difference of "partition" and "cognition". The little circles and the little squares represent two different kinds of speech pattern. Fig. 2(a) denotes partition by HMM, in which the whole speech space is carved into two parts; Fig. 2(b) denotes cognition by multi-weight neural network, and the samples from the same pattern construct a hyper triangle by training process to form a certain covering to finish "cognition".

5 Experiments of Finite Words

All the speech data used in the experiments come from the speech library of the institute of semiconductor and information technology of Tongji University and are recoded in lab conditions. Each sample of speech is a continuous string of 4 digits such as "1359" with the sampling rate of 8000Hz. The library includes 20 persons, and each of them spoke every speech string once with a normal accent. There are 10 different strings, and 200 samples in all. The pre-emphasis coefficient is 0.95; the frame length is 20ms; and the frame overlap is 1/3 of the frame length. In addition, Hamming window is used. The feature parameters are MFCC with 16 elements, and are warped into 512-dimension feature vectors.

We have done 3 experiments in this paper, all of which adopt the multi-weight neural network with one input corresponding to two weights. In experiment I, the ratio of the training samples and the testing samples fixes in 50%, namely each kind of string uses 10 samples to train and another 10 to test. The purpose of experiment I is to test the performance of neural network with different threshold values. In the testing stage, the confirmation of the test sample to its same kind of string is called "recognition"; otherwise called "leakage". Meanwhile, the mistaking test experiment is also done with the samples of other kinds of string. If the samples that belong to other kinds are recognized as belonging to this kind, it is called "mistaking". The results of experiment II are shown in Table 1.

Table 1. Results with the same training set and different threshold values

Threshold	Recognition rate %	Mistaking rate %	Leakage rate %
20	85.3	0.0	14.7
50	93.7	8.5	6.3
100	100.0	17.4	0.0
150	100.0	28.0	0.0

In experiment II, the threshold value of the multi-weight neural network that could make the recognition rate and mistaking rate balanceable is fixed in advance. The training sample number of each kind of string increases from 6 to 10, by which the experiment II test the working efficiency of the multi-weight neural network with the fixed threshold value and the changeable training sample sets. The results of experiment II are shown in Table 2.

As a contrast, experiment III sent the same feature parameters into BP network, RBF network and HMM to be trained and tested. HMM took HTK[6] from Cambridge University as experiment tool, and its initialization followed the optimization rule; BP and RBF were both carried on in the Matlab environment with three-level network structure. The results are shown in Table 3.

Table 2. Results with the same threshold value and different training sets

Training set	Recognition rate %	Mistaking rate %	Leakage rate %
6	79.2	9.8	20.8
7	81.4	9.1	18.6
8	83.3	8.7	16.7
9	87.6	8.4	12.4
10	89.1	8.3	10.9

Table 3. Results of HMM, BP, RBF and Multi-weight neural network

Training set	HMM Recognition rate %	BP Recognition rate %	RBF Recognition rate %	Multi-weight neural network Recognition rate %
6	55.2	51.2	54.6	78.4
7	61.7	60.1	62.1	82.7
8	65.2	60.9	63.7	83.5
9	71.8	66.4	70.6	86.6
10	76.5	70.3	73.4	88.2

It can be seen from Table 1 that the larger the threshold value is set, the higher the recognition rate is. But with the increasing of the recognition rate, the mistaking rate also becomes higher. It means that, the covering area formed by multi-weight neural network with large threshold value can not only comprise the samples belonging to the same pattern, but also include samples belonging to other patterns because of its large scale. In order to reach the balance between recognition and mistaking, it is necessary to choose the most fitful threshold value. So, experiment II firstly calculated the threshold value that could balance the recognition rate and the mistaking rate, then trained and tested the multi-weight neural network with this threshold value. Table 2 shows that the recognition rate of multi-weight neural network increases with the addition of training samples and the leakage rate decreases with this fitful threshold value. Yet the mistaking rate can remain in a relative steady status.

Table 3 shows the comparison of multi-weight neural network method with classic HMM algorithm, and the sophisticated supervised learning models – BP and RBF. It can be drawn from the results that, in the case of small-number training samples, the recognition rate of multi-weight neural network is about 15% higher than that of HMM. The less the training samples are, the differences between the two methods are farther. In the worst condition, namely only 6 samples are used for training, the recognition rate of multi-weight neural network is 78.4%, while HMM can only arrive at 55.2% with the gap of more than 20%. It indicates that a certain covering in the high dimensional space formed by multi-weight neural network with fewer training samples can well express the speech signal itself. That is the main advantage over HMM, which needs lots of data for calculation to reach the best model.

The recognition rates of BP and RBF networks are lower than the other two methods, because the partition area of single weight neuron is relatively simple whereas the distribution of samples in the feature space is much more complicated. The multi-weight neuron can depict the concept of hyper geometrical objects in high dimensional space to make the objects belong to the same pattern closer and separates the objects of different patterns. However, the single weight neuron just separates objects in plane and has no such advantages. It also proves the validity of multi-weight neural network in covering complicated distribution in high dimensional space.

6 Conclusions

It is a novel way to apply the multi-weight neural network to speech recognition, and this paper just did some primary work. Two weights neural network is used in experiments by taking into consideration of the complexity of multi-weight neural network. If three or more weights are adopted, the influence of multi-weight neural network on speech recognition will be much greater. Besides, to reduce calculation load and time, the feature parameters are uniformly compressed into 512 dimensions, which consequentially lose some information of speech. The result would be better by taking more dimensions of feature parameters. Also, it is worthy to study the dynamic characteristic of speech signal, and the addition of time-delayed part before multi-weight neural network will be the next research direction.

References

1. Juang, R.: Fundamentals of Speech Recognition. Prentice-Hall (1983)
2. Hagen, A., Morris, A.: Recent Advances in the Multi-stream HMM/ANN Hybrid Approach to Noise Robust ASR. Computer Speech & Language, **19** (2005) 3-30
3. Wang, S.: Bionic (Topological) Pattern Recognition Theory and its Applications. Chinese Journal of Electronics, **30** (2002) 1417-1420
4. Wang, X, Wang, H., Wang, W.: Artificial Neural Network Principle and Application. NEU Press, Shenyang (2000) 123-125
5. Huang, Acero, Hon: Spoken Language Processing. Prentice-Hall (2001)
6. http://htk.eng.cam.ac.uk/sponsors.shtml

Continuous Speech Research Based on Two-Weight Neural Network

Wenming Cao[1], Xiaoxia Pan[1], and Shoujue Wang[2]

[1] Institute of Intelligent Information System, Information College
Zhejiang University of Technology, Hangzhou, Zhejiang 310032, China
csann@zjut.edu.cn
[2] Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China

Abstract. Two-weight neural network (TWNN) is described in this paper. A new dynamic searching algorithm based on Two-weight neural network is presented. And then it is applied to recognize the Continuous Speech of Speaker-Independent. The recognition results can be searched dynamically without endpoint detecting and segmenting. Different feature-space covers are constructed according to different classes of syllables. Compared with the conventional HMM-based method, The trend of recognition results shows that the difference of recognition rates between these two methods decreases as the number of training increases, but the recognition rate of Two-weight neural network is always higher than that of HMM-based. And both of these recognition rates will reach 100% if there are enough training samples.

1 Introduction

During the last two decades, many approaches have been proposed for recognition. Hidden Markov models (HMMs) (Yang, 1988[1], Chen et al., 1987[2], Lee, 1997[3]) have been applied to recognize in languages, such as Mandarin, Cantonese and Thai. For isolated tone recognition, very high recognition accuracy has been obtained (Yang, 1988[1]). However, for speech recognition in continuous speech, although relatively high tone recognition accuracy has been achieved in Chen and Wang, 1995[4] and Zhang, 2000[6] for Mandarin and Thai, respectively, manual segmentation was done before training the tone models, which is not suitable for automatic speech recognition (ASR). Without phonological constraints, lower recognition scores of 66.4% Lee et al., 2002a [5]have been reported for Mandarin and Cantonese, respectively. Two-weight neural network[5,6.7,8,9] is described in this paper. A new dynamic searching algorithm based on Two-weight neural network is presented. And then it is applied to recognize the Continuous Speech of Speaker-Independent. The recognition results can be searched dynamically without endpoint detecting and segmenting. Different feature-space covers are constructed according to different classes of syllables. Compared with the conventional HMM-based method, The trend of recognition results shows that the difference of recognition rates between these two methods decreases as the number of training increases, but the recognition rate of Two-weight neural network is always higher than that of HMM-based. And both of these recognition rates will reach 100% if there are enough training samples.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 345–350, 2005.

2 The Two-Weight Neural Network

The Two weights neuron's typical feature is that it not only has the direction value "W" of feed forward network, but also has kernel value "Z". Such Two weights neuron is a high rank neuron that has good character of partial reaction and is anisotropic. The Two weights neuron's basal formula is:

$$Y = f\left[\sum_{j=0}^{M}\left(\frac{W_j(X_j - W_j^{'})}{\left|W_j(X_j - W_j^{'})\right|}\right)^S \left|W_j(X_j - W_j^{'})\right|^P - \theta\right] \tag{1}$$

Y is the output of neuron. f is neuron's excitation function. W_j is the direction value that is from the NO. j input port to neuron. $W_j^{'}$ is kernel value that is from the NO. j input port to neuron. X_j is the NO. j input port (positive value). M is input spatial dimension. S is the parameter that decides monomial sign method.

3 The Learning Algorithm of Constructing the Two-Weight Neural Network

Step 1: Suppose a type of number's all sample points' aggregation is $\alpha = \{A_1, A_2, \cdots, A_N\}$, N is the quantity of sample points. Figure out the distance among these points. Find out the shortest distance between two points. Record them as B_1 and B_2. Calculate the sum of the distance form other points to these two points. Let the distance be shortest. Record the point which is not collinear with B_{11}, B_{12} as B_{13}. It composes the first plane triangle-$B_{11}B_{12}B_{13}$. Record it as θ_1. Also use a Two weights neuron to cover. The covering range is:

$$P_2 = \{X \mid \rho_{X\theta_2} \le Th, X \in R^n\}$$

$$\theta_2 = \{Y \mid Y = \alpha_2[\alpha_{1.}B_{21} + (1-\alpha_1)B_{22}] + (1-\alpha_2)B_{23}, \alpha_{1.} \in [0,1], \alpha_2 \in [0,1]\}$$

$\rho_{X\theta_2}$ is the distance between the point X and the space θ_1.

Step 2: To the former constructed geometrical structure P_1, judge whether the surplus points are included by the structure. If they are in the covering range of the structure, remove them. To the sample point outside the structure, accord the method of step 1. Find out the point B_{21} whose distance to the distance's sum of $B_{11}B_{12}B_{13}$ is shortest. Find out the two points whose distance to B_{21} are shortest. Record them as $B_{22}B_{23}$. The $B_{22}B_{23}$ and B_{21} compose the second plane triangle $B_{21}B_{22}B_{23}$. Record it as θ_2. Also cover it with the Two weights neuron. Its covering range is:

$$P_2 = \left\{X \mid \rho_{X\theta_2} \le Th, X \in R^n\right\}$$

$$\theta_2 = \{Y \mid Y = \alpha_2[\alpha_{1.}B_{21} + (1-\alpha_1)B_{22}] + (1-\alpha_2)B_{23}, \alpha_{1.} \in [0,1], \alpha_2 \in [0,1]\}$$

$\rho_{X\theta_2}$ is the distance between the point X and the space θ_2 .

Step 3: Exclude the sample points that are included in the former $(i-1)$ Two weights neuron's covering volume in the surplus points. In the points outside the covering volume, find out the point whose distance to the former $(i-1)$ th triangle's vertexes' distance's sum is shortest. Record it as B_{i1}. Record the $(i-1)$ th triangle's two vertexes whose distance to the point are shortest as B_{i2} B_{i3} to compose the i th plane triangle $B_{i1}B_{i2}B_{i3}$. Record it as θ_3 . Also cover it with the Two weights neuron. Its covering range is:

$$P_i = \left\{X \mid \rho_{X\theta_2} \le Th, X \in R^n\right\}$$

$$\theta_3 = \{Y \mid Y = \alpha_2[\alpha_{1.}B_{i1} + (1-\alpha_1)B_{i2}] + (1-\alpha_2)B_{i3}, \alpha_{1.} \in [0,1], \alpha_2 \in [0,1]\}$$

Step 4: Repeat step 3 until having dealt with all sample points.

Finally the quantity of created Two weights neurons will be m . Each class number's covering range is union of these neurons' covering area:

$$P = \bigcup_{i=1}^{m} P_i \quad .$$

When to recognize, give the Th =0, $pSi3$ neuron's expression as:

$$\rho = \quad \left\| X - \theta_{(W_1,W_2,W_3)} \right\|$$

The neuron's output ρ is the distance from the point X to the finite region $\theta_{(w_1,w_2,w_3)}$. The method to calculate the distance is approximation algorithm.

The distance between the sample X waiting for being recognized and the Two-weight neural network's covering range covered by the i th class number's high dimension space points is

$$\rho_i \quad = \min_{j=1}^{M_i} \rho_{\text{ij}} \quad i = 1, \cdots, 11$$

M_i is the number of neural network's $pSi3$ neuron covered by the i th class number's high dimension space points. ρ_{ij} is the distance between the sample X waiting for being recognized and the neural network's j th neuron's covering range covered by the i th class number's high dimension space points. Let the sort which includes the shortest distance between the sample X waiting for being recognized and the Two weights neuron covered by that class number's high dimension space points as the digital class including the sample X . The discriminance is:

$$j = \arg\min_{i=1}^{11} \rho_i, \quad j \in (1, \cdots, 11)$$

4 Experiment and Analysis

This library includes 24 persons' 2640 MFCC samples of single digital syllable. They are divided into 5 groups (the table 1) according to different sample figures.

Table 1. The HMM state figures and hybrid figures in the gauss probability density function

The sample points' quantity of each class		24	48	96	120	168	240
HMM	State figures	4	4	5	5	5	5
	Hybrid figures in the gauss probability density function	2	4	3	6	6	6

In the Chinese continuous digital speech recognition, we need to consider enough the influence of cooperating pronunciation among figures. The influence of a figure's former one to it is crucial. And the latter one's influence to the figure is much lighter. When choosing every class's digital samples of each group, we review the position of every single digital syllable's modeling sample in 8 bit continuous numeric strand and the situation of its former figure. Let every chosen class's digital sample of each group include the situation that every figure connects with its former one. For example, the sample of number 0 in the first group chooses 11 classes' numbers that are those 0s appeared before the 0, and let the situations that the 11 classes' numbers influence the 0 appear equably. This sample set includes bigger extent and more situations.

There are 29 persons' totally 7308 single digital syllable samples used to test the correctness of modeling.

This paper stresses reviewing HMM's modeling ability to digital syllable's acoustical layer. Whether the modeling ability to digital syllable's acoustical layer is good or not will directly influence the total modeling recognizing result. We tested HMM's modeling ability of acoustical layer with the differently quantitative scavenged single digital syllable samples. Because the single digital syllable samples of this paper are scavenged from the continuous digital speech, it includes the feature of continuous speech. So the isolated word's syllable speech has essential difference.

Construct each group's single digital syllable samples' HMM model. Use the HMM model that is from left to right without jumping. How many of hybrid figures in the gauss probability density function has huge influence on the recognition. Adjust the state figures and hybrid figures in the gauss probability density function to let the test set get the recognition rate as high as possible. Through the repeating tests, each groups' states and hybrid figures in the gauss probability density function are displayed in Figs. 1-2 and total recognition rate comparison when using different size of dataset to model.

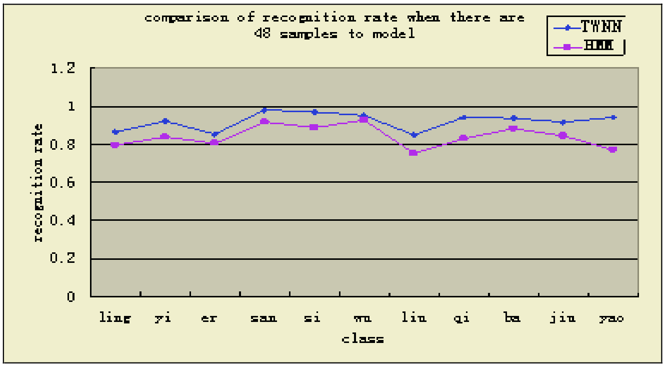


Fig. 1. Recognition rate comparison when there are 48 samples to model

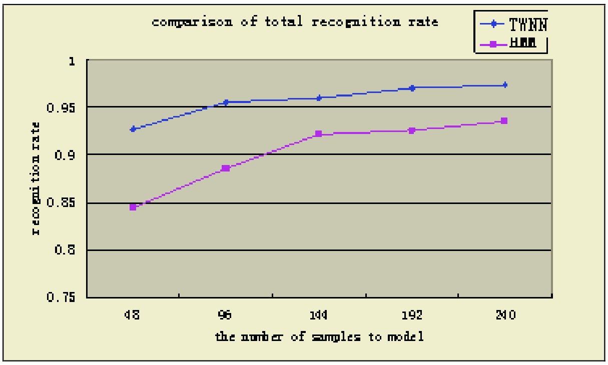


Fig. 2. Total recognition rate comparison when using different size of dataset to model

We can see that the difference between these two methods will be decreased with the number of samples increasing. But the recognition rate of the TWNN method will always be higher than HMM's.

5 Conclusion

Because dividing continuous speech is difficult, and it also directly influences the speech recognition rate. Some experiments that have been researched indicate that if we correct the previous division mistakes, the error rate of system's words would decrease at least 11.7%. So we can say that, to a great extent, the other systems' low recognition rates to continuous speech closely relate to the low accurate rates of the detection of continuous speech endpoint. So this paper changes the traditional speech recognition's pattern that is to be syncopated firstly then to be recognized. It uses the algorithm with dynamic searching. It achieves the continuous speech recognition without syncopating. The paper is from the complicated two weights' geometrical structure. It gives a new algorithm in speech with Two-weight neural network's learning algorithm. We hope it can be used in continuous speech of large vocabulary.

References

1. Yang, W.-J.: Hidden Markov Model for Mandarin Lexical Tone Recognition. IEEE Trans. Acoust. Speech Signal Process, **36** (1988) 988–992
2. Chen, X.-X., Cai, C.-N., Guo, P., Sun, Y.A.: Hidden Markov Model Applied to Chinese Four-tone Recognition. Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP) (1987) 797–800
3. Lee, L.S.: Voice Dictation of Mandarin Chinese. IEEE Signal Processing, **14** (1997) (4) 63–101.
4. Chen, S.H., and Wang, Y.R.: Tone Recognition of Continuous Mandarin Speech Based on Neural Networks. IEEE Trans. Speech Audio Process, **3** (1995) 146–150
5. Wang, S.: Biomimetic (Topological) Pattern Recognition - A New Model of Pattern Recognition Theory and Its Applications (in Chinese). Acta Electronica Sinica, **30** (2002) 1417-1420
6. Wang, S.: Biomimetics Pattern Recognition. Neural Networks Society (INNS, ENNS, JNNS) Newsletter, **1** (2003) 3-5
7. Cao, W., Feng, H.,Wang, S., The Application of DBF Neural Networks for Object Recognition. Inf. Sci., **160** (2004) 153-160
8. Cao, W.: The Application of Direction Basis Function Neural Networks to the Prediction of Chaotic Time Series. Chinese Journal of Electronics, **13** (2004) 395-398
9. Cao, W., Feng, H., Zhang. D.M., et al.: An Adaptive Controller for a Class of Nonlinear System Using Direction Basis Function. Chinese Journal Of Electronics, **11** (2002) 303-306

Two-Domain Feature Compensation for Robust Speech Recognition

Haifeng Shen[1], Gang Liu[1], Jun Guo[1], and Qunxia Li[2]

[1] Beijing University of Posts and Telecommunications, 100876, Beijing, China
shen_hai_feng@126.com, lg@pris.edu.cn, guojun@bupt.edu.cn
[2] University of Science and Technology Beijing, 100083, Beijing, China
kellylqx@163.com

Abstract. In this paper, we develop a two-domain feature compensation approach to the log-filterbank and log-energy features for reducing the effects of noise. The environment model is approximated by statistical linear approximation (SLA) function. The cepstral and log-energy feature vectors of the clean speech are trained by using the Self-Organizing Map (SOM) neural network with the assumption that the speech can be well represented as multivariate diagonal Gaussian mixtures model (GMM). With the effective training of clean speech and environment model approximation, noise statistics is well estimated using batch-EM algorithm in a maximum likelihood (ML) sense. Experiments in the large vocabulary speaker-independent continuous speech recognition demonstrate that this approach exhibits a noticeable performance.

1 Introduction

Recently, varieties of robust compensation approach based on environment model approximation in the feature space were proposed to deal with environment corruption, e.g. VTS (Vector Taylor Series) [1] and SLA (Statistical Linear Approximation) [2]. VTS and SLA apply different strategies to approximate the non-linearized environment model and iteratively estimate the noise statistics using batch-EM algorithm. Because SLA considers the information of higher order statistics of environment model, it can get more accuracy than VTS can. These approaches general compensate the mel-scaled log-filterbank feature or the cepstral feature. Actually, the log-energy feature is an important feature which can greatly improve the speech recognition. In the paper, we develop a new SLA based two-domain approach which compensates both the mel-scaled log-fiterbank feature and log-energy feature. Because the cepstral features are nearly incorrelate, it is more effective to cluster the clean speech into Gaussian mixtures model with diagonal covariance matrices in the cepstral domain than in the log-filterbank domain. Therefore, we train the two-domain features of the clean speech including the 24 dimensional cepstral features and one dimensional log-energy feature by using the Self-Organizing Map (SOM) neural network. With the reasonable clean speech modeling and environment model approximation, the noisy speech obtains effective compensation.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 351–356, 2005.

The paper is organized as follows. The next section briefly describes environment model approximation in the log-filterbank and log-energy domains using SLA. As a special case, we only consider the additive environment model. In section 3, clean speech feature vectors are clustered by SOM neural network and noise statistics are estimated using batch-EM algorithm. The experiment results are given in section 4 and some conclusions are drawn in section 5.

2 Environment Model Approximation Using SLA

In the log mel-scaled spectral domain, the corrupted log-filterbank feature vector can be expressed as [1]

$$y = x + \log(1+\exp(n-x)) = f(x,n) \ . \tag{1}$$

Suppose that the clean speech is independent to the additive noise, the energy of each noisy speech frame can be represented as

$$E_y = E_x + E_n \ . \tag{2}$$

Hence, if transforming to the log-energy domain, the log energy corruption can also be written as the form of Eq. (1). $f(x,n)$ is a nonlinear function. SLA approximates it with a linearized function $g(x,n)$ defined as [2]

$$g(x,n) = A(x-x_0) + B(n-n_0) + C \ , \tag{3}$$

where $\{A,B,C\}$ are constant matrices which relate to the vector points $\{x_0,n_0\}$ and can be solved under the MMSE criterion.

Therefore, the environment model can be simplified to

$$y = \tilde{A}x + \tilde{B}n + \tilde{C} \ , \tag{4}$$

in which $\tilde{A} = A_m, \tilde{B} = B_m, \tilde{C} = C_m - A_m x_0 - B_m n_0$, the subscript "$m$" denotes the m-order SLA approximation.

3 Environment Parameters Estimation

3.1 Distribution Assumption

Assuming that the clean speech can be modeled as Gaussian mixtures

$$p(x) = \sum_{j=1}^{M} p_j N(x, \mu_{xj}, \Sigma_{xj}) \ , \tag{5}$$

where M denotes the total number of mixture components, p_j, μ_{xj} and Σ_{xj} denote the mixture gain, the mean vector and the diagonal covariance matrix for the j th mixture component, respectively. In a similar way, we also assume that each noise vector n

can be represented as a single Gaussian distribution $N(n,\mu_n,\Sigma_n)$. Correspondingly, the noisy speech can be modeled as Gaussian mixtures with the following statistics

$$\begin{cases}\mu_{yj}=\tilde{A}_j\mu_{xj}+\tilde{B}_j\mu_n+\tilde{C}_j \\ \Sigma_{yj}=\tilde{A}_j\Sigma_{xj}\tilde{A}_j^T+\tilde{B}_j\Sigma_n\tilde{B}_j^T\end{cases} j=1,2,\cdots,M \ . \tag{6}$$

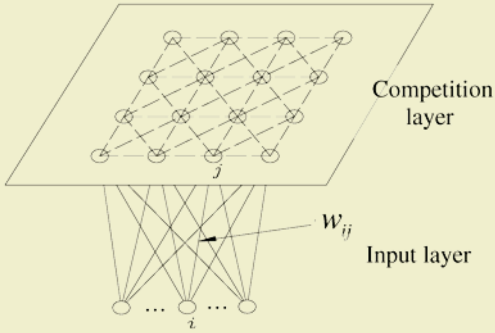


Fig. 1. SOM Network Structure

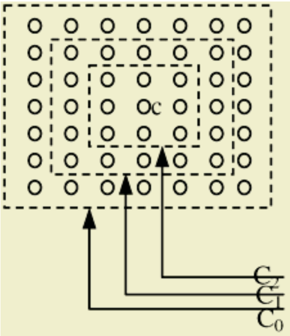


Fig. 2. Topological neighborhood

3.2 Training GMM of the Clean Speech Using SOM

In Fig.1, SOM developed by Kohonen is an unsupervised clustering network [3], which has two layers structure: the input layer and the competition layer. Each input neuron i is connected with each output neuron j with the synaptic weight w_{ij}. In Fig.2, for each representative c, we define three neighborhoods $C(k), k=0,1,2$. Each has a different learning rate η_k. The value η_k will be greater if the neighborhood is closer to the representative c. If output neuron c wins when d-dimensional input vector $x(n)=\{x_1(n),\cdots,x_d(n)\}^T$ and the weight $w_j(n)=\{w_{1j}(n),\cdots,w_{dj}(n)\}^T$ satisfies [3]

$$\left\|x(n)-w_c(n)\right\|=\min\left\|x(n)-w_j(n)\right\| \ . \tag{7}$$

Then update the synaptic weight vector associated with the representative neuron c, the learning rate is given by

$$w_j(n+1)=\begin{cases}w_j(n)+\eta_k(n)(x(n)-w_j(n)), j\in C(k) \\ w_j(n), j\notin C(k)\end{cases} \ . \tag{8}$$

3.3 Noise Statistics Estimation

Suppose that the statistics parameters of clean feature vector are known, given the noisy speech observation sequence $Y=\{y_1,y_2,\cdots,y_T\}$, we use batch-EM algorithm to estimate the noise statistics. In the first step, namely Expectation step (E-step), an auxiliary function is defined as [4] [5]

$$Q(\bar{\lambda} \mid \lambda) = E\{\log P(X, N, J, \bar{\lambda}) \mid Y, \lambda\} \ . \tag{9}$$

In the second step, called Maximization step (M-step), we find the parameter $\bar{\lambda} = \{\mu_n, \Sigma_n\}$ that maximizes the auxiliary function $Q(\bar{\lambda} \mid \lambda)$, i.e. $\bar{\lambda} = \arg\max_{\bar{\lambda}} Q(\bar{\lambda} \mid \lambda)$. Computing the derivatives of $Q(\bar{\lambda} \mid \lambda)$ with respect to $\bar{\lambda} = \{\mu_n, \Sigma_n\}$ and setting the derivatives to zero, we obtain the noise statistics estimate as

$$\begin{cases} \hat{\mu}_n = \dfrac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{M} p(j_t = j \mid y_t, \lambda)\mu_{n_t \mid y_t, j_t = j, \lambda} \\ \hat{\Sigma}_n = \dfrac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{M} p(j_t = j \mid y_t, \lambda)\mu_{n_t^2 \mid y_t, j_t = j, \lambda} - \hat{\mu}_n^2 \end{cases} . \tag{10}$$

An alternative to estimate noise statistics is given below, the auxiliary function can be defined as [1]

$$Q(\bar{\lambda} \mid \lambda) = E[L(Y, J \mid \bar{\lambda}) \mid Y, \lambda] \ , \tag{11}$$

where $J = \{j_1, j_2, \cdots, j_T\}$ is mixture gain sequence of clean feature sequence. The noise mean vector also can be estimated as follows

$$\mu_n = \frac{\sum_{t=1}^{T}\sum_{j=1}^{M} p(j \mid y_t, \lambda)\tilde{B}_j^T \Sigma_{yj}^{-1}[y_t - (\tilde{A}_j \mu_{xj} + \tilde{C}_j)]}{\sum_{t=1}^{T}\sum_{j=1}^{M} p(j \mid y_t, \lambda)\tilde{B}_j^T \Sigma_{yj}^{-1}\tilde{B}_j} \ . \tag{12}$$

From the above auxiliary function, the covariance matrix can't be estimated. So the noise covariance matrix can be estimated in the absence of speech in the observation sequence.

4 Experiments

A continuous HMM-based speech recognition system is used in the recognition experiments for examining the presented approach. The utterances of 82 speakers (41 males and 41 females) taken from the mandarin Chinese corpus provided by the 863 plan (China High-Tech Development Plan) [6] are used to train, and those of 9 speakers are used to test. Two-domain compensation is taken in the mel-scaled log-filterbank and log-energy domain. We use 39 dimensional features consisting of 12 cepstral coefficients and log energy feature with the corresponding delta and acceleration coefficients. A five-state structure of a left-to-right HMM model including self-loop transitions is trained for any of the 61-sub-syllable. Then triphone-based HMM models are used in this continuous speech recognition.

The white and babble noise from NoiseX92 [7] are added to the test set by varying the signal-to-noise ratio (SNR) from 0dB to 20dB. The performance of the presented approach is evaluated by the comparisons to the baseline with no compensation and Cepstral Mean Normalization (CMN). The speech recognition rate was computed as $((N - D - S)/N) \times 100$ where N, D, S are the total number of words in the reference transcriptions, substitution errors and deletion errors, respectively. In Table1 and Table 2, SLA-1, SLA-2 and SLA-3 denote compensation with different order SLA environment model approximation. SLA-1 is equivalent to VTS environment model approximation proposed by Moreno [1].Tests with white and babble noise in different SNR condition indicate that the SLA-based approach is better than the rest approaches in the sense of speech recognition accuracy. Compared to the baseline and CMN, in the white noisy environments, SLA-1 averagely improves 19.55% and 10.99%, SLA-2 averagely improves 19.71% and 11.15%, SLA-3 averagely improves 19.54% and 10.98%, respectively. In the babble noisy environments, SLA-1 averagely improves 9.52% and 4.66%, SLA-2 averagely improves 9.81% and 4.95%, SLA-3 averagely improves 9.80% and 4.93%, respectively.

Table 1. Recognition rates (%) for white noise

SNR	0dB	5dB	10dB	15dB	20dB
Baseline	0.32	2.54	11.14	30.00	56.04
CMN	3.31	10.98	29.99	36.94	61.65
SLA-1	5.30	16.61	37.36	61.34	77.19
SLA-2	5.66	16.74	36.84	61.83	77.53
SLA-3	5.37	16.40	36.78	61.83	77.37

Table 2. Recognition rates (%) for babble noise

SNR	0dB	5dB	10dB	15dB	20dB
Baseline	3.87	24.61	54.84	62.98	80.23
CMN	11.16	32.38	55.95	71.04	80.31
SLA-1	16.32	39.29	61.89	75.48	81.14
SLA-2	17.41	39.67	62.39	75.26	80.84
SLA-3	17.48	39.78	62.27	75.26	80.72

Among these SLA-based approaches, the recognition performance of SLA-3 is equivalent to or better than SLA-1 in the white or babble noisy environments and SLA-2 obtains the best recognition accuracy. Due to the additive white noise is the stationary noise and babble noise is slowly time-varying noise, it is sufficient to deal with them by applying SLA-2 compensation. To the time-varying noise, especially highly non-stationary noise, higher order statistical linear approximation will be suitable to describe the environment contamination procedure.

5 Conclusions

In the paper, we present a two-domain feature compensation approach. Environment model is simplified to the linearized function using SLA environment approximation. The cepstral and log-energy features of the clean speech are trained using SOM neural network. Due to a reasonable clustering, we obtain the effective Gaussian mixtures model in the log-filterbank and log-energy domain using inverse DCT transform and its transpose. In all situations it is found that the presented approach performs better than the baseline and CMN. In addition, in the stationary noisy environment, the performances of the second SLA environment approximation are better than those of the other order SLA environment approximation approach. For unknown highly time-varying environment, we must use high order statistics environment estimation for effectively describing the environment contamination procedure. As the future work, the higher order SLA environment approximation to estimate the highly non-stationary noise is expected.

Acknowledgement

This research was sponsored by NSFC (National Natural Science Foundation of China) under Grant No.60475007 and the Foundation of China Education Ministry for Century Spanning Talent (02029).

References

1. Moreno, P.J.: Speech Recognition in Noisy Environments (Ph. D. Dissertation). ECE Department, CMU. (1996)
2. Kim, N.S.: Statistical Linear Approximation for Environment Compensation. IEEE Signal Processing Letters, **5** (1998)
3. Kohonen, T.: The Self-organizing Map. Proceedings of the IEEE, **78** (1990) 1464-1480
4. Rose, R.C., Hofstetter, E.M., Reyholds D.A.: Integrated Models of Signal and Background with Application to Speaker Identification in Noise. IEEE Transactions on Speech and Audio Processings, 2 (1994) 245-257
5. Kim, D.Y., Un ,C.K., Kim, N.S.: Speech Recognition in Noisy Environments Using First-order Vector Taylor Series. Speech Communication, 24 (1998) 39-49
6. Zu, Y.Q.: Issues in the Scientific Design of the Continuous Speech Database. Available: http://www.cass.net.cn/chinese/s18_yys/yuyin/report/report_1998.htm
7. Varga, A., Steenneken, H.J.M., Tomilson, M., Jones,D.: The NOISEX–92 Study on the Effect of Additive Noise on Automatic Speech Recognition. Documentation on the NOISEX–92 CD–ROMs, (1992)

On Kernel Discriminant Analyses Applied to Phoneme Classification

András Kocsor

Research Group on Artificial Intelligence
Hungarian Academy of Sciences and University of Szeged
H-6720 Szeged, Aradi vértanúk tere 1., Hungary
kocsor@inf.u-szeged.hu

Abstract. In this paper we recall two kernel methods for discriminant analysis. The first one is the kernel counterpart of the ubiquitous Linear Discriminant Analysis (Kernel-LDA), while the second one is a method we named Kernel Springy Discriminant Analysis (Kernel-SDA). It seeks to separate classes just as Kernel-LDA does, but by means of defining attractive and repulsive forces. First we give technical details about these methods and then we employ them on phoneme classification tasks. We demonstrate that the application of kernel functions significantly improves the recognition accuracy.

1 Motivation

In the last two decades the dominant method for speech recognition has been the hidden Markov modeling approach [12]. In the meantime, the theory of machine learning has developed considerably and now has a wide variety of classification algorithms for pattern recognition problems [4]. One such development is the "kernel-idea", which has become a key notion in machine learning [2], [5], [13].

The primary goal of this paper is to show alternative methods for phoneme classification using state of the art kernel discriminant analyses. We describe here both the well-know kernel version of Linear Discriminant Analysis [1], [7], [9] and the Kernel Springy Discriminant Analysis, which we first proposed in [8].

2 Kernel Discriminant Analyses

Without loss of generality we shall assume that as a realization of multivariate random variables, there are n-dimensional real attribute vectors in a compact set $\mathcal{X}$ over $\mathbb{R}^n$ describing objects in a certain domain, and that we have a finite $n \times k$ sample matrix $X = [\mathbf{x}_1, \ldots, \mathbf{x}_k]$ containing k random observations. Let us assume as well that we have r classes and an indicator function $\mathcal{L} : \{1, \ldots, k\} \rightarrow \{1, \ldots, r\}$, where $\mathcal{L}(i)$ gives the class label of the sample $\mathbf{x}_i$. Let k_j further denote the number of vectors associated with label j in the sample data. Now let the dot product be implicitly defined by the kernel function $\kappa : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ in some finite or infinite dimensional dot product space $\mathcal{F}$ with associated mapping $\phi : \mathcal{X} \rightarrow \mathcal{F}$ such that

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 357–362, 2005.

$$\forall \mathbf{x}, \mathbf{z} \in \mathcal{X} \quad \kappa(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{z}). \tag{1}$$

Usually ϕ is the feature map and $\mathcal{F}$ is the kernel feature space. This space and dot product calculations over it are defined only implicitly via the kernel function itself. The space $\mathcal{F}$ and map ϕ may not be explicitly known. We need only define the kernel function, which then ensures an implicit evaluation over $\mathcal{F}$. The construction of kernels, when such a mapping ϕ exists, is a non-trivial problem. Based on Mercer's theorem we can use continuous, symmetric and positive definite functions as kernels [2], [13].

The goal of discriminant analyses is to find a mapping $h : \mathcal{X} \to \mathcal{Y}$ which leads to a new set of features that are optimal according to a given class separation criterion. In the case of kernel discriminant analysis the mapping is nonlinear and has the following form: $\mathbf{z} \to AF^T\phi(\mathbf{z})$, $(\mathbf{z} \in \mathcal{X})$, where A is a method dependent, real $m \times k$ matrix and $F = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_k)]$ is called the image matrix of the sample. We should note here that $K = F^T F$ is the so-called kernel matrix and, $F^T\phi(\mathbf{z})$ can be calculated implicitly via the kernel function, i.e. $F^T\phi(\mathbf{z}) = [\kappa(\mathbf{x}_1, \mathbf{z}), \ldots, \kappa(\mathbf{x}_k, \mathbf{z})]^\top$.

2.1 Kernel-LDA

The 'kernelized‘ counterpart of Linear Discriminant Analysis, the Kernel-LDA defines the row vectors of matrix A by the stationary points of the following Rayleigh-quotient[7]:

$$\tau(\boldsymbol{a}) = \frac{(F\boldsymbol{a})^\top \mathcal{B}(F\boldsymbol{a})}{(F\boldsymbol{a})^\top \mathcal{W}(F\boldsymbol{a})}, \qquad F\boldsymbol{a} \in \mathcal{F}, \tag{2}$$

where $\mathcal{B}$ is the *Between-class*, while $\mathcal{W}$ is the *Within-class Scatter Matrix* of the images of the sample over ϕ. Here the *Between-class Scatter Matrix* $\mathcal{B}$ shows the scatter of the class mean vectors $\boldsymbol{\mu}_j$ around the overall mean vector $\boldsymbol{\mu}$:

$$\begin{aligned} \mathcal{B} &= \textstyle\sum_{j=1}^{r} \frac{k_j}{k}(\boldsymbol{\mu}_j - \boldsymbol{\mu})(\boldsymbol{\mu}_j - \boldsymbol{\mu})^\top \\ \boldsymbol{\mu} &= \textstyle\frac{1}{k}\sum_{i=1}^{k} \phi(\mathbf{x}_i) \\ \boldsymbol{\mu}_j &= \textstyle\frac{1}{k_j}\sum_{\mathcal{L}(i)=j} \phi(\mathbf{x}_i) \end{aligned} \tag{3}$$

The *Within-class Scatter Matrix* $\mathcal{W}$ represents the weighted average scatter of the covariance matrices $\mathcal{C}_j$ of the vectors with the class label j:

$$\begin{aligned} \mathcal{W} &= \textstyle\sum_{j=1}^{r} \frac{k_j}{k}\mathcal{C}_j \\ \mathcal{C}_j &= \textstyle\frac{1}{k_j}\sum_{\mathcal{L}(i)=j}(\phi(\mathbf{x}_i) - \boldsymbol{\mu}_j)(\phi(\mathbf{x}_i) - \boldsymbol{\mu}_j)^\top \end{aligned} \tag{4}$$

After some algebraic rearrangement, Eq. (2) takes the following form:

$$\tau(\boldsymbol{a}) = \frac{\boldsymbol{a}^\top K(R - \hat{I})K\boldsymbol{a}}{\boldsymbol{a}^\top K(I - R)K\boldsymbol{a}}, \qquad F\boldsymbol{a} \in \mathcal{F}, \tag{5}$$

where K is the kernel matrix, $[\hat{I}]_{ij} = 1/k$ and

$$[R]_{ij} = \begin{cases} \frac{1}{k_t} & \text{if } \ t = \mathcal{L}(i) = \mathcal{L}(j) \\ 0 & \textit{otherwise.} \end{cases} \tag{6}$$

This means that Eq. (2) can be expressed in terms of dot products of $\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_k)$ and that the stationary points of this quotient can be computed by solving the generalized eigenvector problem $K(R - \hat{I})K\boldsymbol{a} = \lambda(K(I - R)K)\boldsymbol{a}$. To define the transformation matrix A of Kernel-LDA we use only those eigenvectors which correspond to the m dominant real eigenvalues.

2.2 Kernel-SDA

Kernel Springy Discriminant Analysis (Kernel-SDA) [8] was invented with goals very similar to those of Kernel-LDA. The name Kernel Springy Discriminant Analysis stems from the utilization of a spring & antispring model, which involves searching for directions with optimal potential energy using attractive and repulsive forces. In our case sample pairs in each class are connected by springs, while those of different classes are connected by antisprings. New features can be easily extracted by taking the projection of a new point in those directions having a small spread in each class, while different classes are spaced out as much as possible.

Now let the dot product again be implicitly defined by the kernel function κ in some finite or infinite dimensional feature space $\mathcal{F}$ with associated transformation ϕ such that $\kappa(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{z})$ for all $\mathbf{x}, \mathbf{z}$. Further, let $\delta(F\boldsymbol{a})$ the potential of the spring model along the direction $F\boldsymbol{a}$ in $\mathcal{F}$, be defined by

$$\delta(F\boldsymbol{a}) = \sum_{i,j=1}^{k} ((\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j))^\top F\boldsymbol{a})^2 [M]_{ij}, \quad F\boldsymbol{a} \in \mathcal{F} \tag{7}$$

where

$$[M]_{ij} = \begin{cases} -1, \text{ if } \mathcal{L}(i) = \mathcal{L}(j) \\ \ \ 1, \text{ otherwise} \end{cases} \qquad i, j = 1, \ldots, k. \tag{8}$$

Naturally, the elements of matrix M may be initialized with values different from ±1 as well. The elements can be considered as a kind of force constant and any pair of data points can have different force constant values. Similar to Kernel-LDA, Kernel-SDA defines the row vectors of matrix A by the stationary points of a Rayleigh-quotient, which in this case has the following form:

$$\tau(\boldsymbol{a}) = \frac{\delta(F\boldsymbol{a})}{(F\boldsymbol{a})^\top (F\boldsymbol{a})} = \frac{(F\boldsymbol{a})^\top \mathcal{D} (F\boldsymbol{a})}{(F\boldsymbol{a})^\top (F\boldsymbol{a})}, \tag{9}$$

where

$$\mathcal{D} = \sum_{i,j=1}^{k} \left(\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\right) \left(\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\right)^\top [M]_{ij}. \tag{10}$$

Technically speaking, with the above τ definition Kernel-SDA searches for those directions $F\boldsymbol{a} \in \mathcal{F}$ along which a large potential is obtained. It is straightforward to see that the Rayleigh quotient for Kernel-SDA has the form:

$$\tau(\boldsymbol{a}) = 2 \frac{\boldsymbol{a}^\top K(\tilde{M} - M)K^\top \boldsymbol{a}}{\boldsymbol{a}^\top K \boldsymbol{a}}, \tag{11}$$

where K is the kernel matrix and $\tilde{M}$ is a diagonal matrix with the sum of each row of M in the diagonal. Eq. (11) means that Eq. (9) can be expressed as a function of dot products of $\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_k)$. Now the stationary points of $\tau(\boldsymbol{a})$ can be obtained via an eigenanalysis of the following generalized eigenproblem: $(K(\tilde{M} - M)K^\top)\boldsymbol{a} = \lambda K\boldsymbol{a}$. To define the transformation matrix A we use the dominant m eigenvectors.

3 Phoneme Classification Results

Now we proceed with a description of the experiments. In this section we investigate the effect of the previous methods applied prior to classification in the phoneme classification task.

Evaluation Domain. The classification techniques combined with discriminant analyses as feature space transformation methods were compared using a corpus which consists of several speakers pronouncing Hungarian numbers. 77% of the speakers were used for training and 23% for testing. The ratio of male and female talkers was 50%-50% in both the training and testing sets. The recordings were made using a commercial microphone in a reasonably quiet environment, at a sample rate of 22050Hz. The whole corpus was manually segmented and labeled. Since some of these labels represented only allophonic variations of the same phoneme, some labels were fused, and so we actually worked with a set of 28 labels. We made tests as well with two other groupings where the labels were grouped into 11 and 5 classes, based on phonetic similarity. Hence we had three phonetic groupings, which henceforth will be denoted by *grp1*, *grp2* and *grp3*.

Initial Features. Before feature extraction the energy of each word was normalized. After this the signals were processed in 512-point frames (23.2 ms), where the frames overlapped by a factor of 3/4. A Fast Fourier Transform was applied on each frame. After that 24 critical band energies and 16 mel-frequency cepstral coefficients were calculated. Besides the above ones we also wanted to do experiments with some more phonetically based features like formants. We used gravity centers in 4 frequency bands as a crude approximation for formants. Doing this we got $24 + 16 + 4$ features altogether for each frame. Afterwards, for each feature we took the average of the frame-based measurements for the first quarter, the central part and the last quarter of the phoneme, which led to $44 \times 3 = 132$ features for each phoneme. By adding the duration of each phoneme to this set we finally got a feature set consisting of 133 elements.

Feature Space Transformation. After initial feature extraction we applied feature space transformation methods, hoping for a better classification. Besides LDA and SDA we also employed Principal Component Analysis (PCA), which served as a baseline method for comparison. In the case of PCA, SDA, Kernel-SDA the original feature space was reduced to 32 dimensions, while in the case of LDA and Kernel-LDA the number of features kept was always the number of classes minus one.

Learning Methods. We employed five well-known classification algorithms during the tests. TiMBL [3] is a Memory Based Learner which means a new example is evaluated based on consuming the previous examples stored in the memory. C4.5 [11] is based on the well-known ID3 tree learning algorithm. The OC1 (Oblique Classifier 1) algorithm [10] learns by creating *oblique* decision trees. The fourth, Artificial Neural Networks [4], is a conventional pattern recognition tool. In the experiments we employed the most common feed-forward multilayer perceptron network with the backpropagation learning rule. *Gaussian Mixture Model* (GMM) [4] is a well-known discriminative learning method. It assumes that the class-conditional probability distribution can be well approximated by a convex sum of multidimensional normal distributions.

Experimental Results and Evaluation. The same experiments were carried out on the three phoneme groupings *grp1*, *grp2*, *grp3*, all the learning methods being tested not just on each set but with each transformation technique. Table 1 depicts the recognition accuracies for *grp1*, *grp2* and *grp3*, respectively. The columns show the five feature transformation methods and the rows correspond to the classification algorithms applied. The maximum is shown in bold.

On examining the classifiers the first thing we notice is that the general preference of the methods on the phoneme classification task is the following: C4.5$\prec$OC1$\prec$GMM$\prec$TiMBL$\prec$ANN. As regards the feature space transformation methods we realized that the base-line PCA method was outperformed by LDA (cf. [6]) and SDA, which was in turn surpassed by their kernel versions (Kernel-LDA and Kernel-SDA). Based on these observations we summarize the efficiency relations of the methods as follows: PCA$\prec$(LDA$\approx$SDA)$\prec$(Kernel-LDA$\approx$Kernel-SDA). Another thing we realized was that the efficiency of Kernel-SDA improved when the number of classes decreased. We also noticed that Kernel-SDA considerably helps the efficiency of C4.5, which for instance resulted in the best accuracy value for the *grp*3 recognition problem.

Table 1. Recognition accuracies for the classifier-transformation combinations

phoneme groupings	classifier	PCA	LDA	SDA	K-LDA	K-SDA
Grp1 (28 classes)	TIMBL	75.23	83.33	80.49	89.32	88.07
	C4.5	56.20	67.80	66.90	83.80	78.90
	OC1	60.17	70.86	68.91	75.55	78.56
	ANN	84.46	86.94	83.22	**90.86**	87.76
	GMM	74.82	86.23	79.85	89.20	82.94
Grp2 (11 classes)	TIMBL	82.74	86.11	84.21	93.96	93.96
	C4.5	69.30	83.00	79.60	88.58	92.50
	OC1	74.41	85.22	81.56	91.54	91.54
	ANN	90.60	89.78	89.18	92.84	**94.73**
	GMM	80.91	87.12	84.04	91.89	92.13
Grp3 (5 classes)	TIMBL	88.17	90.95	90.36	95.62	95.62
	C4.5	79.10	92.50	90.90	94.30	**96.60**
	OC1	86.88	92.67	88.83	95.39	92.84
	ANN	93.09	93.09	92.61	94.68	95.80
	GMM	90.13	92.26	89.24	93.61	89.06

4 Conclusions

This paper sought to study the effects of some kernel discriminant analyses on phoneme classification, a basic task of speech recognition. After inspecting the test results we can confidently say that it is worth experimenting with these methods in order to obtain better classification results. The use of non-linearity brought further improvements on the classification accuracy. Still, we should note that the goals of feature space transformation and learning are practically the same. That is, if we have a very efficient learner then there is almost no need for a feature space transformation. Put the other way round, a proper transformation of the feature space may make the data so easily separable that quite simple learners will suffice. These are, of course, extreme examples.

References

1. Baudat, G., and Anouar, F.: Generalized Discriminant Analysis Using a Kernel Approach, Neural Comput., **12** (2000) 2385-2404
2. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and other kernel-based learning methods, Cambridge University Press (2000)
3. Daelemans, W., Zavrel, J., Sloot, K., Bosch, A.: TiMBL: Tilburg Memory Based Learner version 2.0 Reference Guide, ILK Technical Report - ILK 99-01, Computational Linguistics, Tilburg University, The Netherlands (1999)
4. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, Wiley, NY (2001)
5. Kernel Machines Web site: http://www.kernel-machines.org
6. Kocsor, A. et al.: A Comparative Study of Several Feature Transformation and Learning Methods for Phoneme Classification, Int. Journal of Speech Technology, **3** (2000) 263-276
7. Kocsor, A., Tóth, L., Paczolay, D.: A Nonlinearized Discriminant Analysis and Its Application to Speech Impediment Therapy, In V. Matousek et al. (eds.): Proc. of TSD 2001, Springer Verlag LNAI Series, **2166** (2001) 249-257
8. Kocsor, A., Kovács, K.: Kernel Springy Discriminant Analysis and Its Application to a Phonological Awareness Teaching System, in P. Sojka et al. (eds.): Proc. of TSD 2002, Springer Verlag LNAI Series, **2448** (2002) 325-328
9. Mika, S., Rätsch, G., Weston, J., Schölkopf, B., Müller, K.-R.: Fisher Discriminant Analysis with Kernels, In: Y.-H. Hu et al. (eds.): Neural Networks for Signal Processing IX, IEEE (1999) 41-48
10. Murthy, S. K., Kasif, S., Salzberg, S.: A System for Induction of Oblique Decision Trees, Journal of Artificial Intelligence Research, **2** (1994) 1-32
11. Quinlan, J. R.: C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo California (1993)
12. Rabiner, L., Juang, B.H.: Fundamentals of Speech Recognition, Prent. Hall (1993)
13. Vapnik, V. N.: Statistical Learning Theory, John Wiley & Sons Inc (1998)

Automatic News Audio Classification Based on Selective Ensemble SVMs*

Bing Han, Xinbo Gao, and Hongbing Ji

School of Electronic Engineering, Xidian Univ., Xi'an, Shaanxi 710071, China
{hanbing,xbgao}@lab202.xidian.edu.cn,
hbji@xidian.edu.cn

Abstract. With the rapid growing amount of multimedia, content-based information retrieval has become more and more important. As a significant clue for video indexing and retrieval, audio detection and classification attracts much more attention and becomes a hot topic. On the basis of the priori model of news video structure, a selective ensemble support vector machines (SE-SVMs) is proposed to detect and classify the news audio into 4 types, i.e., silence, music, speech, and speech with music background. Experiments with news audio clips of 8514 seconds in total length illustrate that the average accuracy rate of the proposed audio classification method reaches to 98.9%, which is much better than that of the available SVM-based or traditional threshold-based method.

1 Introduction

The task of automatic segmentation, indexing, and retrieval of multimedia data has important applications in professional media production, audiovisual archive management, education, entertainment, surveillance, and so on. A vast amount of audiovisual material has been archived in television and film databases. If these data can be properly segmented and indexed, it will facilitate the retrieval of desired video segments for the editing of a documentary or an advertisement video clip.

Audio classification has been an active research area in recent years, and various audio feature extraction and classification schemes have been proposed for automatic audio and video semantic segmentation. Kimber and Wilcox applied hidden Markov models (HMMs) to classify audio signals into music, silence and speech using cepstral features [1]. In addition, Scheirer and Slaney [2] examined 4 different classification frameworks including multidimensional Gaussian maximum a posteriori estimator, Gaussian mixture model, a spatial partitioning scheme based on *k-d* trees, and a NN classifier for speech/music discrimination. However, the above methods have either large computational burden or low classification accuracy.

To balance the effectiveness and efficiency of the audio classifier, support vector machines are used for the classification and segmentation of audio streams. Lu *et al.* employed SVMs in their work [3], which hierarchically classified audio signals into five classes. Yet this hierarchical classification scheme has a drawback: if the signal

* This work was supported by National Natural Science Foundation of China (No.60202004) and the Key Project of Chinese Ministry of Education (No.104173).

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 363–368, 2005.

is misclassified in an earlier stage, it will never reach the correct type (leaf node). In addition, all of the features are used to classify at each stage in their works, so the computation complexity is very high. During the experiments, we find that not all the features are useful for classifying to certain type audio. For this purpose, a selective ensemble SVMs method is proposed in this paper. Several binary SVM classifiers are trained in parallel using each feature. Then, via the selective ensemble, the decision rules are obtained, and the final classification results can be further achieved. Experimental results show that the proposed method is both effective and robust.

2 Selective Ensemble Support Vector Machines

Support vector machine (SVM) is a promising technique for classification and regression developed by Vapnik V.N. et al. [4]. It has been proved to be competitive with the best available learning machines in many applications [5]. Now, this paper presents the Selective Ensemble SVMs to classify the news audio.

In the beginning of the 1990s, Hansen and Salamon [6] showed that the generalization ability of an artificial neural network system can be significantly improved through ensembling artificial neural networks, *i.e.*, training several artificial neural networks and combining their predictions. Much work has been done in designing ensemble approaches. Maclin and Shavlik [7] utilized competitive learning to generate individual network and then combined their outputs via simple averaging.

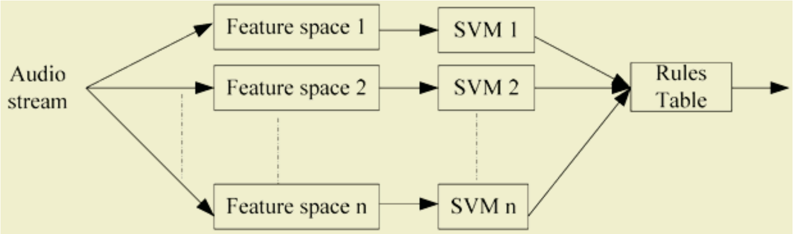


Fig. 1. Selective ensemble support vector machines

The SVM-based method available uses several binary SVMs classifiers to classify different audios [3], in which there are some unnecessary features for different audio clips and the computation complexity is much great. Therefore, the selective ensemble SVM-based method is proposed in this paper. Several different SVMs are ensembled to classify the given audio signals due to different audios can be captured by different features.

Firstly, training set is composed of several demonstrations, each of which uses different feature subspaces. So, the output results cannot be compared directly. In this way, several SVMs are selected, the number of which is equal to feature numbers. Each SVMs is trained on the corresponding feature subspace, which is shown in Figure 1. To obtain the final classification result, we also build a decision table.

3 Audio Feature Selection

For our case, the given audio is first uniformly segmented into non-overlapping clips with 1-second length (Sampling frequency is 44100Hz). And it is further divided into forty 25ms-length frames with 50% overlapping. Currently, 20 audio features are considered in this work. A brief description of several key features is given as follows. Readers can refer to [8],[9] for more detailed discussion.

Short-Time Energy Function (STE): it provides a convenient representation of the amplitude variation over time. Here, the mean and variance of short-time energy are used as the first feature.

Short-Time Average Zero-Crossing Rate (ZCR): for discrete-time signals, a zero-crossing is said to occur if successive samples have different signs. The mean and variance of ZCR are selected as the second feature. In general, speech signals are composed of alternating voiced sounds and unvoiced sounds in the syllable rate, while music signals do not conform to this kind of structure. For the speech with music background, the ZCR changes continually.

Cepstrum Coefficient of Linear Predictive (LPC): it is useful for discriminating speech and non-speech. This coefficient captures the statistic features of signal. In our method, 4-order LPC is used.

Differential Coefficient of LPC (d-LPC)*:* it shows the dynamic feature of the signal, which is calculated by the difference between the two adjacent frames.

Frequency Center (FC): it has been found that FC is related to the human sensation of the brightness of a sound we hear. Generally, the music FC is often stable, while speech FC changes continually.

Bandwidth (BW): it measures the range of frequency. Commonly, the frequency of speech signal ranges from 300Hz to 3.4 kHz while that of music is about 20.05 kHz.

Energy Ratios of Sub-Bands (ERSB): considering the perceptual property of human ears, the entire frequency band is divided into four sub-bands. The energy ratio of sub-band *i* is the ratio of its energy to the total energy of frame STE. So, the ERSB of music is much larger than that of speech. Both mean and variance of ERSB are used as the seventh feature.

Mel-frequency Cepstral Coefficients (MFCCs): we use it to classify the speech sound from other types of audio and further to discriminate the man speech and the women speech. 12-order coefficient is adapted in our experiments.

Differential Coefficient of MFCCs (d-MFCCs): it shows the dynamic feature of the signal, which is calculated by the difference between two adjacent frames.

Wavelet Coefficient (WT-Coefficient): it is obtained by wavelet transform. Here, we use the mean and variance of high frequency component as features because the distribution of music is uniform while that of speech is non-uniform.

4 The Proposed Method

According to the structure of news, there are mainly 4 types of audio in the news, *i.e.*, *silence*, *music*, *speech* and *speech with music background* (S-M). The classification procedure of the above 4 types of audios is shown in Figure 2.

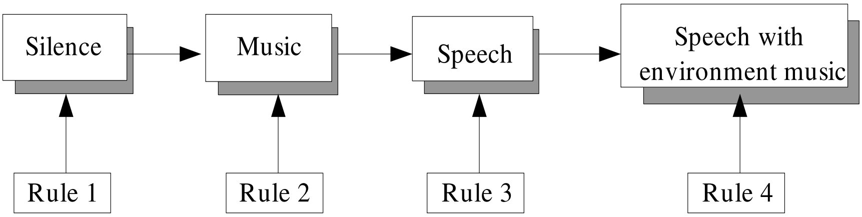


Fig. 2. The procedure of our method

The classification rule-bank is given in Table 1, in which +1 and -1 denote the positive and negative classification results of the SVMs, respectively, and * denotes that the output of SVMs does not influence on the result.

Table 1. The table of audio features

	Silence	Music	Speech	S-M
STE	+1	*	*	*
ZCR	+1	+1	-1	+1
LPC	*	-1	+1	-1
d-LPC	*	-1	+1	-1
FC	*	+1	+1	+1
BW	*	+1	+1	-1
ERSB	*	+1	+1	+1
MFCC	*	*	+1	*
d-MFCC	*	*	+1	*
WT-Coefficient	*	+1	-1	-1

From Table 1, the selective feature decision can be obtained as follows.

Rule 1: $ab \rightarrow Silence$;

Rule 2: $b\bar{c}\bar{d}efgj \rightarrow Music$;

Rule 3: $\bar{b}cdefghi\bar{j} \rightarrow Speech$

Rule 4: $b\bar{c}\bar{d}e\bar{f}g\bar{j} \rightarrow Speech\ with\ environment\ music$;

Obviously, through the selective ensemble, partial feature set can be used to achieve the same decision with the whole features. Therefore the computation complexity is decreased by attribute reduction.

5 Experimental Results

The database used in our experiments is composed of 340,560 audio clips, 8,514 seconds in total length, collected from CCTV-1 news. Moreover, to find the optimal SVM parameters, we also have manually labeled approximately 25-min of speech, 7-min of silence, 8-min of music and 13-min of environmental sound as ground truth. Based on the validation results, we choose the radial basis function (RBF) as kernel and set parameters $\tilde{a}$ to 4 and C to 10. Approximately 99.0% and 98.8% accuracies have been achieved on the training and validation data, respectively.

Table 2 shows the classification results of the 4 sound types in the form of a confusion matrix, where names in the leftmost column represent the actual classes while the ones in the top line are the classes predicted by the classification scheme.

Table 2. SE-SVMs based classification results of four sound types

Sound type	Silence	Music	Speech	S-M
Silence	6968	0	0	0
Music	0	3164	140	20
Speech	0	18	100388	20
S-M	0	40	152	820
Precision	100%	98.8%	99.78%	95.16%

In table 2, there are two false detections in the speech with music background. One is that the speech with music background is classified into music, which results from the existence of breaks in the speech. The other is due to the too weak music background that it is classifier into speech.

Table 3 compares the performance of the 3 classification schemes in terms of audio classification accuracy and CPU time on the same data set. Clearly, the SE-SVM based approach has outperformed the other two for every audio class. Although the classification accuracy of the SVM-based method for speech and silence are fairly good, the accuracies for the rest sound types are rather low, especially for the speech with music background. The CPU times of the 3 methods are normalized by that of the SVMs-based approach. Obviously, the CPU time of the SVMs-based method is much higher than that of our method.

Table 3. Classification accuracy comparisons between the SVM-based [10] and the threshold-based methods

Sound type	SE-SVM	SVM-based	Threshold-based
Silence	100%	100%	96.4%
Music	99.7%	90%	79.7%
Speech	98.1%	99.41%	92.7%
Speech with music background	97.5%	90.30%	78.4%
Average precision	98.9%	94.91%	86.8%
CPU Time	0.79	1.00	0.52

Although the threshold-based method costs less CPU times, it achieves good performance only for the case of tedious fine-tunings of various thresholds for every test news video. Whereas, it is unpractical for the real applications.

6 Conclusions

This paper presents our efforts on applying the Support Vector Machine (SVM) technology to the applications of audio detection and classification. Considering the computational complexity, the selective ensemble support vector machines method (SE-SVM) is proposed for the classification with much more features. Experiments performed on five news videos with about 142-min length in total, have yielded an average 98.9% classification accuracy. The performance comparison among the proposed SE-SVMs method, the SVM-based and the threshold-based approaches demonstrate the superiority of the proposed classification scheme in both the accuracy and time consuming.

References

1. Kimber, D., Wilcox, L.: Acoustic Segmentation for Audio Browsers. Proc. of Interface Conference, Sydney, Australia (1996)
2. Scheirer, E., Slaney, M.: Construction and Evaluation of a Robust Multifeature Speech/Music Discriminator. ICASSP'97, Munich, Germany, **2** (1997) 1331-1334
3. Lu, L., Li, S. Z., Zhang, H. J.: Content-based Audio Segmentation Using Support Vector Machine, ICME, **1** (2001) 749-752
4. Vapnik, V.N.: The Nature of Statistical Learning Theory. 2nd ed. Springer-Verlag, Berlin Heidelberg, New York (1995)
5. Li, S.Z., Guo, G.D.: Content-based Audio Classification and Retrieval Using SVM Learning. Proc. of the Special Session on Multimedia Information Indexing and Retrieval. Sydney, Australia: University of Sydney (2000) 156-163
6. Hansen, L.K., Salamon, P.: Neural Network Ensembles. IEEE Trans. Pattern Analysis and Machine Intelligence, **12** (1990) 993-1001
7. Maclin, R., Shavlik, J.W.: Combining the Predictions of Multiple Classifiers: Using Competitive Learning to Initialize Neural Networks. Proc. of the 14th International Joint Conference on Artificial Intelligence, Montreal, Canada (1995) 524-530
8. Wang, Y., Liu, Z., Huang, J.: Multimedia Content Analysis Using Both Audio and Visual Cues. IEEE Signal Processing Magazine, (2000) 12-36
9. Zhang, T., Kuo, C.C.: Audio Content Analysis for Online Audiovisual Data Segmentation. IEEE Trans. on Speech and Audio Processing, **9** (2001) 441-457
10. Li, Y., Dorai, C.: SVM-based Audio Classification for Instructional Video Analysis. ICASSP, Montreal, Quebec, Canada (2004) 897-900

A Compound Statistical Model Based Radar HRRP Target Recognition[*]

Lan Du, Hongwei Liu, Zheng Bao, and Junying Zhang

National Lab. of Radar Signal Processing, Xidian University,
Xi'an, Shaanxi 710071, China
{dulan,hwliu,zhbao,jyzhang}@mail.xidian.edu.cn

Abstract. In radar HRRP based statistical target recognition, one of the most challenging tasks is how to accurately describe HRRP's statistical characteristics. Based on the scattering center model, range resolution cells are classified, in accordance with the number of predominant scatterers in each cell, into three statistical types. In order to model echoes of different types of resolution cells as the corresponding distribution forms, this paper develops a compound statistical model comprising two distribution forms, i.e. Gamma distribution and Gaussian mixture distribution, for target HRRP. Determination of the type of a resolution cell is achieved by using the rival penalized competitive learning (RPCL) algorithm. In the recognition experiments based on measured data, the proposed compound model not only has better recognition performance but also is more robust to noises than the two existing statistical models, i.e. Gaussian model and Gamma model.

1 Introduction

A high-resolution range profile (HRRP) is the amplitude of the coherent summations of the complex time returns from target scatterers in each range resolution cell, which contains the target structure signatures, such as target size, scatterer distribution, etc., thereby radar HRRP target recognition has received intensive attention from the radar automatic target recognition (RATR) community [1],[2]. However, there has been little work in the field of radar HRRP based statistical recognition, in which how to accurately describe HRRP's statistical property is a challenging task.

Based on the scattering center model [3], range resolution cells are classified, in accordance with the number of predominant scatterers in each cell, into three statistical types. Therefore, in order to model echoes of different types of resolution cells as the corresponding distribution forms, this paper develops a compound statistical model comprising two distribution forms, i.e. Gamma distribution and Gaussian mixture distribution. Determination of the type of a resolution cell is achieved by using the rival penalized competitive learning (RPCL) algorithm. Experimental results for measured data show that the proposed model not only has better recognition perform-

[*] This work was supported by the National Science Foundation of China (NO.60302009).

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 369–374, 2005.

ance but also is more robust to noises than the two existing statistical models, i.e. Gaussian model [1] and Gamma model [2].

2 Statistical Models for Target HRRP

2.1 Statistical Characteristics of Target HRRP

According to the scattering center model [3], the limitation of target-aspect change to avoid scatterers' motion through resolution cells (MTRC) is given by

$$\delta\varphi \le (\delta\varphi)_{\mathrm{MTRC}} = \frac{\Delta R}{L} \ . \tag{1}$$

where ΔR is the length of range resolution cell, and L is the cross length of target. The immovable scatterers and their distribution in a range resolution cell are irrelevant to those in another cell within $(\delta\varphi)_{MTRC}$, thus the resolution cells of an HRRP approximate to be statistically independent. Usually, there are three kinds of scatterer distribution in a range resolution cell.

a) First type of range resolution cell: There are a large number of small scatterers and no predominant scatterers in this type of range resolution cell. Under the hypothesis that the intensities of the small scatterers are almost same and the number of the small scatterers is large enough, the central limit theorem holds. If the echo phases of the small scatterers are uniform-distributed, according to the central limit theorem, the I- or Q-component of the complex echo of this type of cell will follow Gaussian distribution with zero mean and its amplitude will follow Rayleigh distribution.

b) Second type of range resolution cell: This type of range resolution cell consists of a predominant scatterer and a large number of small scatterers. Under the hypothesis similar to the first type, the I- or Q-component of the complex echo of this type of cell still follows Gaussian distribution with the mean determined by I- or Q-component of the echo of the predominant scatterer, in accordance with the central limit theorem. Thus its amplitude follows a Ricean distribution amplitude.

c) Third type of range resolution cell: There are a large number of small scatterers and several predominant scatterers in this type of range resolution cell. If a resolution cell has two predominant scatterers with their echoes' phase difference uniform-distributed, the amplitude will follow a distribution with double peaks whose values will be blunt by the echoes of the small scatterers. The echo of the third type of resolution cell usually follows a distribution with several peaks (mainly double peaks). However, relevant literatures [1],[2] didn't take such distribution into consideration. This paper will introduce the distribution with several peaks into HRRP based statistical recognition.

2.2 Statistical Description of Target HRRP

Now there exist two statistical models for target HRRP, i.e. Gaussian model [1] and Gamma model [2]. Gaussian model is based on that the power transformed echo per resolution cell approximates to be Gaussian-distributed [6]. Two problems arise when Gaussian model is used. Firstly, the power parameter used in power transformation needs to be carefully selected in the training phase, otherwise, the recognition performance will be degraded seriously; Secondly, power transformed HRRPs are more sensitive to the signal-to-noise-ratio (SNR) than original HRRPs. Gamma model is motivated by the fact that Gamma distribution has been empirically observed to model target cross-section fluctuations under the low-resolution condition [4]. However, the statistical characteristics of target HRRP are much more complex than those of point-target's echo with low resolution radar. Statistical description of the three types of echoes will be discussed in the following.

a) Under the ideal condition, the echoes of the first and second types of resolution cells are Rayleigh-distributed and Ricean-distributed, respectively. What the ideal condition requires is a large number of the scatterers with almost same intensities per resolution cell, which ensures that the central limit theorem holds. However, increasing the resolution implies a reduction of the number of scatterers per resolution cell, thus the Rayleigh distribution and Ricean distribution tend to fail for the echoes of the first and second types of cells under the high-resolution condition. More flexible distribution forms should be selected. In this paper, the first and second types of echoes are modelled with Gamma distribution. The pdf of Gamma distribution is

$$p_{\Gamma}(x)=\begin{cases}\dfrac{m}{\mu\cdot\Gamma(m)}\left(\dfrac{mx}{\mu}\right)^{m-1}\exp\left(-\dfrac{mx}{\mu}\right) & , \quad x>0 \\ 0 & , \quad x\leq 0\end{cases} . \tag{2}$$

where μ is the mean, m is the order parameter, and $\Gamma(\square)$ denotes the gamma function. The reasons why Gamma distribution is selected are as follows. i) The echo value per resolution cell is larger than zero, while a Gamma variant is also larger than zero; ii) The pdfs of Gamma family are similar to those of Rayleigh distribution and Ricean distribution, all of which have "heavy tails"; iii) Gamma distribution is a flexible distribution with two parameters, i.e. mean parameter and order parameter. By changing the two parameters, Gamma family can approximate to Exponential distribution, χ^2 distribution, Gaussian distribution and so on; iv) Compared with Ricean distribution and other complex distributions, Gamma distribution is simple, of which the two parameters can be estimated by the maximum likelihood (ML) method.

b) The echo of third type of resolution cell follows a distribution with several peaks. A mixture distribution can approximate such distribution. For the sake of simplicity, this paper selects Gaussian mixture distribution.

Thus this paper proposed that echoes of different types of cells are modeled as the corresponding distribution forms and an HRRP follows the united distribution.

2.3 Distinguishing Between the Three Types of Range Resolution Cells

According to what mentioned above, as far as the composition is concerned, the numbers of predominant scatterers in the three types of cells are different; as far as the distribution form is concerned, the first and second types of echoes follow distributions with one peak, but the third follows a distribution with several peaks. Therefore, the number of clusters of training data can reflect the type of a resolution cell, in other words, the first and second types of echoes should have one cluster, whereas the third should have several. Xu et al. in 1993 proposed the RPCL algorithm that can perform appropriate clustering without knowing the number of clusters by automatically driving extra seed points far away from the input data set [5]. Given a set of data $\{\boldsymbol{y}_i\}_{i=1}^{N}$ ($\boldsymbol{y}_i$ is a C -dimensional vector) that forms J^* clusters (J^* is unknown), the RPCL algorithm is to perform clustering by learning J seed points ($J > J^*$), denoted as $\left\{\boldsymbol{\omega}_j^{(0)}\right\}_{j=1}^{J}$. RPCL algorithm mainly consists of the following two steps.

Let $t=0$, $i=1$ and Th be a small positive number.

Step 1: Let

$$I(j\,|\,\boldsymbol{y}_i)=\begin{cases}1\ , & if\quad j=c\\ -1\ , & if\quad j=r\\ 0\ , & otherwise\end{cases}\qquad 1\le j\le J\ . \tag{3}$$

with

$$c=\arg\min_{j}\gamma_j\left\|\boldsymbol{y}_i-\boldsymbol{\omega}_j\right\|^2,\qquad r=\arg\min_{j\ne c}\gamma_j\left\|\boldsymbol{y}_i-\boldsymbol{\omega}_j\right\|^2\ . \tag{4}$$

where $\gamma_j=\frac{n_j}{\sum_{r=1}^{J}n_r}$ is the relative winning frequency of the seed point $\boldsymbol{\omega}_j$ in the past, and n_j is the cumulative number of occurrences of $I(j\,|\,\boldsymbol{y}_t)=1$ in the past;

Step 2: Update the weight vector $\left\{\boldsymbol{\omega}_j^{(t)}\right\}_{j=1}^{J}$ by

$$\boldsymbol{\omega}_j^{(t)}\,|^{new}=\boldsymbol{\omega}_j^{(t)}\,|^{old}+\Delta\boldsymbol{\omega}_j\ . \tag{5}$$

with

$$\Delta\boldsymbol{\omega}_j=\begin{cases}\alpha_c\left(\boldsymbol{y}_i-\boldsymbol{\omega}_j^{(t)}\right)\ , & if\quad I(j\,|\,\boldsymbol{y}_i)=1\\ -\alpha_r\left(\boldsymbol{y}_i-\boldsymbol{\omega}_j^{(t)}\right)\ , & if\quad I(j\,|\,\boldsymbol{y}_i)=-1\\ 0\ , & otherwise\end{cases}\qquad 1\le j\le J\ . \tag{6}$$

where $0\le\alpha_c,\alpha_r\le 1$ are the learning rates for the winner and rival unit, and often set α_r □ α_c as shown in the literature [5]. Let $i=i+1$. If $i\le N$, the algorithm returns to Step 1.

If $i>N$ and $\left\|\left[\Delta\boldsymbol{\omega}_1^T,\Delta\boldsymbol{\omega}_2^T,\cdots\Delta\boldsymbol{\omega}_J^T\right]\right\|>Th$, let $t=t+1$ and $i=0$. Then the algorithm returns to Step 1. If $i>N$ and $\left\|\left[\Delta\boldsymbol{\omega}_1^T,\Delta\boldsymbol{\omega}_2^T,\cdots\Delta\boldsymbol{\omega}_J^T\right]\right\|\le Th$, the iteration ends. According to the number of samples in each cluster, the cluster label is defined as

$$Cluster\_Index(j)=\begin{cases}1\ , & if\quad Num\{I(j\,|\,\boldsymbol{y}_i)=1\,|\,i=1,2,\cdots,N\}>0\\ 0\ , & if\quad Num\{I(j\,|\,\boldsymbol{y}_i)=1\,|\,i=1,2,\cdots,N\}=0\end{cases}\qquad 1\le j\le J\ . \tag{7}$$

Then the number of clusters after the competitive learning is

$$N_C=Num\{Cluster\_Index(j)=1\,|\,j=1,2,\cdots,J\}\ . \tag{8}$$

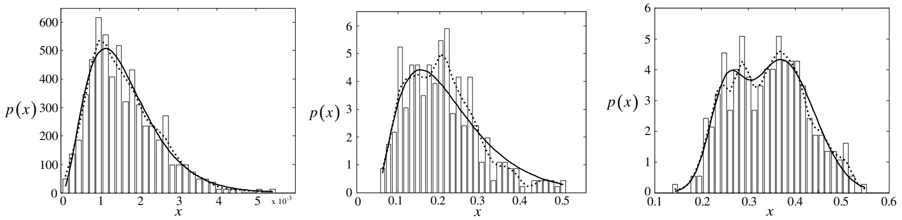
When the number of clusters per resolution cell is estimated, its type can be distinguished and its echo can be modeled as the corresponding distribution form. Thus we can build a compound statistical model to represent the statistical characteristics of target HRRP. Parameter estimation of Gamma distribution and Gaussian mixture distribution is achieved by using the ML method [8] and the expectation-maximization (EM) algorithm [7], respectively.

3 Recognition Experiments Based on Measured Data

The recognition experiments performed here are based on real airplane data. Fig. 1 shows the approximated pdfs of the echoes of some typical resolution cells, where the bar represents that approximated by histogram method, the dotted line represents that by Pazern windows using Gaussian kernels and the real line represents that by the statistical model proposed in this paper. There are two mixture components in the third type of cells in our experiment. The proportions of the third type of cells are 28.00%, 30.50% and 47.00% for “Yark-42”, “Cessna” and “An-26”, respectively. Table 1 shows a comparison in recognition performance between Gaussian model with the optimal power transformation ($\alpha^* = 0.5$), Gamma model and the proposed compound model. Obviously, our model has the largest average recognition rate. Moreover, in order to compare the effect of noises on the three models, Gaussian white noises with their powers determined by the SNR are added to the I- and Q-echo of each resolution cell, respectively. The average recognition rates of the three models versus the SNR are shown in Fig. 2. Obviously, Gaussian model with power transformation is the most sensitive to noises. With SNR=20 dB , the average recognition rates of Gaussian model, Gamma model and our compound model are 60.67%, 70.67% and 80.13%, respectively.

Table 1. Confusion matrices of Gaussian model, Gamma model and the compound model

	Gaussian model $\alpha^* = 0.5$			Gamma model			compound model		
	Yark-42	Cessna	An-26	Yark-42	Cessna	An-26	Yark-42	Cessna	An-26
Yark-42	99.50	1.05	16.05	99.25	0.30	12.25	100.00	0.80	6.40
Cessna	0	96.00	9.95	0.25	91.65	4.40	0	94.40	0.80
An-26	0.50	2.95	74.00	0.50	8.05	83.35	0	4.80	92.80
Average		89.83			91.42			95.73	



(a) the first type (b) the second type (c) the third type

Fig. 1. the approximated pdfs of the echoes of some typical resolution cells

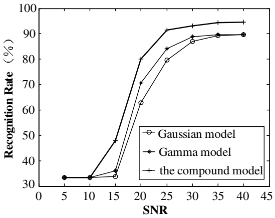


Fig. 2. The average recognition rates versus the SNR

4 Conclusions

There are two statistical models available for target HRRP, i.e. Gaussian model and Gamma model. This paper develops a compound statistical model comprising two distribution forms, i.e. Gamma distribution and Gaussian mixture distribution, by which echoes of different types of resolution cells are modeled as the corresponding distribution forms and an HRRP follows the united distribution. Determination of the type of a resolution cell is achieved by using the RPCL algorithm. Experimental results for measured data show that the proposed model not only has better recognition performance but also is more robust to noises than Gaussian model and Gamma model.

References

1. Jacobs, S. P., O'sollivan, J. A.: Automatic Target Recognition Using Sequences of High Resolution Radar Range Profiles. IEEE Trans. on AES, **36** (2000) 364-380
2. Webb, A. R.: Gamma Mixture Models for Target Recognition. Pattern Recognition, **33** (2000) 2045-2054
3. Ye, W.: Study of the Inverse Synthetic Aperture Radar Imaging and Motion Compensation. Xidian University (1996)
4. Skolnik, M.: Introduction to Radar Systems (2nd). McGraw-Hill, New York (1980)
5. Xu, L., Krzyżak, A., Oja, E.: Rival Penalized Competitive Learning for Clustering Analysis, RBF Net, and Curve Detection. IEEE Trans. on Neural Networks, **4** (1993) 636-648
6. Fukunaga, K.: Introduction to Statistical Pattern Recognition (2nd), Academic, Boston (1990)
7. Duda, R. O., Hart, P. E., Stork, D. G.: Pattern Classification (2nd). John Wiley and Sons, New York (2001)
8. Evans, M., Hastings, N., Peacock, B.: Statistical Distributions (2nd). John Wiley and Sons, New York (1993)

A Radar Target Multi-feature Fusion Classifier Based on Rough Neural Network

Yinshui Shi[1,2], Hongbing Ji[1], and Xinbo Gao[1]

[1] School of Electronic Engineering, Xidian University, Xi'an, Shaanxi 710071, China
[2] Air Defense Force Command Academe, Zhengzhou, Henan 450052, China
zz_sys@gr.xidian.edu.cn

Abstract. Fusing Multi-feature will benefit radar target classification with much more belief. However, since radar target attributes such as high range resolution profiles, waveforms, frequency spectra, time-frequency spectra, higher order statistics, polarization spectra and flight path are of different dimensions, it is hard to make decision by fusing multi-feature directly. Fortunately, rough set makes decision by examining the fitness of each condition attribute separately, while neural network is powerful for dealing with nonlinear problems. With radial projection of target dimension, cruising velocity and height as condition attribute, a multi-feature rough neural network fusion classifier is presented. Simulation of the proposed classifier based on an information system with 25 targets belonging to 6 classes shows accuracy not less than 93%, while attributes are subjoined with typical radar errors.

1 Introduction

Single feature based target classification for conventional and high range resolution radar has made great progress. But any single feature could not make a full target description separately. More than one feature such as waveform, flight path, polarization, etc, can be obtained simultaneously in operation. All these features in different dimensions reflect different characters of targets and are important evidences for target classification. Making full use of these features jointly is one of the novel topics of target classification, as well as an effective route to improve accuracy. However, the attributes' being of different dimensions makes it hard to optimize the weighted attribute fusion decision algorithm.

Rough Set is a favorable alternative for mixed dimension attributes fusion decision. It does not need any priori or additional information about data, while it has many advantages such as providing efficient algorithms for generates minimal sets of *if-then* decision rules from data, offering straightforward interpretation of obtained results [1]. Above all, instead of comparing and synthesizing attribute values with different dimensions as other algorithms, rough set makes decision by examining the fitness of each condition attribute separately. Concerning no attributes comparison and integration makes it a powerful tool for dealing with mixed dimension attributes fusion decision, as well as multi-feature fusion radar target classification.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 375–380, 2005.

Neural network is a powerful alternative when there exist no satisfactory analytic model or appropriate linear model. However, neural network does not provide us clear interpretation of knowledge it contains. Rough set has demonstrated the usefulness of upper and lower approximations in the rule generation. One of the application combining the rough set theory and neural networks is to incorporate rough set methods into construction of the neural network by using rough neurons [5].

With radial projection of radar target dimensions, cruising velocity and height as condition attributes, a multi-feature rough neural network (RNN) fusion target classifier is presented. Simulation of the proposed classifier based on an information system with 25 targets belonging to 6 classes shows accuracy not less than 93%, while attributes are subjoined with typical radar errors. The contributions of this paper are the algorithm of multi-feature fusion based radar target classification, and the fusion decision of mixed dimensions stationary stochastic attributes using rough neural network.

This paper is organized as follows. The rough set decision and rough membership function (RMF) underlying the design of RNN are presented in section 2. Section 3 deals with an algorithm for radar target multi-feature fusion classification and its simulation test. The paper concludes in section 4.

2 Basic Concepts of Rough Neural Network [1],[2],[3]

2.1 Rough Membership Function

Each subset $\phi \neq P \subset A$ of information system $S = \{U, A, V, f\}$ defines an indiscernibility relation denoted by

$$IND(P) = \{(x, y) \in U \times U | f(x,a) = f(y,a), \forall a \in P\}. \tag{1}$$

With every $X \subset U$ and $P \subset A$, we associate two sets defined as follows: The lower approximation of $X : \underline{P}X = \{y \in IND(P) | [y]_p \subset X\}$; and the upper approximation of $X : \overline{P}X = \{y \in IND(P) | [y]_p \cap X \neq \phi\}$.

We use attributes to classify the objects into classes. But there may be redundant attributes in primary information systems. Reducing the redundant attributes will reduce the computation while holding classification ability. Attribute $a \in P$ is superfluous in P, if $IND(P - \{a\}) = IND(P)$. Otherwise it is indispensable. Attributes reduction is an effective feature compression method for radar target classification.
The RMF is defined as follow.

$$\mu_x^P : U \qquad [0,1], \qquad \mu_X^P(x) = |[x]_P \cap X| / |[x]_P| \cdot \tag{2}$$

Which qualifies the degree of relative overlap between the set X and the equivalence class $[x]_P$ to which x belongs.

2.2 Rough Set Decision

Every object $x_i \in U, i = 1,2,\cdots,|U|$ of a decision system $S = \{U, C, \{d\}\}$corresponds to a decision rule. Let $X_i \in U / IND(C)$ is an equivalent class based on C; $Y_j \in IND(\{d\})$ is an equivalent class based on$\{d\}$; $des(X_i)$ is the value assignment description of X_i for each condition attribute; $des(Y_j)$is the value assignment description of Y_j for the decision attribute. We can create the decision rule to determine $X_i \in X$ belongs to Y_j as $r_{ij} : des(X_i) \to des(Y_j)$ from the information system or reduced attribute set.

2.3 Rough Neurons

Approximation neurons and decider neurons are used to construct the rough neural classifier (Figure 1). $a_i, i = 1,2,\cdots,n$ are feature (attribute) values of the target to be classified, which stimulates the upper approximate neurons.

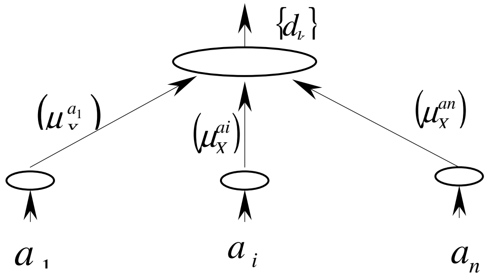


Fig. 1. A rough neural network for fusion classification

Approximation neurons compute rough memberships $\mu^{ai}_{\overline{ai}X}, i = 1,2,\cdots,n$, which describe the degree of relative overlap between $[u]_{ai}$ and $\overline{a_i}X$ or the decision uncertainty for input targets.

Decider neurons store decision rules set $R = \{a_i \to d_i\}$ derived from current information. Under the stimuli of input object attributes, approximation neurons compute rough memberships and pass to decider neurons. Decider neurons make decision in steps as follows: (1) constructing a condition vector $a_{\exp} = \{\mu^P_X(\overline{P}X, [u]_P)\}$ from its input which are RMF values; (2) discovering the rule $a_i \to d_i$ with a condition vector a_i which matches the input condition vector $a_{\exp}$; (3) outputting the decision d_i.

3 Multi-feature RNN Fusion Radar Target Classifier [2],[4]

3.1 Features Extraction

For conveniency, range width derived from high range resolution profile, cruising velocity and height output from data processor are chosen as condition attributes. The radial projection width D_r of target can be derived from range profiles, which is denoted as following in the radar reference frame

$$D_r = f(\Delta R,\tau,c) = \max \{f_1(L,\beta,\varepsilon,R,R_e), f_2(W,\beta,\varepsilon,R,R_e)\}. \tag{3}$$

Where f is the target' radial projection function. $f_1(\cdot)$ is the length projection function, while $f_2(\cdot)$ is the width function. ΔR is the radar echo's effective width. τ is the emitter pulse width. c is the propagation velocity of electromagnetic wave. β is the azimuth in which target is lighten, while ε is the elevation. R is the target slant range. R_e is the earth radium. L is the target length, while W is the width.

The attribute values of cruise velocity V and cruise height H are assigned by radar data processor.

Table 1 presents a current information system containing 25 targets attributes, including length, width, cruise velocity and height.

3.2 Rough Neural Classifier

Because the signal integration of target classification usually takes place in smooth flight or cruise, the attribute values from the signal processor and data processor are normal distributed stationary stochastics, which decrease classification accuracy.

The RNN classifier consists of one decider neuron and three upper approximation neurons defined relative to a single attribute separately. Computation of $\mu^{ai}_{\underline{ai}X}, i = 1,2,3$ is mainly for the purpose of improving classification robustness affected by the attribute stochastics. There are different classes that can be classified from different aspects. We classify targets into large jet (LJ), middle jet (MJ), small jet (SJ), large propeller (LP), small propeller (SP) and cruise missile (CM). We have derived attribute value sets from Table 1 as follows.

$$V_{a1} = \{0-18.4,18.4-19.8,19.8-22.4,22.4-36.1,36.1-38.0,38.0-70.70\}$$
$$V_{a2} = \{0-400,400-700,700-1910\}$$
$$V_{a3} = \{0-500,500-8000,8000-12000,12000-17500\}$$
$$V_d = \{LJ, MJ, F, LP, SP, CM\}. \tag{4}$$

The decision rules derived from reduced decision table are given as follows.

$$\mu^{a1}_X(2/7 or 6/7) \quad and \quad \mu^{a2}_X(1) \quad and \quad \mu^{a3}_X(1) \rightarrow LJ$$
$$\mu^{a1}_X(1/6 or 1/2 or 1) \quad and \quad \mu^{a2}_X(1) \quad and \quad \mu^{a3}_X(1/6 or 5/6) \rightarrow MJ$$
$$\mu^{a1}_X(1/3 or 1/2 or 1) \quad and \quad \mu^{a2}_X(1) \quad and \quad \mu^{a3}_X(1/6 or 5/6) \rightarrow LP$$
$$\mu^{a1}_X(2/3 or 1/2) \quad and \quad \mu^{a2}_X(1) \quad and \quad \mu^{a3}_X(1) \rightarrow SP$$

$$\mu_X^{a1}(1/5 or 1) \quad and \quad \mu_X^{a2}(1) \;\rightarrow\; SJ$$
$$\mu_X^{a1}(1) \quad and \quad \mu_X^{a2}(1) \quad and \quad \mu_X^{a3}(1) \;\rightarrow CM\,. \tag{5}$$

Rough approximation neurons and decider neurons are implemented using Matlab. A typical set of conventional radar accuracy including radial width measurement mean square error (MMSE) $\sigma_L = 1m$, velocity MMSE $\sigma_V = 4km/h$, height MMSE $\sigma_H = 400m$. The input attribute values of rough approximation neurons are

$$a_i = a_{0i} + \sigma_j \cdot randn(1) \qquad i = 1,2,3 \quad j = L,V,H \tag{6}$$

Where a_{0i} is the attribute value in table 1, $randn(1)$ is a normal distributed random number with mean 0, variance 1and standard deviation 1.

Table 1. A target information system

Obj	length	Width	Radial W.(a_1)	cruise V.(a_2)	cruise H.(a_3)	Type	Class
1	63.70	60.90	44.26-63.7	935	11000	B777	LJ
2	56.30	59.60	41.11-59.6	935	11280	B747SP	LJ
3	70.70	59.60	46.15-70.7	935	10670	B747F	LJ
4	33.40	28.90	22.00-33.4	856	10670	B737	LJ
5	59.40	58.60	41.86-59.40	850	10670	A340	LJ
6	37.60	33.90	25.30-37.60	850	10670	A320	LJ (MJ)
7	31.00	26.30	20.10-31.00	720	9449	Bae146	MJ
8	56.00	48.00	36.10-56.00	850	11000	IL86	LJ(MJ)
9	61.20	51.70	40.00-61.20	850	9450	MD11	LJ
10	35.50	28.10	22.10-35.50	800	11000	Fok100	MJ
11	33.10	38.00	25.00-38.00	550	10000	AN12	LP
12	23.70	29.20	18.50-29.20	423	6000	Y-7	LP (SP)
13	25.70	27.40	18.60-27.40	470	4500	DHC8	LP (SP)
14	36.40	34.90	25.30-36.40	750	7600	Yak42	MJ (LJ)
15	15.80	19.80	12.40-19.80	318	3040	DHC6	LP (SP)
16	12.20	14.00	9.20-14.00	277	7600	K. A	SP
17	14.90	17.30	11.30-17.30	292	<7000	Y-12	SP
18	12.70	18.20	10.40-18.20	220	<1500	Y-5	SP
19	13.90	7.150	6.40-13.90	975	<17500	J-7	SJ
20	14.64	9.040	7.70-14.60	900	<17500	J-6	SJ
21	17.07	11.43	11.40-17.07	<1910	<15240	F-18	SJ
22	15.03	9.450	8.00-15.00	849	11000	F-16	SJ
23	6.250	2.670	2.50-6.25	850	<100	BGM109	CM
24	40.40	29.30	23.80-40.40	556	6096	L-100	LJ
25	22.40	12.71	11.10-22.40	975	<17500	JH-7	SJ

The simulate RNN classifier operates 500 times for every target in Table 1. The confusion matrix of the mean accuracy in Table 2 shows the validity of RNN to make decision with radar target's mixed dimension attributes.

Obviously, the proposed classifier shows an excellent performance not less than 93%. It is also should be observed that errors of radar signal processor and data processor give a limitation to classification accuracy, and the error can also confuse classification as shown in Table 2. Enduing attributes in subsection is useful for improving the robustness of the RNN.

Table 2. RNN Classifier Confusion Matrix

	LJ	MJ	SJ	LP	SP	CM
LJ	99.7	0.08	0	0	0	0
MJ	2.7	96.6	0.2	0	0	0
SJ	0	0	99.2	0	0	0
LP	0	0	0	97.4	0	0
SP	0	0	0	0	93.6	0
CM	0	0	6.3	0	0	93.7

4 Conclusions

A RNN is utilized to fuse multiple stationary stochastic features in different dimensions of radar targets and make decisions to classify the targets successfully. Apparently, the more the radar target features are used, the more robust and believable the classification is. Although radar target features are affixed with a normal random error, it is fortunate that RNN can make decision by examining the fitness of every attribute of condition vector separately. Simulate classification verifies the effectiveness of the proposed algorithm.

References

1. Pawlak, Z.: Why Rough Sets? Proceedings of the Fifth IEEE International Conference on Fuzzy Systems, **2** (1996) 738-743
2. Petters, J.F., et al.: Rough Neural Computing in Signal Analysis. Computational Intelligence, **17** (2001) 493-513
3. Petters, J.F., et al.: Towards Rough Neural Computing Based on Rough Membership Functions. Theory and Application, RSCTC 2000, LNAI 2005, Springer-Verlag Berlin Heidelberg (2001) 611-618
4. Petters, J.F., et al.: Design of Rough Neurons. Rough Set Foundation And Petri Net Method, ISMIS 2000, LNAI 1932, Springer-Verlag Berlin Heidelberg (2000) 283-291
5. Wu, Y., et al.: A Rough Neural Network for Material Proportioning System. Communications, Circuits and Systems and West Sino Expositions, IEEE 2002 International Conference on, **2** (2002)

Automatic Digital Modulation Recognition Based on ART2A-DWNN

Zhilu Wu, Xuexia Wang, Cuiyan Liu, and Guanghui Ren

School of Electronics and Information Technology, Harbin Institute of Technology
Harbin, Heilongjiang 150001, China
{wuzhilu,wangxuexia,liucuiyan,rgh}@hit.edu.cn

Abstract. A novel automatic digital modulation recognition classifier combining adaptive resonance theory 2A (ART2A) with discrete wavelet neural network (DWNN), called ART2A-DWNN, is proposed in this paper. The modified ART2A network with a low vigilance parameter is used to categorize input modulation schemes into some classes and then DWNN is employed in each class to recognize modulation schemes. Moreover, error back propagation (BP) learning algorithm with momentum is adopted in DWNN to speed up the training phase and improve the convergence capability. Simulation results obtained from modulated signals corrupted with Gaussian noise at 8dB Signal to Noise Ratio (SNR) are given to evaluate the performance of the proposed method and it is found that the benefits of the developed method include improvement of recognition capability, training convergence enhancement and easiness to accommodate new patterns without forgetting old ones.

1 Introduction

Recently, one of the hottest topics in wireless communication is software-defined radio (SDR). With the number of modulation schemes increasing, automatic digital modulation recognition (AMR) has become more important in the research of SDR. AMR is used to identify different modulation schemes in SDR. Furthermore, AMR can also be used in civil applications as well as in military applications, such as interference identification, spectrum management, and electronic warfare [1].

Many kinds of ANNs using BP learning rule have been maturely applied in AMR. However, under certain conditions, the BP network classifier can produce non-robust classification results and easily converge to a local minimum. Moreover, it is time consuming in training phase and is not convenient to accommodate new patterns. To solve these problems, a new type of neural network, ART2A-DWNN, is proposed in this paper, which employs an improved unsupervised ART2A network to sort a large number of input patterns into several classes, and then use a three-layer supervised DWNN after each class nodes in the output layer of ART2A for further classification.

This paper is organized as follows. Section 2 states the ART2A-DWNN including discussions of ART2A network and DWNN. Section 3 describes the AMR system using ART2A-DWNN. Section 4 summarizes simulation results of the presented solution. A conclusion is given in Section 5.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 381–386, 2005.

2 ART2A-DWNN

ART2A-DWNN is composed of ART2A neural network and DWNN, as shown in Fig.1. Its training-recognition procedure can be described as follows. Input patterns are clustered into classes by ART2A layer. At this stage, coarse classification is carried out so that patterns with similar features are clustered together. Patterns in each class are then put forward as the inputs of the corresponding DWNN for further classification.

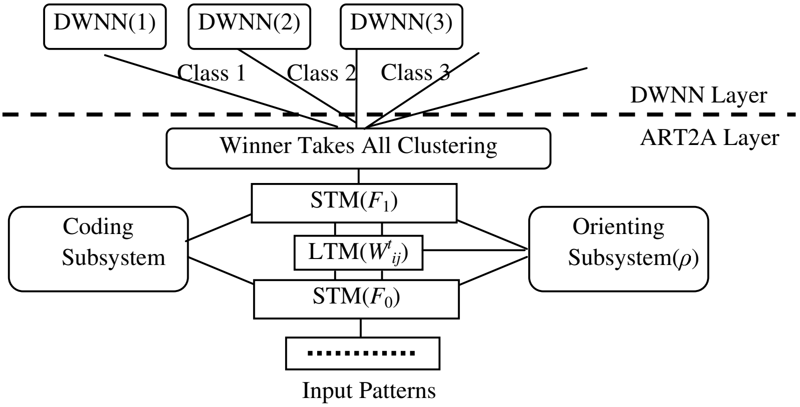


Fig. 1. The structure of ART2A-DWNN

2.1 ART2A Neural Network

ART2A is a modified network of ART2, which is a category learning system that self-organizes a sequence of either binary or analog inputs into various recognition classes. References [2],[3] have given the mechanism of ART2A in detail. ART2A architecture depicted in Fig.1 consists of an input layer F_0, a recognition layer F_1, a coding subsystem and an orienting subsystem that controls the stability-plasticity trade-off. The interconnection weights between F_0 and F_1 are considered as long-term memory (LTM). ART2A network follows a winner-take-all competitive learning rule.

The modified self-organizing processing by an ART2A network consists of preprocessing-stage, searching-stage and adaptation-stage. The searching-stage will be a procedure of *choice*, *match* and *reset*. This modified ART2A network adopts a Euclidean measurement of similarity and skip the length normalization of inputs in the preprocessing and adaptation stage. All of the input vectors X should be fitted to the internal [0, 1]. For each node j in F_1 layer, the *choice* function T_j is defined by $T_j=1-\|X-W_j\|/\sqrt{N}$, where W_j is the only feed-forward connection weight vector of node j, $j=1, 2, 3, \ldots, N$. The choice of a winner node is indexed at J, where $T_J=\max\{T_j$: for node j in $F_1\}$. Mismatch reset happens when the network fails to locate a winner category after the first input is presented, or when the choice score T_J doesn't reach the vigilance value $T_J <\rho$ $(0<\rho<1)$. At the same time, a new category K is created by copying X as its weight vector W_K=X. Otherwise, the network is ready to reach resonance while the coding subsystem updates the weight vector W_J according to $W_J(t+1)=W_J(t)+\eta[X(t)-W_J(t)]$ $(0<\eta<1)$.

Apparently, the computational complexity and the dynamics of ART2A are determined by the vigilance parameter ρ and the learning rate η. It is feasible that the small vigilance value ρ is beneficial to the stability of ART2A for coarse classification. Hence we choose ART2A as the first layer of ART2A-DWNN.

2.2 Discrete Wavelet Neural Network

The concept of DWNN has been inspired by both the technologies of wavelet decomposition and neural networks. In DWNN, the nonlinearity is approximated by superposition of a series of discrete wavelet functions. Structure of DWNN is similar to that of BP network, but the activation functions of the units in hidden layer and the output layer are replaced by wavelet functions $\Psi_{jk}(x)$ and linear summing functions respectively [4],[5]. The output of the nth unit in the output layer is given as:

$$y_n = \sum_{t=1}^{M} w_{nt} 2^{j/2} \Psi(2^{j/2} \sum_{s=1}^{I} u_{ts} X(s) - k) \quad j,k \in Z \tag{1}$$

where w_{nt} indicates the weight between the nth node of output layer and the tth node of hidden layer, u_{ts} is the weight between the tth node of hidden layer and the sth node of input layer, j is the dilation parameter and k is the translation parameter. The number of nodes of hidden layer and input layer are M and I respectively.

An error vector E can be defined as follows:

$$E = 0.5 \sum_{n=1}^{N} (d_n - y_n)^2 \tag{2}$$

where d is the desired output vector. Generally, the training is based on the minimization of E, performed by iterative conjugated gradient-based method. For each iteration, the parameters w_{nt} and u_{ts} are modified using the conjugated gradient with momentum according to:

$$\Delta\theta^{t+1} = -\eta \partial E / \partial \theta^t + \alpha \Delta \theta^t, \theta^{t+1} = \theta^t + \Delta\theta \tag{3}$$

where η is the learning rate, α is the momentum term, θ denotes the parameters w_{nt} and u_{ts} which should be trained.

Because of the summing activation function of output layer, time-frequency analysis feature of wavelet and error back propagation learning with momentum, DWNN benefits from fast training without local minima and have high probability of recognizing intertwined signals. However, the simulation results show that the more the modulation types, the worse the convergence and classification capability of DWNN.

2.3 The Advantages of ART2A-DWNN Network

The Recognition capability is improved by ART2A-DWNN [6]. Normalized input vectors are mapped from an R^n space to an R^2 space, shown in Fig.2. In this figure, D_1, D_2 and D_3 denote the classes categorized by ART2A, and m_1, m_2 and m_3 are centroids of them. It is found that the winning synaptic vector m_j of ART2A equals the centroid of the Class D_j. It means that m_j defines the deterministic center of the Class D_j.

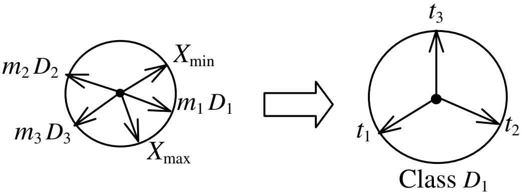


Fig. 2. Three patterns in Class D_1 are mapped into classified patterns by Sammon's scheme

The range of class D_1 is bounded by $[X_{\min}, X_{\max}]$ which depends on the vigilance parameter. The small vigilance parameter can increase the range of the class. And suppose there are three patterns t_1, t_2 and t_3 in Class D_1 which has a centroid m_1. Now we assign each pattern a specific vector in the output layer of the DWNN. By using the Sammon's nonlinear mapping algorithm, t_1, t_2, t_3 will be located around a circle as depicted in Fig.2. The DWNN can recognize those three patterns successfully because explicit boundary can be easily formed. So ART2A-DWNN network can classify two patterns intertwined, but neither the ART2A network nor the DWNN can.

When a new example pattern is added, it will be either categorized into one of the existing classes or a new class by ART2A. Then only the class which this new pattern belongs to needs to be retrained. This character is very important from the extendibility viewpoint. On the contrary, if only the DWNN is applied, it has to be retrained using the whole (old and new) patterns. After clustered by ART2A, fewer patterns will be located in the same class. Thus, when applying DWNN to each class, training time is dramatically reduced. And just as analysis on DWNN in section 2.2, problems of converging to local minima will be diminished. Therefore, the DWNN can produce satisfactory results.

3 AMR System Based on ART2A-DWNN

In an AMR system, the wireless communication signals are received by RF front-end including amplification, filtering and IF down-conversion or base-band down-conversion, then followed by analogue digital converter and digital signal processing. The next processing module is the AMR module composed of feature extraction and ART2A-DWNN classifier proposed above. Here the features chosen to characterize the digitally modulated signals are $\gamma_{\max}$, σ_{ap}, σ_{dp}, σ_{aa} and σ_{af} which are set up by Azzouz and Nandi [7]. And the "morlet" function is defined as:

$$\Psi(x) = -(\cos 1.75x)e^{-0.5x^2} \tag{4}$$

Function (4) which has a high resolution in both frequency domain and time domain is selected as the mother wavelet. Moreover, we choose a simple structure of DWNN with a 5-node input layer, a single hidden layer with 25 nodes and only a 2-

node output layer instead of a 3-node output layer for classifying 3 signals. At the end of its training, the network performs a binary classification on each given input pattern. The value of each node in the output layer are designed as ‘1’ and ‘0’, which forms the output vector from ‘01’ to ‘11’ to express different signals.

4 Experimental Results

In this experiment, the simulation is carried out in MATLAB environments, and seven digital modulation schemes (2FSK, 4FSK, 2PSK, 4PSK, GMSK, 2ASK and 4ASK) have been chosen as the training signals which are corrupted with additive white Gaussian noise at 8dB SNR. The vigilance parameter is 0.78 for ART2A layer, and the translation and dilation parameters range from -2 to +2 for DWNN layer, but only integers are valid for them. Three kinds of experiments are implemented to measure the performance of the proposed method. Firstly, we utilize DWNN, ART2A and ART2A-DWNN classifiers respectively to recognize the constant envelope modulation signals mentioned above. Then new data of 2ASK and 4ASK are added to the recognition system to evaluate its capability of accommodation. At last, the curves of the training convergence of DWNN and BP network for DQPSK, GMSK and π/4DQPSk are shown in Fig. 3. And the results of all simulations are based on 400 realizations for each modulation type at 8dB SNR.

Table 1. The accuracy rates for different classifiers at 8dB SNR

Classifiers	Class 1			Class 2	Class 3
	2FSK	4FSK	4PSK	2PSK	GMSK
DWNN	97.1%	98.2%	95.2%	70.6%	70.1%
ART2A	96.7%	23.3%	95.3%	100%	92.1%
ART2A-DWNN	97.4%	98.6%	98.6%	94.5%	98.5%

Table 2. The accuracy rates after adding new data of 2ASK and 4ASK at 8dB SNR

Classifiers	Class 1			Class 2	Class 3	Class 4	
	2FSK	4FSK	4PSK	2PSK	GMSK	2ASK	4ASK
Commited output nodes of ART2A	1~2	1~2	1~2	3~4	5~7	8~14	8~14
Accuracy rate	96.4%	99.6%	94.5%	98.6%	99.6%	91.9%	85.2%

Comparing the results of Table.1, we can see that ART2A-DWNN can classify the five similar modulation schemes with average accuracy rate greater than 94% successfully, but neither ART2A network nor DWNN does. Hence it has higher recognition capability than DWNN and ART2A network. And the results obtained from Table.2 show that the new data of 2ASK and 4ASK can be self-organized into a new class and only the new DWNN requires to be trained. Furthermore, the accuracy recognition rates for old patterns are not affected by 2ASK and 4ASK, and all the patterns have high recognition rates except 4ASK. And because of the faster training phase of DWNN than BP network demonstrated in Fig.3, DWNN is selected as the second layer of ART2A-DWNN neural network.

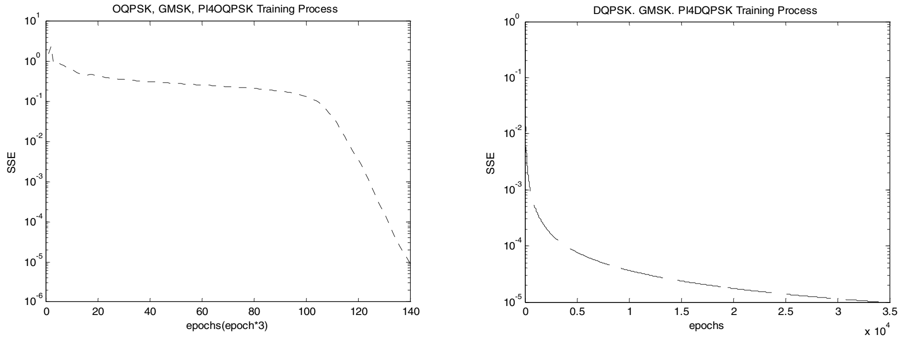


Fig. 3. Comparison of training phases for DWNN network (*left*) and conventional BP neural network (*right*): the output error vs. epochs. When the maximum output error *E* is 0.00001, 420 iterations, η=0.1 for DWNN and 34000 iterations, η=0.8 for BP network

Simulations at 5dB SNR and 15dB SNR have also been carried out to measure the noise immunity of the developed system. While the random noise level decreases from 15dB to 5dB, the ART2A-DWNN still can give high accurate results and there is no obvious change in the accuracy rates of different SNRs.

5 Conclusions

In this paper, we have developed a new automatic digital modulation recognition method based on the combination of ART2A neural network and DWNN, namely ART2A-DWNN. The experimental results have indicated the advantages of ART2A-DWNN, such as the higher recognition capability, the improved training convergence and convenience of accommodating new patterns. We may claim that the ART2A-DWNN can handle much more patterns than the ART2A network or the DWNN. So ART2A-DWNN neural network is a better alternative for AMR. Since the normalization method of input vectors of DWNN is very important for the recognition performance, the relative low recognition rate for 4ASK may be improved by finding an appropriate normalization solution. Our future work will continue in this direction.

References

1. Louis, C., Sehier, P.: Automatic Modulation Recognition with a Hierarchical Neural Network. IEEE (1994) 713-717
2. Carpenter, G.A., Grossberg, S., Rosen, D.: ART 2-A: An Adaptive Resonance Algorithm for Rapid Category Learning and Recognition. IEEE (1991) 151-156
3. Thomas, F., Karl-Friedrich, K., Torsten, K.: Comparative Analysis of Fuzzy ART and ART-2A Network Clustering Performance. IEEE Transactions on Neural networks
4. Huang, Y.C., Huang, C.M.: Evolving Wavelet Networks for Power Transformer Condition Monitoring. IEEE Transactions on Power Delivery, **17** (2002) 412-416
5. Li, H.L., Xiao, D.M., Chen, Y.Z.: Wavelet ANN Based Transformer Fault Diagnosis Using Gas-in-oil Analysis. IEEE (1999) 147-150
6. Wu Y.T., Tai, H.M., Reynolds, A.C.: An ART2-BP Neural Net and Its Application to Reservoir Engineering. IEEE (1994) 3289-3294
7. Azzouz, E.E., Nandy, A.K.: Procedure for Automatic Modulation Recognition of Analog and Digital Modulations. IEEE Pro-Commun, **143** (1996)259-266

Recognition of Radiated Noises of Ships Using Auditory Features and Support Vector Machines*

Xinhua Zhang, Chunyu Kang, and Zhijun Xia

Research Center of Signal and Information, Dalian Navy Academy
Dalian, Liaoning 116018, China
dlkangcy@sohu.com

Abstract. In order to make effective recognition of radiated noises of ships, on the basis of the auditory Patterson-Holdsworth cochlear model and Meddis' Inner Hair Cell (IHC) model, a feature extraction of radiated noises of ships model simulating the partial auditory system is set up to obtain the average firing rate. Then an algorithm (One-Against-All: OAA) of multi-class Support Vector Machines (SVMs) is defined. Finally, the extracted feature vectors are used to classify three different classes of targets using SVMs, BP Neural Network (BPNN) and K-Nearest Neighbor (KNN) methods. At the same time we compare the recognition performance of average firing rate feature with general power spectrum feature. Results show that the statistical recognition corrective rate of average firing rate feature exceeds 96.5% using SVMs.

1 Introduction

The central technology of underwater acoustic equipments and intelligence weapon systems is automatic recognition for underwater acoustic targets. The target feature extraction and recognition are two key techniques to solve that problem. Recognition of underwater acoustic targets is a special pattern recognition problem with high credibility and simultaneity. The one of main difficulty is to obtain a better feature representation describing target characteristics under noise interference. Many specialists and scholars have been studying on this subject. As we known, Hominine ears have the high capability of recognition by the nature evolution. Sonar operator can make valid judgment to a target sound through mass training. This stimulates us to explore auditory models for extracting more stable features for underwater targets recognition.

This paper is structured as follows: In Section 2 we present the feature extraction model of radiated noises of ships using auditory Equivalent Rectangular Bandwidth (ERB). In section 3 we propose a Support Vector Machines (SVMs) model for recognition. Finally, in section 4 we make the comparison and analysis about the simulation results using different feature extraction methods and different recognition methods.

2 Feature Extraction Model of Radiated Noises of Ships

2.1 Patterson-Holdsworth Cochlear Model

The cochlea is responsible for converting sounds from mechanical vibrations into electrical signals. The electrical signals, which code the sound's characteristics, are

* This Paper was supported by NSFC (No. 60472108).

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 387–392, 2005.

carried to the brain by the auditory nerve. The movement of the stapes results in the liquid intensity of pressure changes in cochlea. The sound signal is decomposed different frequency characteristic signal at different space axes through the cochlea processing, thus the selection of sound frequency is realized. This frequency selectivity can be described in terms of an "Equivalent Rectangular Bandwidth (ERB) " as a function of center frequency. And we known the radiated noises of ships are formed by mechanism noises, screw noises, hydrodynamic noises and so on. So radiated noises of ships contain different frequency. In this paper, we adopt cochlea model defined by Patterson-Holdsworth for simulating the cochlea[1], structuring a set of Gammatone filters. The filter bank is designed as a set of parallel band-pass filters, each tuned to a different frequency. Each Gammatone filter channel is composed of four half orthonormal second order filters. The transfer function for these four filters share the same poles but have different zeros, and can be written as Eq.(1).

$$H(Z)=\frac{A_0+A_{k1}Z^{-1}+A_2Z^{-2}}{B_0+B_1Z^{-1}+B_2Z^{-2}} \qquad k=1,2,3,4\,\cdot \tag{1}$$

Where $B_0=1$, $B_1=-\frac{2\cos(2\pi f_cT)}{e^{BT}}$, $B_2=e^{-2BT}$, $A_2=0$, $A_0=T$, $A_{11}=\frac{-T\cos(2\pi f_cT)-\sqrt{3+2^{1.5}}\,T\sin(2\pi f_cT)}{e^{BT}}$, $A_{21}=\frac{-T\cos(2\pi f_cT)+\sqrt{3+2^{1.5}}\,T\sin(2\pi f_cT)}{e^{BT}}$, $A_{31}=\frac{-T\cos(2\pi f_cT)-\sqrt{3-2^{1.5}}\,T\sin(2\pi f_cT)}{e^{BT}}$, $A_{41}=\frac{-T\cos(2\pi f_cT)+\sqrt{3-2^{1.5}}\,T\sin(2\pi f_cT)}{e^{BT}}$, $T=\frac{1}{f_s}$, $B=2\pi*1.019(\frac{f_c}{9.26449}+24.7)$, f_s is the sample frequency, f_c is the center frequency corresponding each filter channel. And each filter's gain is

$$G=\left|\frac{F_1F_2F_3F_4e^{8BT}}{\left[-1+e^{BT}+e^{(BT+4f_cT\pi i)}-e^{(2BT+4f_cT\pi i)}\right]^4}\right|\cdot \tag{2}$$

Where $i=\sqrt{-1}$, $F_1=S_1+S_2[S_3+\sqrt{3+2^{1.5}}S_4]$, $F_2=S_1+S_2[S_3-\sqrt{3+2^{1.5}}S_4]$, $F_3=S_1+S_2[S_3+\sqrt{3-2^{1.5}}S_4]$, $F_4=S_1+S_2[S_3-\sqrt{3-2^{1.5}}S_4]$, $S_1=-Te^{4f_cT\pi i}$, $S_2=Te^{(-BT+2f_cT\pi i)}$, $S_3=\cos(2f_cT\pi)$, $S_4=\sin(2f_cT\pi)$.

The impulse response of the whole Gammatone filter is defined as $g(t)=\frac{at^{n-1}\cos(2\pi f_ct+\varphi)}{\exp(2\pi bt)}$, where t is time, a is gain, n is the order of filters, f_c is the center frequency, b is the bandwidth of filters. If we change the center frequency and bandwidth parameters we can get different band-pass filters. Fig.1 is the impulse response of Gammatone filters when $a=1$, $n=5$, $f_c=2000Hz$, $b=200$. The frequency response of a bank of Gammatone filters of twenty ERB filters between $0.1Hz$ and $12500Hz$ is showed as Fig.2.

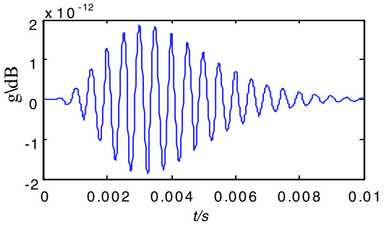


Fig. 1. Impulse response of Patterson cochlear model

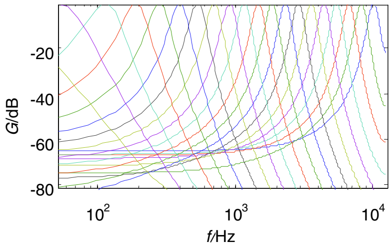


Fig. 2. Frequency response of a bank of Gammatone filters

2.2 Meddis' Inner Hair Cell Model

After the sound signal is filtered by Gammatone filters, the Inner Hair Cells (IHC) transform the mechanical displacement of the basilar membrane into firing activity at the nerve fibers. The transformation from mechanical motion to receptor potential involves several nonlinearities, such as the half-wave rectification (HWR), short-term adaptation and so on. So there are inhibition and competition among different frequency. And the frequency components which have higher energy are strengthened, the lower energy frequency components are more weaken. On the other hand the radiated noises of ships are formed by many frequency components, so we add the Meddis' IHC model[2],[3] to the back-end of a Gammatone filter bank to form a "complete" auditory model to process the radiated noises of ships. Fig.3 is Meddis' IHC model. We conclude $A = 10$, $B = 1140$, $C = 1000$ through mass test.

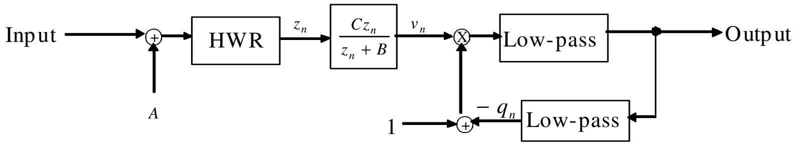


Fig. 3. Meddis' inner hair cell model

2.3 Feature Extraction of Radiated Noises of Ships

According to the analysis mentioned above, we firstly make a preemphasis of the radiated noises with a first order high pass filter. This flattens the typical spectral tilt of signals and reflects the transfer function of the outer ear. The preemphasized signal is then filtered by a Gammatone filter-bank using 100 frequency channels equally spaced on the ERB scale with center frequency from $0.1Hz$ to $12500Hz$. After Gammatone filtering, the signal of each frequency channel is passed through Meddis' IHC model and low-pass filter. Then we get the average envelope of firing-rate patterns. These averaged firing-rates construct a feature vector for target recognition. Fig.4 shows this feature extraction process of radiated noises of ships.

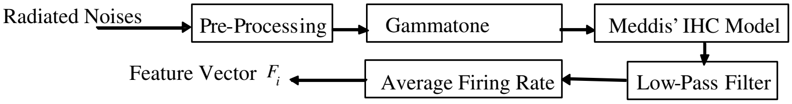


Fig. 4. Feature extraction of radiated noises of ships

3 Support Vector Machines Classifier

SVMs are based on the structural risk minimization principle. Vapnik[4] showed that for SVMs, this goal can be translated into finding the hyper-plane with maximum margin for separable data. In the basic form, SVMs classifiers learn binary classes, using linear decision rules $h(x) = sign(w \cdot x - b)$, where w is the vector of learned weights, x is the input vector, b is the threshold and $h(x)$ is the hypothesis for which one can guarantee the lowest probability of error. $h(x) = 1$ and $h(x) = -1$ represents the different hypothesis.

The standard SVMs are designed for dichotomic classification problem. But the recognition and classification of radiated noises of ships are multi-class classification problem. The multi-class classification problem is commonly solved by a decomposition to several binary problems[5]. For instance, one-against-all (OAA) decomposition, one-against-one (OAO) decomposition and directed-acyclic-graph (DAG) methods are often applied. In this paper, we use OAA decomposition to finish the recognition and classification of radiated noises.

3.1 One-Against-All Method

A typical SVMs model of OAA method[6],[7] used for recognition of radiated noises is illustrated as Fig.5.

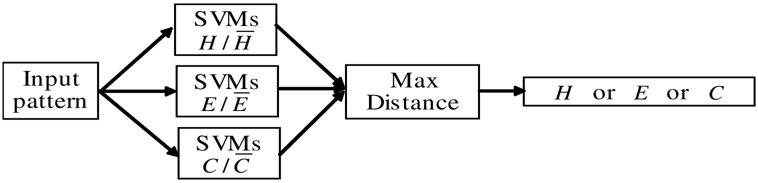


Fig. 5. One-against-all multi-class classification

Three binary SVMs classifiers, $H/\overline{H}$, $E/\overline{E}$, $C/\overline{C}$ are constructed. The input vectors, the feature vectors extracted using above methods, are transformed to a high dimensional space and compared to the support vectors via a kernel function for each classifier. The results are then linearly combined by using parameters α_i , $i = 1,2,\cdots,N$ that are found by solving a quadratic optimization problem. The process of training of each classifier $k/\overline{k}$, $k \in \Sigma_T$ is to to maximize $Q^k = \sum_{i=1}^{N} \alpha_i^k - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i^k \alpha_j^k T_i^k T_j^k \mathrm{K}(v_i, v_j)$, subject to $0 \le \alpha_i^k \le \gamma$ and $\sum_{i=1}^{N} \alpha_i^k T_i^k = 0$, where $\{v_1, v_2, \cdots, v_N\}$ are training feature vectors, $\{T_1^k, T_2^k, \cdots, T_N^k\}$ are class labels.

The algorithm, when the cost function Q^k is optimized, yields a classifier for $k/\bar{k}$ with maximum margin of separation. The summations of the maximizing function Q^k run over all training patterns. $\mathrm{K}(v_i, v_j) = \phi(v_i)\phi(v_j)$ denotes the kernel function where ϕ is a mapping used to convert input vectors v_i into a high dimensional space, T_i^k encodes the secondary structure such that a binary value $+1$ if the secondary of the residue r_i is the secondary structure k or -1 otherwise, and γ is a positive constant used to decide the trade-off between training error and the margin. Once the parameters α_i are obtained from the above algorithm, the resulting discriminant function is $D^k(v_j) = \sum_{i=1}^{N} T_i^k \alpha_i^k \mathrm{K}(v_i, v_j) + b^k = w^k \phi(v_j) + b^k$, where the bias b^k is chosen so that $T_i^k D^k(v_i) = 1$ for any i with $0 < \alpha_i^k < \gamma$ and the weight vector $w^k = \sum_{i=1}^{N} T_i^k \alpha_i^k \phi(v_i)$. The secondary structural type t_j of the residue r_j is determined by taking the highest value of three discriminant function values $t_j = \arg \max_{k \in \{H, E, C\}} D^k(v_j)$.

3.2 Selection of Kernel Functions

The kernel function, in common use, has different types. In this paper, we select Radiated Basis Function (RBF): $\mathrm{K}\left(v_i, v_j\right) = \exp\left(-\mu \left\| v_i - v_j \right\|^2\right), \mu > 0$ (where μ is the kernel parameters) as the kernel function. First, RBF can non-linearly map samples to higher dimension space, thereby it solves the non-linearly connection between class labels and properties. Second, RBF has less numbers of hyper-parameters than polynomial kernel function, thus the complexity of selecting the model becomes easy. Finally, numerical value limit of RBF is less.

4 Experiments and Conclusions

All of the experimental data come from real radiated noise signal of three types targets (I, II, III) which work in various operating, ocean and weather conditions. The train and test samples are selected according to 1:10.

We obtain the average firing rate feature vectors of radiated noises of ships using above method, and obtain the power spectrum feature vectors according to Welch spectrum evaluation[8],[9], then we adopt above SVMs classifier to finish the recognition of radiated noises. The results of recognition are compared with the BPNN and the KNN method. Table 1. is the right recognition probability of each type targets.

Table 1. Right recognition probability of each type targets

Feature type	**Average firing rate**			**Welch Spectrum**		
Method	**SVMs**	**BP NN**	**KNN**	**SVMs**	**BPNN**	**KNN**
I	89.80%	87.41%	82.65%	88.44%	86.48%	82.40%
II	98.70%	97.58%	96.02%	97.98%	97.49%	96.58%
III	88.33%	86.67%	82.00%	88.00%	85.33%	81.33%
Total	96.50%	95.16%	92.88%	95.74%	94.84%	93.21%

According to the recognition experiment results, the conclusions are obtained.

- The feature vector based on auditory model has a good classification effect to radiated noises of ships recognition, and can provide evidence for the Identification Friend or Foe in future war. It shows that the average firing rate feature vectors have separability among types.
- The recognition probability based on average firing rate feature vectors is better than the general power spectrum feature vectors. It shows that the speech signal analysis technology can be used at sonar signal processing field.
- Different classifier has different recognition results. It shows that the classifier acts as an important part for target recognition. Next, we'll study the recognition performance of other type SVMs.
- How to improve the feature vectors of radiated noises of ships extracted using auditory model to adapt different target types, various operating conditions and different hydrological conditions and to resist noise is our main work next too.

References

1. Patterson R.D, Robinson K, Holdsworth J,et al.: Complex Sounds and Auditory Images. In Auditory Physiology and Perceptionm (1992) 429-446
2. Meddis R.: Simulation of Mechanical to Neural Transduction in The Auditory Receptor. Journal of the Acoustical Society of America, **79** (1986) 702-711
3. Hewitt, M.J, Meddis, R.: Implementation Details of a Computation Model of the Inner Hair-Cell/Auditory-Nerve Synapse. Journal of the Acoustical Society of America, **87** (1990) 1813-1816
4. Vapnik, V.: The Nature of Statistical Learning Theory. NewYork: Springer-Verlag (1995)
5. Wang, L.: Support Vector Machines: Theory and Applications. Springer (2005)
6. Minh, N.N, Jagath, C.R.: Multi-Class Support Vector Machines for Protein Secondary Structure Prediction. Genome Informatics, **14** (2003) 218-227
7. Franc, V., Hlavac, V.: Multi-class Support Vector Machine. Pattern Recognition, Proceedings. 16th International Conference, **2** (2002) 236-239
8. Kang, C.Y, Zhang, X.H, Zhang, A.Q.: A Method of Passive Sonar Target Recognition Based on Spectrum Evaluation. Journal of Harbin Engineering University, **24** (2003) 627-631
9. Kang, C.Y, Zhang, X.H, Zhang, A.Q, Lin, H.W.: Underwater Acoustic Targets Classification Using Welch Spectrum Estimation and Neural Networks. Advances in Neural Networks ISNN2004, Springer (2004) 930-935

Feature Selection and Identification of Underground Nuclear Explosion and Natural Earthquake Based on Gamma Test and BP Neural Network

Daizhi Liu, Xihai Li, and Bin Zhang

Section 602, Xi'an Research Institute of High Technology
Hongqing Town, Baqiao, Xi'an 710025, China
xihai_li@163.com

Abstract. Feature selection is a very important and difficult problem in the identification of underground nuclear explosions and natural earthquakes. To solve this, Gamma test is proposed to select a best feature set from all features of underground nuclear explosions and natural earthquakes in the sense of the smallest estimated mean-squared error between feature input and target output, and then an identification experiment based on BP Neural Network is carried on with these selected features. To show the advantages of this method, all features are also identified based on BP Neural Network, the result is that these two identification rates are almost the same, this fully indicates this feature selection and identification method can reduce the complexity of identification system, and improve the efficiency of classification.

1 Introduction

In September 1996, a treaty banning underground nuclear test was subscribed in Geneva, seismic methods were the first chosen detection methods to monitor compliance with this treaty, and this led immediately to the problem of distinguishing between the signals from underground nuclear explosions and those of natural earthquakes. In order to distinguish the two kinds of signals, there are three steps to perform, i.e., pattern collection, feature extraction and selection, and pattern classification. With the in-depth research of feature extraction methods, the features are more and more. Theoretically, the more the features, the easier the identification of the underground nuclear explosions and natural earthquakes[1]. However, the more the features, the bigger the complexity of the identification system, and some features are useless to the identification system. Therefor, feature selection must be applied to reduce the complexity of the identification system.

In this paper the feature selection method based on the Gamma test[2] is proposed to select the best feature set of underground nuclear explosions and natural earthquakes. This procedure appears to give accurate (probabilistic) estimates for the mean-squared error of the classification variable, for a wide

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 393–398, 2005.

class of feature vectors, independently of any detailed knowledge of the function from input feature space to output classification target except that it should be smooth (bounded first and second order partial derivatives).

2 The Gamma Test [2]

Let a data sample be represented by

$$((x_1, ..., x_m), y) = (X, y)$$

in which we think of the vector $X = (x_1, ..., x_m)$ as *input*, confined to a closed bounded set $C \subseteq R^m$, and the scalar y as the *output*.

We focus on the case where samples are generated by a suitably smooth function (bounded first and second order partial derivatives)$f : C \subseteq R^m - > R$ and

$$y = f(x_1, ..., x_m) + r$$

where r represents an indeterminable part, which may be due to real noise or might be due to lack of functional determination in the posited input/output relationship. The Gamma test is designed to give a data-derived estimate for Var(r).

We assume that the training and testing data are different sets in which: (a) the training set inputs are non-sparse in the input-space; (b) each output is determined from the inputs by a deterministic process which is the same for both the training and testing sets; (c) each output is subjected to statistical noise with finite variance whose distribution may be different for different outputs but which is the same in both training and testing sets for corresponding outputs.

Suppose (X, y) is a data sample. Let (X', y') be a data sample such that $|X' - X| > 0$ is minimal. Here $|.|$ denotes Euclidean distance and the minimum is taken over the set of all sample points different from (X, y). Thus X' is the nearest neighbor to X (in any ambiguous case we just pick one of the several equidistant points arbitrarily). The Gamma test is based on the statistic

$$\gamma = \frac{1}{2M} \sum_{i=1}^{M} (y'(i) - y(i))^2$$

where M is the number of input/output training pairs. It can be shown that $\gamma - > Var(r)$ in probability as the nearest neighbor distances approach zero. In a finite data set we cannot have nearest neighbor distances arbitrarily small so the Gamma test is designed to estimate this limit by means of a linear correlation.

Given data samples $(X(i), y(i))$, where $X(i) = (x_1(i), ..., x_m(i)), 1 \leq i \leq M$, let $X(N(i,p))$ be the pth nearest neighbor to $X(i)$, typically we take p in the range $20 - 50$.

We write

$$\Delta(p) = \frac{1}{p} \sum_{h=1}^{p} \frac{1}{M} \sum_{i=1}^{M} |X(N(i,h) - X(i))|^2$$

and

$$\Gamma(p) = \frac{1}{p}\sum_{h=1}^{p}\frac{1}{2M}\sum_{i=1}^{M}(y(N(i,h) - y(i))^2$$

then$\Delta(p)$ is the mean square distance of the $h \leq p$ nearest neighbors and $\Gamma(p)$ is an estimate for the statistic γ based on the $h \leq p$ nearest neighbors.

3 Feature Selection Based on Gamma Test

In this section, we use Gamma test to select the best feature set from 12 features in seismic pattern recognition of underground nuclear explosion, they are: first order correlation, first order spectrum correlation, second order correlation, second order spectrum correlation, third order correlation, third order spectrum correlation, complexity of waveform, short time spectrum features, spectrum ratio, third moment of frequency, Markel coefficients of AR Model, real cepstrum. These features are often used in seismic pattern recognition and their recognition ratios are high [4].

Using the method of attractor analysis, we find that to distinguish short period underground nuclear explosion signals and natural earthquake signals, the least number of feature vector is 6 [5]. Thus, we use Gamma test to select 6 features from those 12 features to form the best feature set.

We have 100 short period underground nuclear explosion events and natural earthquakes respectively, for each data sample of underground nuclear explosions and natural earthquakes, we extract those 12 features according to above mentioned, so the feature vector is $X(i) = (x_1(i), ..., x_{12}(i))$, and $1 \leq i \leq 100$. To the best feature set, the length of feature vector is 6 according to the method of attractor analysis , thereafter there are $C_{12}^6 = 924$ feature combinations totally. Using each feature combination vector $X(i) = (x_1(i), ..., x_6(i))$ as *input*, and the *output* is the class number to which the data sample belongs. Then calculate the following according to Gamma test:

$$\Delta(p) = \frac{1}{p}\sum_{h=1}^{p}\frac{1}{M}\sum_{i=1}^{M}|X(N(i,h) - X(i))|^2$$

and

$$\Gamma(p) = \frac{1}{p}\sum_{h=1}^{p}\frac{1}{2M}\sum_{i=1}^{M}(y(N(i,h) - y(i))^2$$

where $p = 1, ..., 30$ and $M = 80$ (the number of the training set is chosen to be 80 in this case). Finally, we can obtain a γ for each feature combination. The total values of γ are listed in Fig 1.

The feature combination to which the minimal γ belongs is the best feature vector. From Fig.1,the minimal feature combination index is 845, which stands for the best feature vector: second order spectrum correlation, third order correlation, third order spectrum correlation, complexity of waveform, Markel coefficients of AR Model, real cepstrum according to the rule of Gamma test. To validate the efficiency of this feature selection method, we use BP neural network to distinguish these two events with these selected features.

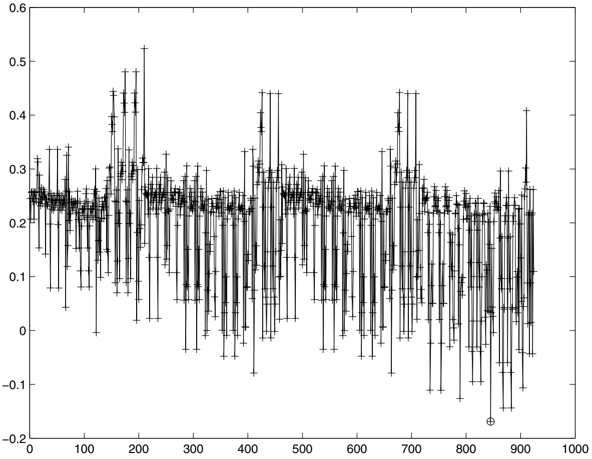


Fig. 1. γ value of all feature combinations ($'o'$ denotes the minimal γ)

4 Recognition Based on BP Neural Network

BP network is one of the widely used models. In 1989, Robert Hecht-Nielson proved that a 3-layer BP neural network can fulfill the map from n dimension to m dimension. To reduce the structure redundancy of BP network, we choose a 3-layer BP network to recognize underground nuclear explosions and natural earthquakes. The neuron number of hidden layer of BP network is always worth to study, in general case, it has essential relation with the problem to be solved, the number of training samples and the neuron number of input and output layer. In this paper, we choose the neuron number of hidden layer according to the formula of paper [3]. The formula is $J = \sqrt{I + K} + b$, where J is the neuron number of hidden layer, I the input neuron number,K output neuron number, b a constant between [1,10]. According to the experimental result of document[6], different b has little influence to the identification results, so we choose $b = 4$ arbitrary.

We use the selected 6 features as input of BP network, and output of the network is the class number to which the data sample belongs, then the neuron number of the hidden layer is $\sqrt{6 + 1} + 4 = 6.646 \simeq 7$, and the network structure is $6 - 7 - 1$.

To compare the identification results of the selected features and with that of all features, we also use all 12 features as input of the BP neural network, and output of the network is the class number to which the data sample belongs, then the neuron number of the hidden layer is $\sqrt{12 + 1} + 4 = 7.61 \simeq 8$, and the network structure is $12 - 8 - 1$.

Choose 80 samples from underground nuclear explosion set and natural earthquake set respectively to train BP neural network, and the rest 20 samples (40 samples totally) are tested as testing samples. The recognition results are listed

Table 1. Testing identification rate of the selected features

	identification rate
$N\%$	93.2
$E\%$	84.3
$T\%$	88.75

Table 2. Testing Identification of all features

	identification rate
$N\%$	93.5
$E\%$	84.3
$T\%$	88.9

in Table 1 and Table 2, (where N denotes nuclear explosion, E denotes natural earthquake, T denotes total recognition rate).

From Table 1 and Table 2, we find that these two identification result are almost same, however the first results are obtained only by using 6 features as input, this can reduce the complexity of the identification system, especially in the case of high dimension.

5 Conclusions

In this paper, we first apply Gamma test to select the best feature set of underground nuclear explosions and natural earthquakes, then we use BP neural network to distinguish these two events with the best feature set. Experimental results and the corresponding analysis provided suggest that Gamma test and BP neural network are very effective to feature selection and identification of underground nuclear explosions and natural earthquakes, and they can reduce the complexity of the identification system greatly.

Acknowledgment

This work is supported by the National Natural Science Foundation (No. 40274044).

References

1. Li, L.: Introduction to Pattern Recognition (in Chinese), High Education Press, Beijing (1994)
2. Aδalbjörn Stefásson, Končar, N., and Jones, Antonia, J.: A Note on The Gamma Test, Neural Computing and Applications, **5** (1997) 131-133
3. Zhang, L.: Model of Artifical Neural Network and Its Application(in Chinese), Fudan University (1992)

4. Liu, L.: On Seismic Pattern Recognition System of Underground Nuclear Explosions, Journal of Hi-technology of Institute of Engineering(in Chinese), **9** (1995) 1-6
5. Liu, L., Zou, R., et al.: Attractor Analysis of Seismic Pattern Recognition of Nuclear Explosion, Acta Electronica Sinica (in Chinese), **25** (1997) 122-125
6. Li, X., Zhao, K., et al.: Feature Extraction and Identification of Underground Nuclear Explosion and Natural Earthquake Based on FM^{m}let Transform and BP Neural Network, Advances in Neural Networks-ISNN2004, **3174** (2004) 925-930

An Adaptive Neural Network Classifier for Tropical Cyclone Prediction Using a Two-Layer Feature Selector

Bo Feng and James N.K. Liu

Department of Computing, The Hong Kong Polytechnic University
{csbfeng,csknliu}@comp.polyu.edu.hk

Abstract. We are in need of more accurate, automated prediction and classification methods for the determination of weather patterns all over the world, especially for the identification of severe weather patterns such as tropical cyclones (TC). They help to discover hazardous meteorological phenomena, providing an early warning to save people's lives and properties. In this paper, we propose an adaptive neural network classifier to predict the intensity of a tropical cyclone based on associated features, which is preprocessed by a two-layer feature selector. A binary trigger is used to adjust the neural network topology adaptively when necessary by controlling the validity of each hidden node. Experimental results show that our proposed classifier is a preferable one on learning speed and predictive accuracy comparing to other neural algorithms.

1 Introduction

Recently, research on artificial neural networks (ANNs) on weather prediction has been widely explored with promising achievements, including theoretical models [1], learning algorithms [2], practical applications [3], and so on. It is recognized that the simulating ability of ANNs is superior to traditional algorithms such as self-organization [4]. However, ANN still has serious limitations which are hard to be absolutely solved. Take backpropagation neural network [5] as a typical example, it is hard to determine the number of hidden units; the training time cost is large; it is difficult to support online learning; and it is easy to fall into a local minimum. Such limitations stimulate deeper research into theories and models of ANNs.

In this paper, we proposed an adaptive neural network classifier to predict the intensity of a tropical cyclone. A two-layer feature selector is designed to organize the various types of attributes of an input instance, normalized, categorized, and fed into the classifier. A binary trigger connecting to each hidden units is used to determine how many hidden units are going to be involved in the next learning loop by adopting the average summed squared error (SSE) as a threshold. This can overcome the disadvantage of manually determining the number of hidden units of most neural networks. Experimental results show that this classifier performs better than other algorithms on learning speed and predictive accuracy.

The rest of this paper is organized as follows. In section 2, we describe the details of the classifier, including network topology, learning algorithm and feature selection

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 399–404, 2005.

and trigger function. In section 3 we report on experiment results. Finally in section 4 we conclude and indicate some issues for future work.

2 Network Topology Design

Figure 1 shows a proposed adaptive neural network classifier network topology, which is composed of a two-layer feature selector and a three-layer neural network. The inputs of the feature selector are the properties of a tropical cyclone such as Time, Position, Mean Sea Level Pressure (MSLP), Speed. The output of the classifier is the intensity of that tropical cyclone, which is one of four types: Tropical Depression (TD), Tropical Storm (TS), Severe Tropical Storm (ST) and Typhoon (TY).

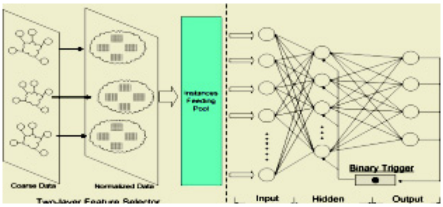


Fig. 1. Topology of the adaptive neural network classifier

2.1 Two-Layer Feature Selector

The selector performs critical tasks of normalizing input patterns and changing them into binary representations. In Coarse Data layer, each cloud represents an input tropical cyclone pattern and the correlated small circles represent the attributes of that tropical cyclone such as MSLP, speed, longitude, latitude and so on. These attributes are normalized and changed into binary form in Normalized Data layer. We use equal width bin approach to normalize each attribute, and the bin width is calculated using equation 1 shown below, where *max (att)* and min *(att)* are corresponding to the maximum and minimum value of that attribute respectively, and *bin* is the number of bins to be divided into.

$$width = \frac{\max(att) - \min(att)}{bin} \tag{1}$$

After normalization, each pattern representation is transformed into a set of binary bits. Taking attribute MSLP 100 as an example, it is supposed to be located into the 10th bin [97, 108], out of total 10 bins of MSLP. So, it can be transformed into 0000000001, where the binary number 1 on the 10th bit represents that it belongs to the 10th bin, and the number of bits accords to the number of bins. Note that only one bit can be 1 at one time. The Instances Feeding pool is used to cache enormous patterns for the classifier.

2.2 Adaptive Neural Network

After the input patterns have been normalized and well-represented, they are fed into an adaptive feedforward neural network for training and validation. A binary trigger is used to adaptively determine the number of hidden units to be validated during the training process, based on the Summed Squared Error (SSE) calculated among the output units. Its major purpose is to find the optimizing weight for every connection among the units for next step training. The activation function of the neural units is the *sigmoid* function. The number of the hidden units can be initially determined as one tenth of the number of input units empirically.

Assuming that instances of the patterns input to the input layer units are $A_k=(a_k^1 , a_k^2 , ..., a_k^n)$ $(k = 1,2,..., m)$, where k is the index of the instance, and n is the number of the input unit. The hidden layer unit j computes its activation values as below:

$$o_j = f(\sum_{i=1}^{n} a_k^i w_{ij} - \beta_j) \tag{2}$$

where β_j is the bias of the j^{th} unit of the hidden layer, a_k^i is the input instance, w_{ij} is the feed-forward weight connecting hidden unit j to input unit i, and f is the sigmoid function in Equation 3. Note that w_{ij} is randomly distributed in [0.0, 1.0].

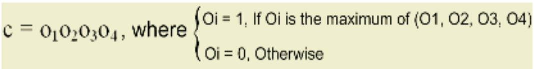
Next we apply a winner-take-all style competition in the output layer to determine the classification result for the input pattern. The activation function of the output layer is computing using equation 4:

$$f(net) = \frac{1}{1 + e^{-net}} \tag{3}$$

$$o_l = \alpha_j * f(\sum_{j=1}^{n} o_j w_{jl} - \beta_l) \tag{4}$$

where α_j is a binary value derived from the binary trigger, o_j is the activation value of the hidden unit $j,$ w_{jl} is the feed-forward weight connecting unit j to output unit o, andβ_o is the bias of unit $o.$ α_j is initialized to 1 for all j to enable all hidden units. Ifα_j is set to 0, it means that the binary trigger determines to disable hidden unit j, by setting a zero value for all activation values of output units connecting to hidden unit j. Then unit j will not contribute in the successive training process, while other hidden units will continue to work.

After the neural network is well-rounded training for a given pattern, the unit of the output layer which has the largest activation value will be considered as the category that the input pattern is classified. Or, we can use Equation 5 to represent it. For example, if c = 0010, then the input pattern is classified into the third class, as the third bit is set to 1.



$$c = o_1 o_2 o_3 o_4 \text{, where} \begin{cases} \text{Oi = 1, If Oi is the maximum of (O1, O2, O3, O4)} \\ \text{Oi = 0, Otherwise} \end{cases} \tag{5}$$

$$Err = \frac{1}{N} \sum_{l=1}^{N} (d_l - d_l^k)^2 \tag{6}$$

After the process of training a single input pattern has been completed, the error between real network output and expected output is computed using the average summed squared error:

Where N is the number of the output units, d_l is the real output of the output unit k and d_l^k is its expected output. If the summed square error Err is within a predefined rangeΨ, it means that the current pattern can be characteristically recognized and classified by the neural classifier. Then the neural network gives out a stimulus signal and feeds it back to the hidden layer and then to the input layer, adjusting the weight for every two connecting units by equation 7.

Where η is learning rate and experiments show that η=0.1 could achieve preferable results , δ is the error signal between units l and k in two connecting layers, and o_l is the activation value for unit l. On the other hand, if Err is beyond the

$$w^k{}_{i+1} = w^k{}_i + \eta * \delta_{kl} * o_l \tag{7}$$

predefined rangeΨ, it means the current pattern's characteristics are not satisfied for the neural classifier. We need to find out whether to delete this pattern from the training set, or adjust the structure of the neural network to cover this pattern. If we

$$Err_h = \frac{1}{N} \sum_{l=1}^{N} (w_{hl} - d_l)^2 \tag{8}$$

choose the latter, we calculate the hidden units' SSEs to determine their validation status for next training loop. We use equation 8 to find out the hidden units' SSEs.

Where N is the number of output units, w_{hl} is the feed-forward weight connecting unit h to output unit l in the output layer, and d_l is the expected output of unit l. Then the unit with the maximum Err_h among all units of the hidden layer is selected, which means that this unit contributes most to calculate the error of unit connecting to unit l of the output layer. It satisfies equation 9, where M is the number of hidden units.

After we have selected the hidden unit with the largest error, the binary trigger changes the binary valueα_j of that unit from 1 to 0, thus equation 4 gets a zero. So this hidden unit is disabled in the successive training. The process from equations 4 to

$$s = \max_{h=1}^{M} (Err_h) \tag{9}$$

9 will be repeated until the training SSEs for that given input pattern satisfies the criteria. If the number of the hidden units is reduced to a predefined threshold, no hidden units will be disabled furthermore, and if the training SSE is still out of the rangeΨ, this pattern will be discarded at last.

As the binary trigger is to find the optimizing weights for each connection during the training for a single given pattern, when a new pattern is fed into the neural classifier, the binary trigger is going to be reset to all 1's to enable all hidden units.

3 Experimental Results and Analysis

To show that our proposed approach is a preferable one on learning speed and predictive accuracy comparing to other neural networks, we carry out a set of experiments in which a static neural network is adopted to compare with our adaptive neural network. Two networks share the same structure except the number of hidden units. We have collected 2,597 samples of 268 tropical cyclones passing through Hong Kong from January 1999 to December 2004 from Hong Kong Observatory (HKO) [6]. These samples are time series distributed and recorded every 6 hours during their lives. 7 attributes are concerned with every pattern and they are: Name, Time, Intensity, Latitude, Longitude, Mean Sea Level Pressure (MSLP), and Speed. Before we feed these attributes (except Intensity as it is to be predicted) into the designed classifier, we normalize them into binary representations. Table 1 shows the normalization result and the number of units for every network layer.

Table 1. Input attributes and initial number of layer units

Input Pattern				**Initial number of**		
				Input units	**Hidden units**	**Out-put units**
Attribute	**No.**	**Example**	**Representation**			
Time	*Year (5) x Quarter (4) = 20*	*2003.12*	*100…0 (nineteen 0s)*	*140*	*14*	*4*
Position	*Longitude (10) x Latitude (10) = 100*	*N_{ij}^{th} block, i=20,j=20*	*100…0 (ninety-nine 0s)*			
MSLP	*Bin(10) x 1 = 10*	*109*	*100…0 (nine 0s)*			
Speed	*Bin(10) x 1 = 10*	*136*	*100…0 (nine 0s)*			

Initially we set the number of hidden units to 14 for both static and adaptive neural network. We select 2,000 pattern samples out of 2,597 as the training set and the rest as the testing set. In our experiments, two typical learning functions, Backpropagtaion and Backprop-Momentum, are adopted to train both neural networks, with the same set of parameters. A computer with 2.26 GHz Intel Pentium CPU and 512 RAM is used for the simulations. All other windows applications are shut down to ensure the most precise time-cost. Table 2 shows the testing results for both approaches.

Table 2. Testing results for static classifier and neural classifier

Classifier	**Learning Functions**	**Training Accuracy**	**Testing Accuracy**	**Training Time (s)**	**Testing Time (s)**
Static Classifier	**Backpropagtaion**	100 %	91.3 %	1,235	13
	Backprop-Momentum	100 %	87.6 %	1,469	11
Adaptive Classifier	**Backpropagtaion**	100 %	94.7 %	1,113	9
	Backprop-Momentum	100 %	90.1 %	1,411	9

From Table 2, we can observe that our proposed adaptive classifier is preferable than the static one, and it achieves a higher accuracy with the less time consumption. Figure 2 shows the error graphs for learning functions respectively, from which we also can find that curves c and d converge to the learning cycles axis faster than curves a and b, which means that training process of c and d will complete faster than a and b.

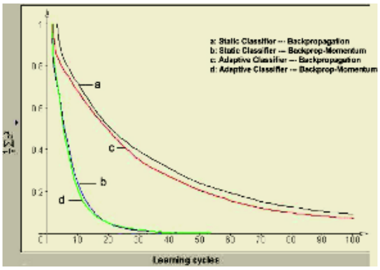


Fig. 2. Error graphs for respective learning functions

4 Conclusion and Future Work

In this paper, we have proposed an adaptive neural network classifier to predict the intensity of a tropical cyclone based on associated features normalized by a two-layer feature selector. A binary trigger is used to adjust the network topology by disabling the hidden nodes using the average summed squared errors for every node. Experimental results show that this adaptive classifier is a preferable one on learning speed and predictive accuracy over a conventional static classifier.

In the future, we need to find a more efficient and accurate criteria for determine the node disability, as well as a more efficient representation for the input pattern. Also, as temporally we can only disable the number of hidden nodes, finding a way to make the binary trigger more flexible to control the validation of the hidden nodes is another critical research task in our future work.

Acknowledgements

The authors would like to acknowledge partial support through the research grant CERG B-Q515 and departmental research grant H-ZJ87 of The Hong Kong Polytechnic University.

References

1. Lee, R.S.T., Liu, J.N.K.: An Automatic Tropical Cyclone Pattern Recognition and Classification System Using Composite Neural Oscillatory-Based EGDLM. Journal of Fuzzy Systems **4** (2002) 616-625
2. Liu, J.N.K., Kwong, R.W.M., Feng B.: Chart Patterns Recognition and Forecast Using Wavelet and Radial Basis Function Network. International Conference on Knowledge-Based Intelligence Information & Engineering Systems, (2003) 564-571
3. Feng, B., Liu, J.N.K.: Semo-Mamo, A 3-Phase Module to Compare Tropical Cyclone Satellite Images Using a Modified Hausdorff Distance. The International Conference on Machine Learning and Cybernetics, (2004) 3808-3813
4. Shah-Hosseini, H., Safabakhsh, R.: TASOM: A New Time Adaptive Self-Organizing Map. IEEE Transactions on Systems, Man and Cybernetics. **33** (2003) 271-282
5. Hecht-Nielsen, R.: Theory of the Backpropagation Neural Network. International Joint Conference on Neural Networks, **1** (1989) 593-605
6. Hong Kong Observatory, HKO, http://www.hko.gov.hk/contente.htm

Feature Point Matching of Affine Model Images Using Hopfield Network

Jinsi Tian and Jianbo Su

Department of Automation, Shanghai Jiao Tong University
Shanghai 200030, China
jstian@sjtu.edu.cn

Abstract. This paper presents an approach to match feature point of a pair of 3-dimensional affine model images. The affine transferring parameters are computed by a set of corresponding feature points, which are obtained based on 2D Hopfield neural network. The design of energy function of the neural network optimizes the matching error of the feature points. Two affine geometric constraints, epipolar and homography are used without the restriction to scene's particularity. A pair of affine model images tests the performance of the proposed method.

1 Introduction

Hopfield network is an excellent tool for optimization computation when a problem exits many solutions by searching the minimal value of the designed objective function, which acts as the energy function in the network. Feature matching problem is actually an optimization problem [1]. Since affine camera is a well-conditioned approximation it is wide used in industrial object matching and model-based object recognition [2]. Feature point matching is a main instrument to obtain affine transforming parameters, from which two-view images corner matching is the most common one [3]. Much attention is attached on the parallel binocular feature matching, which is hard to achieve of the stereo rig calibration [4]. The matching veracity decides the estimation correctness of the affine transforming matrix's elements. However, due to the occlusions, it's a crucial approach to match the corners precisely as in the perspective cases. Peng [5] used 4-order Hopfield neural network to point correspondence problem for affine invariant matching. The neighborhood information is used to reduce the network's dimension. Koenderink and Van [6] established the foundation of using two affine cameras resume the affine reconstruction. The affine geometrical information is adopted and the structure is resumed from the objects' motion under the smooth deformational assumption.

As affine camera captures the images, this special kind of case provides much simplification compared to the usual perspective cameras. However, the images suffer from noises and stereo rig has a wide baseline in industrial model matching. Traditional matching methods such as correlation or using smoothness constraint are not

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 405–410, 2005.

suitable for this kind of problem. Homography is an important geometry constraint besides epipolar. We add the homography constraint to the wide used epipolar constraint to simplify the searching progress as in perspective case [7]. The CHNN (Continues Hopfield Neural Networks, CHNN) is used to optimize the matching error under the two geometry constraints. Compared with the gray level similarity matching method it leads to matching results robust to noises and hasn't any restriction to special scenes.

We match the affine models by a set of corresponding points. The point pair set computes the affine transform parameters including rotation, translation, scaling and shearing. The transformed 3-dimensional model matches the camera model well by the computing affine parameters as what the experimental results show.

The rest of this paper is organized as follows: Section 2 presents the affine geometric constraints. Section 3 gives the energy function of Hopfield type of neural network and the matching process. Section 4 gives experimental results with a pair of affine model, followed by a conclusion of our research.

2 Affine Geometry Constraints

Similar to geometry constraints for feature matching of perspective image, affine image pair satisfies affine epipolar constraint. Given a pair of images of an object model from two views by an affine camera, there is a unique affine fundamental matrix F_A, which is of size 3×3 and its format is shown in formula (1). Denote a pair of corresponding points (p, p'), the affine epipolar constraint provides [2]:

$$p'^T F_A p = 0, \text{ and } F_A = \begin{bmatrix} 0 & 0 & a \\ 0 & 0 & b \\ c & d & e \end{bmatrix}. \qquad (1)$$

At least four pairs of correspondences are needed for computing the parameters of F_A. Besides the epipolar, affine homography constraint is another geometry constraint regardless the scene's particularity. Given three arbitrary noncollinear correspondences, they are connected by a 3×3 matrix H_A with 6 dof. The affine transformation matrix can be confirmed by the following formula [2]:

$$p' = H_A p, \text{ and } H_A = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & 1 \end{bmatrix}. \qquad (2)$$

For a pair of affine images there is a special relationship between the two geometry constraints. Three noncollinear points can define a plane, and when a forth noncoplanar correspondence is added, we can compute the epipolar line $l'(l_1', l_2', 1)$ of the forth point pair on the second image. From the four pairs of correspondences the affine fundamental matrix can be given as:

$$\begin{aligned} F_A &= [(-l'_2, l'_1, 0)^T]_\times H_A, \\ \text{and } l' &= ((l'_1, l'_2, l'_3)) = (H_A p_4) \times p'_4. \end{aligned} \tag{3}$$

3 Corner Matching Using CHNN

Number the feature points in the left image from 1 to m, and the feature points in the right image from 1 to n. We use binary neural to present the affine matching problem. The structure of two-dimensional Hopfield Neural Network for feature point matching of a pair of affine images is of size m×n. The output status of the neural v_{ix} represents the matching state of point No. i, and No. x. If i matched x, v_{ix} will be set to 1, else v_{ix} will be set to 0.

We define the error function as the computation error of affine fundamental matrix. Four correspondences, which are not coplanar, act as a group and compute an F_A. However, the feature point is not precise and the elements of F_A are not robust. If all the correspondences are correct, the difference of F_{Ai} from each group should be a zero vector. Therefore, the error function of CHNN is given as following:

$$\begin{aligned} E = {} & \frac{A}{2}\sum_{i=1}^{m}\sum_{x=1}^{n}\sum_{y=1,y\neq x}^{n} v_{ix}v_{iy} + \frac{B}{2}\sum_{x=1}^{n}\sum_{i=1}^{m}\sum_{j=1,j\neq i}^{m} v_{ix}v_{jy} - \frac{C}{2}\sum_{i=1}^{m}\sum_{x=1}^{n} v_{ix} \\ & + \frac{D}{2}\sum_{i=1}^{m}\sum_{x=1}^{n}\sum_{j=1,j\neq i}^{m}\sum_{y=1,y\neq x}^{n} f_{ixjy} v_{ix} v_{jy} \end{aligned}. \tag{4}$$

The first and the second terms in formula (4) guarantee that only one neural' state in a row or in a column is 1. The third term is added to avoid all the neural in the network are zeros. The forth one is matching error under the affine epipolar constraint and affine homography constraint. While matching by using CHNN, the degenerate case should be considered, together with formula (1)~(3) we have:

$$\begin{aligned} h_A &= \begin{cases} 0^T, \text{ if the points are collinear} \\ h((p_i, p'_x), (p_j, p'_y), \mathrm{adj}(p_{i,j}, p'_{x,y})), \text{ otherwise} \end{cases}, \\ f_A &= \begin{cases} 0_5^T, \text{ if } H_A \times \mathrm{adj}(p_{j,i}, p'_{y,x}) = 0 \text{ or } h_A = 0^T \\ f(H_A, \mathrm{adj}(p_{j,i}, p'_{y,x})), \text{ otherwise} \end{cases}. \end{aligned} \tag{5}$$

Where $h_A=(H_A(1,:)\ H_A(2,:))^T$, and $f_A=(a, b, c, d, e)^T$. If the outer production between the forth point pair and H_A is a zero vector, the matched points are all the correct correspondences and we set $f_A=0^T$. adj(p,p') means the adjacent matching point pair in the network, and we use the nonzero neural in left column of v_{ix} and v_{jy}. When f_{ixjy} is computed, a third adjacent neural is added to compute another f_A. The vector inner product of the two vectors' difference should be minimized. f_{ixjy} can be computed as following:

$$f_{ixjy} = \left\| (f_A^{ix})_1 - (f_A^{jy})_2 \right\|_2 . \tag{6}$$

The energy function of standard two-dimensional network is depicted as follows [1]:

$$E = -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{x=1}^{n}\sum_{y=1}^{n} T_{ixjy} v_{ix} v_{jy} - \sum_{i=1}^{m}\sum_{x=1}^{n} I_{ix} v_{ix} . \tag{7}$$

T_{ixjy} is the connection weight between neural v_{ix} and v_{jy}, and I_{ix} is the bias input of each neural. Compared formula (4) with formula (7), we can give the network's parameters.

$$\begin{gathered} T_{ixjy} = -A\delta_{ij}(1-\delta_{xy}) - B\delta_{xy}(1-\delta_{ij}) - \frac{C}{n} - Df_{ixjy}(1-\delta_{ij})(1-\delta_{xy}) \\ I_{ix} = C \qquad \text{and } \delta_{ij} = \begin{cases} 1 & (i = j) \\ 0 & (i \neq j) \end{cases} . \end{gathered} \tag{8}$$

The dynamic equation of the network v_{ix} is [1]:

$$u_{ix}(t+1) = u_{ix}(t) + [\sum_{i=1}^{m}\sum_{x=1}^{n} T_{ixjy} v_{jy} - \frac{u_{ix}(t)}{\tau}]\Delta t \tag{9}$$

We set the transfer function of each neural to sigmoid function. The characteristic of transfer function relies on three things: first it should have limitation; second it should increase monotonously and it should be continuous and differentiable. Take formula (8) to (9) and when the network is convergent to a stable state, the neurals' statuses give the matching results. The convergence of the network's energy function can be obtained as the method presented in reference [5]. Thus the algorithm of the CHNN for matching a pair of affine model images can be summarized as follows:

- Extract the corners on the edge lines and number each point in the two images respectively.
- Set the initial state of the network and update the state of the network until a stable status is obtained.
- Compute the parameters of the affine transform by the matched corners and match the model images by the affine parameters.

During the energy computation in the network, an important note is put on the degenerate case while select the three pairs for homography matrix. Because the edges are used, the points should better be selected on different lines, such as two in a same line and one in the other edge; or the three points in three different lines. Such operation can avoid degenerate case effectively.

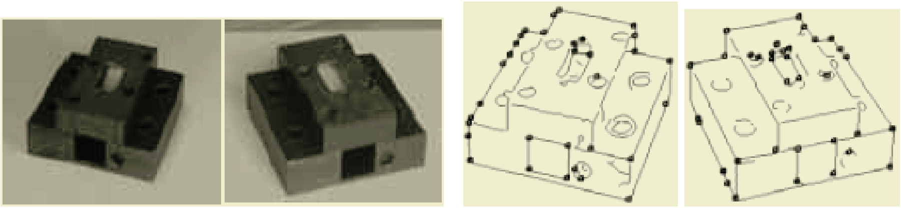
4 Experiment

The experiment uses a pair of model images by an affine camera from two views. The original images come from: http://rsise.anu.edu.au/~hartley/My-Papers.html. The feature points are extracted along the model edges with high curvature. The edges are detected by gray level segmentation and fitted to lines and the feature is selected on the edge lines. Some of the corners are occluded on the image pair. Fig. 1 (a) shows the original affine image pair of the model. Fig. 1 (b) shows the binary image pair of the edges with corners imposed on. During the matching process the network parameters in formula (8) and (9) are set as: A=1, B=1, C=1, D=1, $\tau = 1$, and $\Delta t = 0.0001$. The 2D Hopfield network is convergent after 30 iterations. The correspondences are shown in fig. 2. The residual error is 0.207.

Affine image matching has important applications on industrial model recognition. The affine transform parameters can be obtained by the correspondences. Once the point set is known, the affine transfer satisfies: $p' = Ap + t$. A and t are the coefficient matrices. Fig.3 shows the model from several views. In figure3, (a) gives the model by an affine camera from the left view. Fig.3 (b) is the transferred model by using the computed affine parameters. Fig.3 (c) is the edge model and the corners after the corner matching process. Fig.3 (d) is the matching error distance of the corners between the original image and the transferred image. From the figure we show that the estimated model and the original model image accords well with some occlusions.

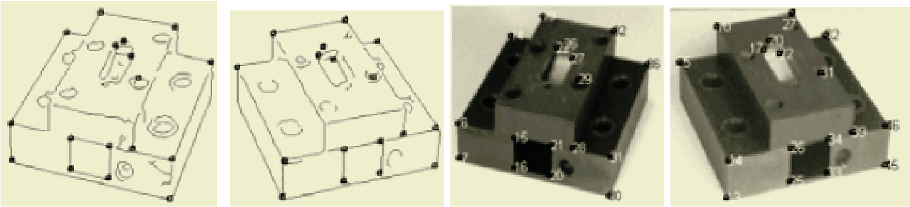
5 Conclusions

In this paper, a static stereo matching method of affine images by using CHNN is presented. The two-view feature point-matching problem of affine images is boiled down to an optimal problem and is realized by two-dimensional continuous Hopfield neural network. Two geometric constraints: affine epipolar constraint and affine homography constraint are used to establish the energy function by the rule that the computing error of affine fundamental elements is minimized. Experimental results show that the method is effective even with occlusions caused by wide baseline. Our work is suitable for industrial implemental recognition. We also give the affine transferring model matching results from multiple views by an uncalibrated affine camera, which lead to a high matching precision.



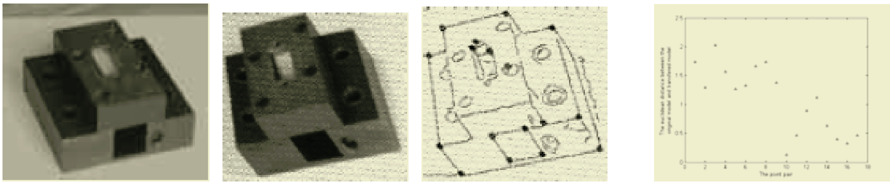
(a) (b)

Fig. 1. The original model image pair and the features (a) A pair of affine model images, (b) The corners imposed on the edges



(a) (b)

Fig. 2. The matching results (a) Correspondences of the edge model image pair (b) Correspondence of the model images.



(a) (b) (c) (d)

Fig. 3. Matching results of the transformed model against the model image (Affine parameters are computed by the point pairs) (a) The original image of the model with $\theta = 0^\circ, t = 0^T$; (b) The transferred model of the left image with the parameters: $A = \begin{bmatrix} 0.8082 & 0.6203 \\ -0.3765 & 0.8352 \end{bmatrix}$, $t = \begin{bmatrix} -0.3765 \\ 86.6797 \end{bmatrix}$; (c) The edge model with corners; (d) The matching error of the affine model images.

References

1. Hopfield, J.J., Tank D.W.: Neural Computations of Decisions in Optimization Problems. Biol. Cybern., **52** (1985) 141-152
2. Hartley, Richard, Zisserman, Andrew.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge, UK (2002)
3. Shapiro, L.S., Zisserman, A., Brady, M.: 3D Motion Recovery Via Affine Epipolar Geometry. The International Journal of Computer Vision, **16** (1995) 147-182
4. Arbter, K., Snyder, W.E., and Bukhardt, H., et al.: Application of Affine-Invariant Fourier Descriptors to Recognition of 3-D Objects. IEEE Transaction on Pattern Analysis and Maching Intelligence, **12** (1990) 640-647
5. Peng, M.K., Gupta, N.K.: Occluded Object Recognition by Hopfield Networks. Proc. IEEE Int. Conf. Neural Networks, **7** (1994) 4309-4315
6. Loenderink, J.J., and van Doorn, A.J.: Affine Structure from Motion. J. Opt. Soc. Am. A, **8** (1991) 377-385
7. R.Chung, Jianbo Su: Stereo Vision for Curved Surface without Using the Smoothness Constraint. Proc. the 3rd Conf. Asian Control, Shanghai, China (2000) 2839-2845

Nonlinear System Modeling Using Wavelet Networks

Seda Postalcioglu[1] and Yasar Becerikli[2,*]

[1] Technical Education Faculty, Electronic-Comp. Educ. Department
Kocaeli University, Izmit, Turkey
pseda@kou.edu.tr
[2] Computer Engineering Department, Kocaeli University, Izmit, Turkey
becer@kou.edu.tr

Abstract. In this paper we examine modeling of a nonlinear system using wavelet networks. Wavelet networks are similar to neural networks for the structure and the learning approach. But training algorithms for wavelet networks require a smaller number of iterations when compared with neural networks. Also interpretation of the model with neural networks is so hard. Gaussian based mother wavelet function is used as an activation function. Wavelet networks have these parameters; dilation, translation, and weights. Wavelets are rapidly vanishing functions. For this reason heuristic procedure has been used. Selecting initial values of weights are made randomly. Then parameters are optimized during learning. To update parameters, gradient method has been applied by using momentum. Quadratic cost function is used for error minimization. Two test data have been used for the simulations. One of them is a static function and the other one is a second order nonlinear function.

1 Introduction

We study a nonlinear static and dynamic system modeling with wavelet networks. Recently wavelet networks have been used as an alternative of the artificial neural networks because of interpretation of the model with neural networks is so hard [1]. On the other hand training algorithms for wavelet networks require a smaller number of iterations when compared with neural networks [2].

The wavelet network is an approach for system identification in which nonlinear functions are approximated as the superposition of dilated and translated versions of a single function [1]-[3]. Wavelet network uses a wavelet like an activation function. The structure of a wavelet network is shown in Fig. 1 [4].

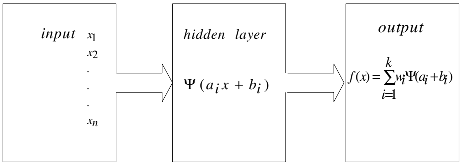


Fig. 1. The structure of a wavelet network as the one of the neuron

[*] Corresponding Author.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 411–417, 2005.

Wavelets show local characteristics which is the main property in both space and spatial frequency [5]. Families of wavelet functions especially wavelet frames are universal approximators for identification of nonlinear systems. Wavelet networks have been used both for static modeling [1],[6] and for dynamic modeling [7]. The parameters of wavelet networks are dilation (d), translation (m), bias a_0, and weights (a,c). Parameters are optimized during learning phase. Gradient method has been used for the optimization of parameters. The continuous dynamic version of the wavelet networks called "dynamic wavelet networks-DWN" is the other approach to modeling and control of the systems using wavelets [11],[12].

2 Structure of Wavelet Networks

Wavelet functions can be classified with two categories. These are orthogonal wavelet and wavelet frames. Wavelet frames are used for application of function approximation and process modeling due to the orthogonal wavelets can not be expressed in closed form [8]. Wavelet frames are constructed by mother wavelet (see Fig. 2).

A wavelet $\Phi_j(x)$ is derived from its $\phi_z(x)$ mother wavelet:

$$\Phi_j(x) = \prod_{k=1}^{N_i} \phi(z_{jk}) \tag{1}$$

$$z_{jk} = \frac{x_k - m_{jk}}{d_{jk}} \tag{2}$$

where N_i, is the number of inputs, N_w is a layer of multidimensional wavelets. The network output y is computed as:

$$y = \Psi(x) = \sum_{j=1}^{N_w} c_j \Phi_j(x) + a_0 + \sum_{k=1}^{N_i} a_k x_k \tag{3}$$

Where a_0, a, c are adjustable parameters of wavelet networks

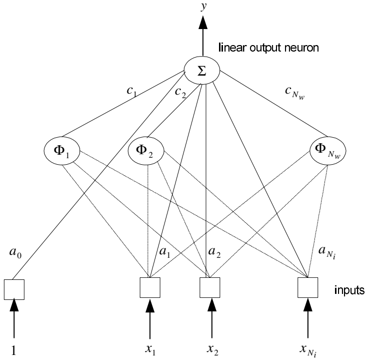


Fig. 2. Wavelet network structure

2.1 Mother Wavelet Function

The term mother wavelet gets its name due to two important properties of the wavelet analysis. The term wavelet means a small wave. The term mother implies that the functions with different region of support which are used in the transformation process. They are derived from one main function, or the mother wavelet. In other words, the mother wavelet is a prototype for generating the other window functions [10]. Mother wavelet function gives an efficient and useful description of the signal of interest [9]. This function has universal property [7]. In this study the first derivative of a Gaussian function has been used as a mother wavelet;

$$\phi(x) = -xe^{-\frac{1}{2}x^2} \tag{4}$$

3 Static Modeling Using Wavelet Networks

The main purpose is to update the parameters during the training phase. θ is the set of adjustable parameters.

$\theta = \{m_{jk}, d_{jk}, c_j, a_k, a_0\}$ j=1...N_w, k=1...N_i

θ is to be estimated by training so that equation (3) approximates the unknown function f on the domain defined by the training set. The static behavior of the process is can be described as y(p)=f(x) (see Fig. 3). f is an unknown nonlinear function. The training is based on the minimization so that quadratic cost function has been used.

$$\mathbf{j}(\theta)=\frac{1}{2}\Sigma(y_p - y\ )^2 \tag{5}$$

where y is the network output The minimization is performed by iterative gradient based methods. The partial derivative of the cost function with θ is,

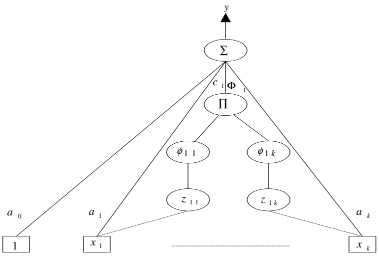


Fig. 3. Static wavelet network architecture for N_w=1

$$\frac{\partial j}{\partial \theta} = -\Sigma e \frac{\partial y}{\partial \theta} \tag{6}$$

Partial derivative w.r.t. parameter a_0, direct connection parameters, weights, translations, dilations, respectively;

$$\frac{\partial y}{\partial a_0} = 1, \ \frac{\partial y}{\partial a_k} = x_k \quad \text{k=1 to N}_\text{i}, \ \frac{\partial y}{\partial c_j} = \Phi_j(x) \qquad \text{j=1 to N}_\text{w}$$

$$\frac{\partial y}{\partial m_{jk}} = \frac{\partial y}{\partial \Phi_j} \frac{\partial \Phi_j}{\partial \phi_{jk}} \frac{\partial \phi_{jk}}{\partial z_{jk}} \frac{\partial z_{jk}}{\partial m_{jk}}, \ \frac{\partial y}{\partial d_{jk}} = \frac{\partial y}{\partial \Phi_j} \frac{\partial \Phi_j}{\partial \phi_{jk}} \frac{\partial \phi_{jk}}{\partial z_{jk}} \frac{\partial z_{jk}}{\partial d_{jk}}$$

For parameters update, equation (7) has been used as below.

$$\theta_{t+1} = \theta_t + \mu(-\frac{\partial j}{\partial \theta}) \tag{7}$$

θ_t shows vector of parameters which is used for square error. θ_{t+1} shows update vector of parameters which is used for decrease of square error. μ is the learning rate.

In this study we have used momentum which increases the learning speed without an oscillation.

$$\theta_{t+1} = \theta_t + \mu(-\frac{\partial j}{\partial \theta}) + \alpha[\theta_t - \theta_{t-1}] \tag{8}$$

α (interval is [0,1]) is the momentum coefficient in equation (8).

3.1 Initialization of The Network Parameters

The initialization values of the all parameters may make some wavelets too local and make the components of the gradient of the cost function very small in areas of interest [7]. Selecting initial values of dilation (d_{jk}), and translation (m_{jk}) randomly may be unsuitable for process modeling. Because of wavelets are rapidly vanishing functions [8]. In this study the vector m of wavelet j at the center of parallelepiped defined by the Ni intervals $\{[\alpha_k, \beta_k]\}$;

$$m_{jk} = \frac{1}{2}(\alpha_k + \beta_k) \tag{9}$$

The dilation parameters are initialized to the value;

$$d_{jk} = 0.2(\beta_k - \alpha_k) \tag{10}$$

These initializations guarantee that as shown in equation (9-10), the wavelets extend initially over the whole input domain. The choice of the weights (a,c) is less critical. They are initialized to interval (0,1).

3.2 Stopping Conditions for Training

Parameters of the wavelet networks are training during learning phase for approximation the desired function. Gradient methods have been applied for adjustable parameters. When variation of gradient and variation of parameters reaches a lower bound or the number of iterations reaches a fixed maximum, training is stopped.

4 Dynamic Modeling Using Wavelet Networks

System identification problems can be classified under three groups; parallel, serial-parallel and reverse system identification models. In this study we have applied serial-parallel identification model as shown in Fig. 4.

This structure doesn't use feedback. Real system outputs have been used for prediction of the future system outputs. So that stability and approximation of the network is guaranteed.

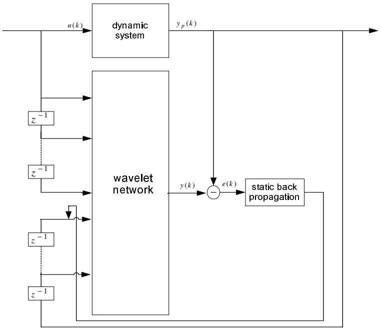


Fig. 4. Dynamic system modeling with wavelet network

5 Simulation Results

Static System Simulation

For the static system modeling equation (11) has been used 0.

$$f(x) = \begin{cases} -2.186x - 12.864 & x \in [-10, -2[, \\ 4.246x & x \in [-2, 0[, \\ 10e^{(-0.005x-0.5)\sin(x(0.03x+0.7))} & x \in [0, 10] \end{cases} \quad (11)$$

Systems shows different characteristic for different input domains. The domain of the x data is transformed into [-1,1]. The learning procedure is applied on this domain. The number of training sequence is 200 examples which are uniformly distributed in the interval of interest.

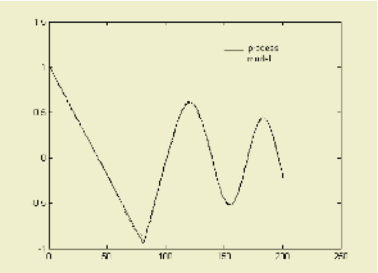


Fig. 5. Nonlinear static model and process outputs

We have obtained these results with a network of 10 wavelets. Learning iteration is 3000. Momentum was selected .09. Training Mean Square Error (TMSE) 5.6699e-004. The performance is excellent in this application (see Fig. 5).

TMSE=$\frac{1}{N}\sum_{n=1}^{N}(y_p(n)-y^n)^2$, where N is the number of inputs.

Dynamic System Simulation

For dynamic system modeling we used second order nonlinear function. The input sequence is {u(n)} and the measured process output is {y_p(n)}. As in the static case, the aim is to approximate f by a wavelet network.

$$\begin{aligned} y_p(n)&=f(y_p(n-1),y_p(n-2),\cdots,y_p(n-N_s),u(n-1),\cdots,u(n-N_e)) \\ y(n)&=f(y_p(n-1),y_p(n-2),\cdots,y_p(n-N_s),u(n-1),\cdots,u(n-N_e)) \end{aligned} \tag{12}$$

Inputs are past outputs of the process y_p and the external inputs u for equation (12). **f** is a unknown nonlinear function, which is to be approximated by a wavelet network Ψ . For the dynamic system modeling we use equation (13) [7].

$$\begin{aligned} y_p(n)&=f(y_p(n-1),y_p(n-2),u(n-1))=\frac{24+y_p(n-1)}{30}\,y_p(n-1)-0.8\frac{u(n-1)^2}{1+u(n-1)^2}\,y_p(n-2)+0.5u(n-1) \\ y(n)&=\Psi(y_p(n-1),y_p(n-2),u(n-1),\theta) \end{aligned} \tag{13}$$

The input sequence for training consists of pulses with random amplitude in the range [-5,5] and with random duration between 1 and 20 sampling periods.

Fig. 6. and Fig. 7. show that input sequence and output sequence for nonlinear dynamic system, respectively.

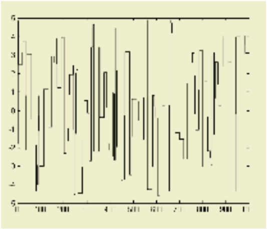


Fig. 6. Input sequence for system

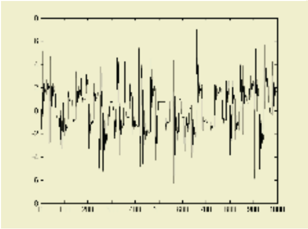


Fig. 7. Output sequence for system

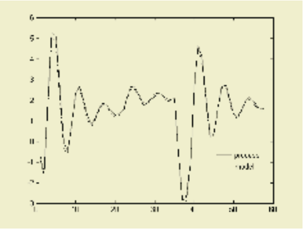


Fig. 8. Nonlinear dynamic model and process outputs

Network has 5 wavelets for Fig. 8. Learning iteration is 4000. Momentum was selected 0.3. TMSE is 3.02e-002.

6 Conclusions

Wavelet networks are an alternative network to neural networks for system modeling. Wavelets show local characteristic so the initial values of translations and dilations requires more care than the initial value of the weights. If selecting initial values of the translation and the dilation properly, training is shorter than which is modeled by neural networks. This study shows that wavelet networks can be used for system modeling and how to train this kind of networks with appropriate initialization of the translation and dilation parameters. To update parameters, gradient method has been applied by using momentum. Quadratic cost function has been used for error minimization. Gaussian based mother wavelet function has been used because generally this function has been preferred for these kinds of studies and it has universal properties.

References

1. Zhang, Q., Benveniste, A.: Wavelet Networks. IEEE Transaction on Neural Networks, **3** (1992) 889-898
2. Galvao Roberto K. H., Becerra V. M.: Linear-Wavelet Models for System Identification. IFAC15th Triennial World Congress, Barcelona, Spain (2002)
3. Zhang, Q.: Using Wavelet Network in Non parametric Estimation. IEEE trans. Neural Networks, **8** (1997) 227-236
4. Thuillard, M.: Review of Wavelet Networks, Wavenets, Fuzzy Wavenets and Their Applications. ESIT 2000, Aachen, Germany (2000)
5. Polycarpou, M., Mears, M. and Weaver, S.: Adaptive Wavelet Control of Nonlinear Systems. Proceedings of the1997 IEEE Conference on Decision and Control (1997) 3890-3895
6. Pati, Y.C., Krishnaprasad, P. S.: Analysis and Synthesis of Feedforward Neural Networks Using Discrete Affine Wavelet Transformations. IEEE Trans. On Neural Networks, **4** (1993) 73-85
7. Oussar Y., Rivals I., Personnaz L., and Dreyfus G.: Training Wavelet Networks for Nonlinear Dynamic Input Output Modeling. Elsevier neurocomputing, **20** (1998) 173-188
8. Oussar Y , Dreyfus G.: Initialization by Selection for Wavelet Network Training. Neurocomputing, **34** (2000) 131-143
9. Reza, A. M.: Wavelet Characteristics. White Paper, Spire Lab. UWM (1999)
10. Polikar R.: The wavelet tutorial. http://engineering.rowan.edu/~polikar/WAVELETS. (2001)
11. Becerikli, Y., Oysal,Y., Konar, A.F.: On a Dynamic Wavelet Network and Its Modeling Application. Lecture Notes in Computer Science (LNCS), **2714** (2003) 710-718
12. Becerikli, Y.: On Three Intelligent Systems: Dynamic Neural, Fuzzy and Wavelet Networks For Training Trajectory. Neural Computing & Applications (NC&A), **13** (2004) 339-351

Robust Modeling for Nonlinear Dynamic Systems Using a Neurofuzzy Approach with Iterative Optimization

Shirong Liu[1], Simon X. Yang[2], and Jinshou Yu[3]

[1] Institute of Electrical Engineering and Automation, Ningbo University
Ningbo, Zhejiang 315211, China
liushirong@nbu.edu.cn
[2] School of Engineering, University of Guelph, Guelph, Ontario N1G 2W1, Canada
syang@uoguelph.ca
[3] Research Institute of Automation
East China University of Science and Technology Shanghai 200237, China
jshyu@vip.sina.com

Abstract. A neurofuzzy modeling approach for nonlinear dynamic systems is proposed in this paper. An iterative optimization approach for a class of neurofuzzy systems is developed, which integrates the model structure analysis and simplification, model parameter estimation, compatible cluster merging and redundant cluster deleting, performance evaluation for neurofuzzy models. The effectiveness of the proposed modeling approach is illustrated by the Mackey-Glass chaotic time series. The simulation studies show that the parsimonious neurofuzzy model is beneficial to the robustness of model.

1 Introduction

The neurofuzzy model with the simplest fuzzy rule base (called the parsimonious neurofuzzy model) is always pursued in modeling and control for nonlinear dynamic systems. A parsimonious neurofuzzy model has three advantages. Firstly the generalizing ability of the model increases as the number of the model's rules decreases, and the over-fitting of the model can be prevented. Secondly the parsimonious model can be easily explained and understood when expert knowledge is embedded in the model. Finally, the disturbance rejection and output tracking ability of the parsimonious model are better than ones of the non-parsimonious model. In practice the robustness of model is a very significant issue in industrial applications, such as the model-based state estimation and prediction, fault detection and diagnosis, and soft sensors. The model parsimony is related to the model robustness, but the conclusion in theory has not been found.

In this paper we propose to use a general basis function neurofuzzy network (GBFNFN) based on fuzzy clustering for nonlinear system modeling. The neurofuzzy network can represent a class of feedforward neural networks and the Takagi-Sugeno fuzzy model. An iterative optimization approach to the GBFNFN structure and parameters is developed in order to reduce the model complexity, decrease the model fitting error, and enhance the robustness of model. The proposed approach integrates the fuzzy clustering in the input space, model performance evaluation, model struc-

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 418–423, 2005.

ture analysis with SVD-QR, compatible cluster merging, redundant cluster deleting, and parameter identification. The effectiveness of the proposed approach has been illustrated by the Mackey-Glass chaotic time series. The simulation studies show that the parsimonious neurofuzzy model has the advantage of good robustness, which can be employed to model uncertain nonlinear dynamic systems.

2 Framework for Model Structure Analysis

Considering a multi-input and single-output system, a generalized basis function neurofuzzy network (GBFNFN) with a single hidden layer can be defined as

$$y = f(\mathbf{x}) = \sum_{i=1}^{M} \phi_i(\mathbf{x}) f_i(\mathbf{x}) , \; \phi_i(\mathbf{x}) = \frac{\prod_{k=1}^{n} \mu_{F_k^i}(x_k)}{\sum_{i=1}^{M} \left[\prod_{k=1}^{n} \mu_{F_k^i}(x_k) \right]} , \; \sum_{i=1}^{M} \phi_i(\mathbf{x}) = 1 , \tag{1}$$

where $\mathbf{x} \in R^n$ is the input, $\phi_i(\mathbf{x})$ are the fuzzy basis functions in normalized form, $\mu_{F_k^i}(x_k)$ are the membership functions of input variables, $f_i(\mathbf{x})$ are the weighted functions from the hidden nodes to the output of the network, and M is the number of fuzzy rules. The weighted function could be a linear or nonlinear polynomial, or a constant. In practice $\phi_i(\mathbf{x})$ could be not normalized. The membership function could be a Gaussian or triangular or trapezoid or B-splines function.

It is obvious that the GBFNFN is the general form of the Takagi-Sugeno fuzzy inference system (T-S model), which is one of the major models in fuzzy modeling and control [1]. The functional equivalence between the RBFN and the T-S model or the spline-based network and the T-S model has been demonstrated in [2], [3], [4]. The cerebella model articulation controller (CMAC) can be viewed as a special basis function network. The output of the multilayer feedforward network (MLFN) with an output neuron can be abstractly defined as the sum of a series of products of a weighted coefficient (or function) and a generalized nonlinear function, which is similar to (1). If $\phi_i(\mathbf{x})$ is not normalized in (1), the outputs of the MLFN, CMAC, RBFN and B-splines network have the generalized form defined in (1).

Equation (1) can be viewed as a special case of the linear regression model [5]

$$y(k) = \sum_{i=1}^{M} \phi_i(\mathbf{x}(k)) f_i(\mathbf{x}(k)) + e(k) , \tag{2}$$

where $e(k)$ is an error signal at time k . For simplifying analysis, let $f_i(\cdot) = \bar{y}_i$, $i = 1, \cdots, M$, $\bar{y}_i$ are the constant consequent constituents. Given N input-output pairs $\{\mathbf{x}(k), y(k)\}$, $k = 1, 2, \cdots, N$, it is convenient to express (2) in the matrix form

$$\mathbf{y} = \mathbf{\Phi}\bar{\mathbf{y}} + \mathbf{e} , \tag{3}$$

where

$\mathbf{y} = [y(1), \cdots, y(N)]^T \in \mathbf{R}^N$, $\bar{\mathbf{y}} = [\bar{y}_1, \cdots, \bar{y}_M]^T \in \boldsymbol{R}^M$, $\mathbf{\Phi} = [\mathbf{\Phi}_1, \cdots, \mathbf{\Phi}_M] \in \mathbf{R}^{N \times M}$ with $\mathbf{\Phi}_i = [\phi_i(1), \cdots, \phi_i(N)]^T \in \mathbf{R}^N$, $\mathbf{e} = [e(1), \cdots, e(N)]^T \in \mathbf{R}^N$, and $N \geq M$. We will call $\bar{\mathbf{y}}$ the constant consequent vector and $\mathbf{\Phi}$ the antecedent inference matrix related to input cluster set and $\mathbf{e}$ the model residual vector. Note that each column of $\mathbf{\Phi}$ corresponds to one rule in the fuzzy rule base or one cluster in the fuzzy cluster set.

The goal of model structure optimization is to find a parsimonious model with the simplest fuzzy cluster set to balance the tradeoff between the model precision and the model complexity. The model structure analysis is based on the rank of the antecedent inference matrix $\mathbf{\Phi}$ in (3). If $\text{rank}(\mathbf{\Phi}) = r = M$, it shows that the all clusters are linearly independent of each other. If $\text{rank}(\mathbf{\Phi}) = r < M$, it implies that there are $(M - r)$ redundant clusters in the fuzzy cluster set. These redundant clusters (rules) are caused by some process noise data or redundant expert knowledge. Moreover, if there exists one column of $\mathbf{\Phi}$ and its norm is small enough, the corresponding cluster will become less important to the underlying model.

These redundant and lees important clusters can be recognized by examining the singular values of $\mathbf{\Phi}$ and positioning the columns of $\mathbf{\Phi}$. The SVD-QR with *column pivoting algorithm* was originally employed to solve the problem of *subset selection* in matrix analysis [6]. It was proposed in [7] to determine the hidden nodes and input nodes of feedforward neural networks and in [8] and [9] to extract the important fuzzy rules from a candidate fuzzy rule base. The SVD and QR can be used to compute the singular values of $\mathbf{\Phi}$ and position the columns of $\mathbf{\Phi}$ respectively.

Let r' ($r' \le r$) be the numerically determined estimate of the rank r of $\mathbf{\Phi}$. This implies that $(r - r')$ clusters corresponding to $(r - r')$ small singular values will be removed from the cluster set. Using SVD, the matrix $\mathbf{\Phi}$ can be approximated by

$$\mathbf{\Phi} = \mathbf{U}\begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix}\mathbf{V}^T = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T \approx \sum_{i=1}^{r'} \sigma_i \mathbf{u}_i \mathbf{v}_i^T , \tag{4}$$

where $\Sigma_r = diag\{\sigma_1, \cdots, \sigma_r\}$ is a diagonal matrix with $\sigma_1 \ge \cdots \ge \sigma_r > 0$, and σ_i are the singular values of $\mathbf{\Phi}$. Equation (3) can be further expressed as

$$\mathbf{y} = \sum_{i=1}^{r'} \sigma_i \mathbf{u}_i \mathbf{v}_i^T \bar{\mathbf{y}} + \mathbf{e} . \tag{5}$$

The performance of model can be evaluated by a modified statistical information criterion [9]. A neurofuzzy statistical information criterion (NFSIC) is defined as

$$\text{NFSIC}(C) = \log(\hat{\sigma}_v^2) + \frac{K(\lambda C)}{N} , \tag{6}$$

where $\hat{\sigma}_v^2$ is the estimated variance of model residuals, C is the total cluster number of the model, λ is the weighted coefficient, and K represents the coefficient of the different criteria; Akaike information criterion (AIC): $K = 2$; Bhansali-Downham information criterion (BDIC): $K = \alpha$ ($2 \le \alpha \le 5$, generally $\alpha = 4$) ; Schwarz-Rissanen information criterion (SRIC): $K = \log N$. The goal of optimization is that finding the cluster number C makes the NSFIC take minimum. The fitting degree and generalization of model can be measured by the mean-squares errors (MSE).

3 Model Iterative Optimization

The framework of iterative optimizing process for the structure and parameter identification of neurofuzzy systems are shown in Fig.1. It includes three phases, i.e., the model initializing, structure identification, and parameter identification and performance evaluation. Let the membership function be Gaussian form. The iterative optimization process is briefly explained below.

Model Initializing: Generate a set of candidate models with different cluster number. First, determine the cluster member for every candidate model in sequence (from small to large, or from large to small), and then generate a set of cluster centers and radii using Fuzzy c-mean Clustering (FCM) and the radius algorithm in input data space. Second, construct the antecedent inference matrix $\mathbf{\Phi}$, and then estimate the parameters of the consequent function vector $\overline{\mathbf{y}}$ for every candidate model. Finally, compute the output $\mathbf{y}$ and the values of the NFSIC criteria for every candidate model.

Structure Identification: Make the model structure analysis with SVD-QR. Detect the redundant, less important and compatible clusters, check the "holes" in the input space because of clustering, and then handle these clusters through the redundant and less important clusters removed from the cluster set and the compatible clusters merged , and the "holes" filled by extra clusters.

Parameter Identification and Performance Evaluation: Construct the new antecedent inference matrix $\mathbf{\Phi}$ and estimate the new consequent parameters again. Compute the output $\mathbf{y}$ and the NFSIC and MSE values for every candidate model. An optimal model is selected from the candidate models. If the model can be further simplified, the iterative process will go back to the structure identification phase and continue the optimization process.

In the iterative optimization, the effective algorithms on cluster merging and cluster filling were presented in details in [9], [10].

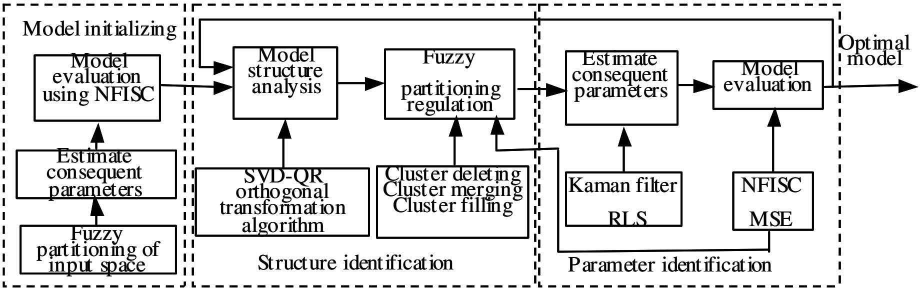

Model initializing
Model evaluation using NFISC
Estimate consequent parameters
Fuzzy partitioning of input space
Model structure analysis
Fuzzy partitioning regulation
SVD-QR orthogonal transformation algorithm
Cluster deleting
Cluster merging
Cluster filling
Structure identification
Estimate consequent parameters
Model evaluation
Optimal model
Kaman filter
RLS
NFISC
MSE
Parameter identification

Fig. 1. Framework of iterative optimization

4 Robust Modeling for Nonlinear Dynamic Systems

The neurofuzzy model for a nonlinear dynamic system is defined as

$$y_m(k+1) = \sum_{i=1}^{M} \phi_i(\cdot) f_i(\cdot) = \sum_{i=1}^{M} \phi_i(y_p(k), \cdots, \\ y_p(k-n_y+1), u(k), \cdots, u(k-n_u+1)) f_i(\cdot), \tag{7}$$

where $f_i(\cdot) = a_{i0} + a_{i1} y_p(k-1) + \cdots + a_{in_y} y_p(k-n_y) + b_{i1} u(k-1) + \cdots + b_{in_u} u(k-n_u)$, or $f_i(\cdot) = w_i$. In (7), y_m and y_p are the model and plant outputs respectively, u is the input, n_y and n_u are the orders of the output and input variables respectively. The proposed modeling approach has been evaluated by various examples. The robust modeling for the Mackey-Glass chaotic time series is given as follows.

The Mackey-Glass chaotic time series (M-G model) is one of benchmark examples for modeling method evaluation [5], described by a differential equation with time-delay

$$\frac{dy}{dt} = \frac{a\, y(t-\tau)}{1+y^{10}(t-\tau)} - 0.1 y(t), \quad y(0) = 1. \tag{8}$$

If time-delay $\tau \geq 17$, the system will cause the chaos phenomenon. Here set $a = 0.2$ and $\tau = 17$. The neurofuzzy model form for the M-G Model is selected as

$$y(t+1) = f(y(t), y(t-6), y(t-12), y(t-18)). \tag{9}$$

In practice it is difficult to build the full-process model of the M-G series only using a single neurofuzzy model. Two time-segment models are adopted, and two time intervals are $t = 0 \sim 100$ and $t = 100 \sim 2000$ respectively. The connection between two time-segment models is smoothed by a simple moving algorithm.

In $t = 0 \sim 100$, the model structure is optimized sequentially in 50-19-13-10, and $C = 10$ through four iterative steps. In $t = 100 \sim 2000$, the model structure is simplified sequentially in 24-20-14, and $C = 14$ through three iterative optimizations using the input-output data pairs from time 100 to 1100. For simplicity, the optimized model is denoted as 10/14 model. The outputs of the M-G series and neurofuzzy model in $t = 0 \sim 400$ are shown in Fig.2, respectively.

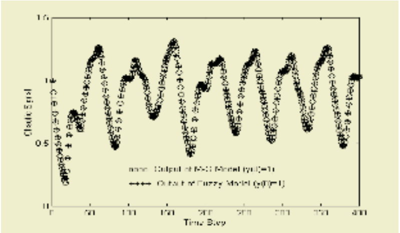


Fig. 2. Training result of model (τ=17)

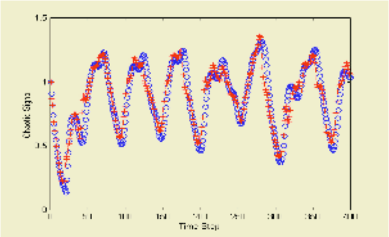


Fig. 3. Outputs of plant and model (τ=20)

It is well known that the response of nonlinear dynamic system is related to the model's parameter, order, time-delay, initial state and input signal. In practice the robustness of model is a very important issue in system modeling based on data and/or knowledge, but there is no conclusion on robust modeling in theory. In fact, a parsimonious neurofuzzy model has good robustness of model. For example, the time-delay of the M-G series is a very sensitive parameter. Consider the time-delay τ vary from 17 to 20, the output of the parsimonious model is able to track the output of the M-G series with $\tau = 20$, as shown in Fig. 3. Of course, if the time-delay further increases, the neurofuzzy model will tend to be invalidated.

5 Conclusion

The proposed neurofuzzy modeling with the iterative optimization can get a parsimonious neurofuzzy model. By simulation studies, the robustness of the neurofuzzy model with the simplest rule base can be effectively improved, which has been verified by the M-G series with the time-delay drifting. The characteristic is very important in the model-based fault detection and diagnosis, soft sensor and so on.

References

1. Takagi, T., Sugeno, M.: Fuzzy Identification of Systems and Its Applications to Modeling and Control. IEEE Trans. on Systems, Man and Cybernetics, **15** (1985) 116-132
2. Jang J.-S Roger, Sun C-T.: Functional Equivalence between Radial Basis Function Networks and Inference System. IEEE Trans. Neural Networks, **4** (1993) 156-158
3. Hunt, K.J., Haas, R., Brown, M.: On the Functional Equivalence of Fuzzy Inference Systems and Spline-based Networks. Intl. J Neural Systems (1995)
4. Hunt, K.J., Haas, R., Murray-Smith R.M.: Extended the Functional Equivalence of Radial Basis Function Networks and Fuzzy Inference Systems. IEEE Trans. Neural Networks, **7** (1996) 776-781
5. Wang, L.X., Mendel, J.M.: Fuzzy Basis Functions, Universal Approximation, and Orthogonal Least Squares Learning. IEEE Trans. Neural Networks, **3** (1992) 807-814
6. Golub, G.H., Van Loan, C.F.: Matrix Computations. 2nd ed., John Hopkins Univ. Press, Baltimore, MD (1989)
7. Kanjilal, P.P., Banerjee, D.N.: On the Application of Orthogonal Transformation for the Design and Analysis of Feedforward Networks. IEEE Trans. on Neural Networks, **6** (1995) 1061-1070
8. Mouzouris, G.C., Mendel, J.M.: Designing Fuzzy Logic Systems for Uncertain Environments Using a Singular-Value-QR Decomposition Method. In Proceedings of 1996 IEEE International Conference on Fuzzy Systems, (1996) 295-301
9. Yen, J., Wang, L.: Application of Statistical Information Criteria for Optimal Fuzzy Model Construction. IEEE Trans. on Fuzzy Systems, **6** (1998) 362-372
10. Liu, S., Yu, J.: Model Construction Optimization for a Class of Fuzzy Models. Chinese J. Computers, **24** (2001) 164-172
11. Liu, S., Yu, J. J., Lin, W., Yu, J.: Heuristic Fuzzy Cluster Learning and Its Applications in Function Approximation and Nonlinear System Modeling. Pattern Recognition and Artificial Intelligence, **16** (2003) 230-235

Modelling of Chaotic Systems with Recurrent Least Squares Support Vector Machines Combined with Stationary Wavelet Transform

Jiancheng Sun[1], Lun Yu[1], Guang Yang[2], and Congde Lu[3]

[1] College of Physics and Information Engineering, Fuzhou University
Fuzhou, Fujiang 350200, China
sunjc@mailst.xjtu.edu.cn
[2] Department of communication Engineering
Xi'an Institute of Posts and Telecommunications
Xi'an, Shaanxi, China
Sharon.yg@163.com
[3] Department of Information and Communication Engineering
Xi'an Jiaotong University
Xi'an, Shaanxi 710075, China
congdelu@mailst.xjtu.edu.cn

Abstract. A new strategy for modeling of chaotic systems is presented, which is based on the combination of the stationary wavelet transform and Recurrent Least Squares Support Vector Machines (RLS-SVM). The stationary wavelet transform provide a sensible decomposition of the data so that the underlying temporal structures of the original time series become more tractable. The similarity of dynamic invariants between the origin and generated time series shows that the proposed method can capture the dynamics of the chaotic time series effectively.

1 Introduction

Various techniques for modeling and predicting nonlinear time series are investigated in past years, such as linear regression and ARIMA models [1], local linear mode [2] and artificial neural networks [3].

In this paper, a new method combining Recurrent Least Squares Support Vector Machines (RLS-SVM) and with stationary wavelet is developed for chaotic time series reconstruction. The strategy is cutting the reconstruction into small tasks. Firstly, stationary wavelet transform is used to lay bear useful information, which is then treated by the RLS-SVM. Finally, the reconstruction is completed by the stationary wavelet inverse transform. Support Vector Machines(SVM) have become a subject of intensive study and have been applied successfully to classification tasks such as OCR and regression [4]. Least squares (LS) version of SVM can greatly simplify the problem since its solution is characterized by a linear system [5]. To deal with problems where iterative operations are necessary, Recurrent Least Squares Support Vector Machines (RLS-SVM) [6] was developed.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 424–429, 2005.

2 Embedding Phase Space of Dynamical System

Deterministic dynamical systems describe the time evolution of a system in some phase space $\Gamma \subset \boldsymbol{R}^m$. For simplicity it is assumed that the phase space is a finite dimensional vector space. A state is specified by a vector $\boldsymbol{x} \in \boldsymbol{R}^m$. Then the dynamics can be described by an explicit system of *m* first-order ordinary differential equations[7]

$$\frac{d}{dt}\boldsymbol{x}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t)), t \in \boldsymbol{R}, \boldsymbol{x} = (x_1, x_2, \ldots, x_m) \tag{1}$$

or in discrete time $t = n\Delta t$ by maps of form

$$\boldsymbol{x}_{n+1} = \boldsymbol{f}(\boldsymbol{x}_n) \tag{2}$$

A time series can be thought of as a sequence of observations $\{s_n = h(\boldsymbol{x}_n) | n = 1,2,\ldots,N_T\}$ performed with some measurement function, where N_T is the number of data points. Since the sequence $\{s_n\}$ in itself does not properly represent the multidimensional phase space of the dynamical system, some techniques are employed to unfold the multidimensional structure using available data. Takens embedding theorem guarantees the reconstruction of a state space representation from a scalar signal alone[8]. Takens state that if the sequence $\{s_n\}$ does consist of scalar measurements of the state of a dynamical system, then under certain genericity assumptions, the time delay embedding provides a one-to-one image of the original set $\{\boldsymbol{x}\}$, provided *m* is large enough. For almost all τ_d and for some m, Takens embedding theorem ensures that there is a smooth map $f : \boldsymbol{R}^m \to \boldsymbol{R}$ such that

$$s_{(n+1)\tau_d} = f\left(s_{n-(m-1)\tau_d}, \ldots, s_{n-\tau_d}, s_n\right) = f(\boldsymbol{s}_n) \tag{3}$$

Where the number *m* of elements is called the embedding dimension, the time τ_d is generally referred to as the time delay or lag. The problem of remodeling becomes equivalent to the problem of estimating the unknown function f in the embedding phase space.

To compare different dynamic modeling methods, rules such as correlation dimension and Lyapunov exponents were proposed because of their ability to describe the global properties of the attractor. It is proven that the Lyapunov exponent is a useful dynamic invariant to characterize the chaotic dynamic system. We can deduce that the system is a chaotic if at least a positive Lyapunov exponent exists. Formally, the Lyapunov exponent is defined by Wolf *et al.* [9]: given an *n*-dimensional phase space, the long-term evolution of an infinitesimal sphere is monitored. As the sphere evolves, it will turn into an *n*-ellipsoid. The *i*-th one-dimensional Lyapunov exponent is then defined in terms of the length of the resulting ellipsoid’s principal axis.

$$\lambda_i = \lim_{t \to \infty} \frac{1}{t} \log_2 \frac{p_i(t)}{p_i(0)} \tag{4}$$

The difference between strange attractors and purely stochastic (random) processes is that the evolution of points in the phase space of a strange attractor has definite structure. The correlation integral provides a measure of the spatial organization of this structure, and is given by

$$C(r) = \lim_{N \to \infty} \frac{1}{N(N-1)} \sum_{i \neq j} \Theta\left(r - \left\|\mathbf{s}(i) - \mathbf{s}(j)\right\|\right) \tag{5}$$

where Θ is the Heaviside function. Grassberger and Procaccia found that, for a strange attractor, $C(r) \propto r^{\nu}$ for a limited range of r [10]. The power ν is called the correlation dimension of the attractor. Thus, we can plot the $\log C(r) - \log r$ graph to identify an attractor.

3 Stationary Wavelet Transform

The SWT of a signal $x(k)$ is calculated according to the procedure shown in Fig. 1 [11], where $WT_x^d(j,k)$ and $WT_x^a(j,k)$, $j = 1,\ldots, j_m$, are called the detail and approximation SWT coefficients of the signal $x(k)$ at scale j respectively, j_m is the number of scales. The filters h_j and l_j ($j = 2,\ldots, j_m$) are obtained by upsampling the filters used for the previous step (i.e., h_{j-1} and l_{j-1}) by 2 respectively, and h_1 and l_1 for the first step are equal to the standard decomposition high-pass and low-pass Wavelet filters used in Mallat algorithm , respectively.

The detail coefficients $WT_x^d(j,k)$ are equivalent to the output of a bandpass filter $g_{b,j}$, with $x(k)$ as the input, where $g_{b,j} = l_1 * l_2 \ldots * l_{j-1} * h_j$ and *denotes convolution. The approximation coefficients $WT_x^a(j,k)$ are equivalent to the output of a lowpass filter g_{l,j_m} , with $x(k)$ as the input, where $g_{l,j_m} = l_1 * l_2 \ldots * l_{j_m}$, According to the time-frequency properties of wavelet transform [13], $g_{b,j}$, $j = 1,\ldots, j_m$ and g_{l,j_m} are a bank of ideal narrowband filters.

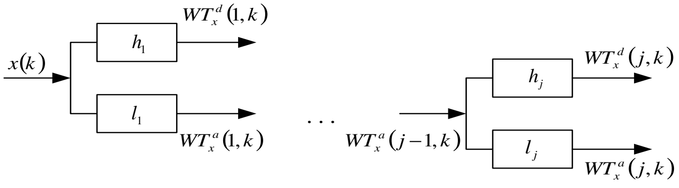


Fig. 1. Calculation algorithm of SWT.

4 Recurrent Least Squares Support Vector Machines and Learning Algorithm

The foundations of Support Vector Machines (SVM) have been developed since 1990[4]. Suykens and Vandewalle proposed the Recurrent Least Squares Support Vector Machines(RLS-SVM) to deal with problems requiring iterative operations. In following section, we formulated our algorithm based on the RLS-SVM and stationary wavelet transform:

1. generate a time series $x_i, i = 1,2,\ldots,N$, where N is the number of the data points.
2. perform SWT on the x_i, the detail and approximation coefficients at scale j are denoted by $s_{j,i}$, $j = 1,2,\ldots L$
3. choose optimal m, τ_d (scale with discrete sample units) for $s_{j,i}$,
4. constructed the trajectory matrix and build training data:

$$D = \left\{\left(\left(s_{j,k-(m-1)\tau_d},\ldots,s_{j,k-\tau_d},s_{j,k}\right),s_{j,k+1}\right) \in \boldsymbol{R}^m \times \boldsymbol{R}\right\}$$

give initial condition $\hat{s}_{j,i} = s_{j,i}$ for $i = 1,2,\ldots m$, the prediction problem is given by:

$$\begin{aligned}\hat{s}_{j,(k+1)\tau_d} &= f\left(\hat{s}_{j,k-(m-1)\tau_d},\ldots,\hat{s}_{j,k-\tau_d},\hat{s}_{j,k}\right) \\ &= w^T\varphi\left(\left[\hat{s}_{j,k-(m-1)_{\tau_d}},\ldots,\hat{s}_{j,k-\tau_d},\hat{s}_{j,k}\right]\right) + b\end{aligned} \tag{6}$$

where $\varphi(\cdot): \boldsymbol{R}^m \to \boldsymbol{R}^{n_h}$ is a nonlinear mapping in future space, $w \in \boldsymbol{R}^{n_h}$ is the output weight vector and $b \in \boldsymbol{R}$ is bias term. choose the $\varphi(\cdot)$ and estimate the function $f(\cdot)$by using training data D

5. generate the $s_{j,i+p}$ by performing the iterative process, where $p = 1,2,\ldots,l$, l is the length of reconstruction time series.
6. reconstructed time series of the system is acquired by performing the inverse stationary wavelet transform on regenerated detail and approximation coefficients.

In literature [6], Suykens and Vandewalle proposed that the eq.(6) can be convert into optimal problems which can be described as follows.

$$\min_{w,e} J(w,e) = \frac{1}{2}w^T w + \gamma\frac{1}{2}\sum_{k=m+1}^{N+m} e_k^2 \tag{7}$$

subject to $s_{j,k+1} - e_{k+1} = w^T\varphi\left(\boldsymbol{s}_{j,k} - \boldsymbol{e}_k\right) + b, k = m+1,\ldots,N+m$ (8)

where $e_k = s_k - \hat{s}_k$ $\boldsymbol{s}_{j,k} = \left\lfloor s_{k-(m-1)\tau_d};\ldots;s_{k-\tau_d},s_k\right\rfloor$ $\boldsymbol{e}_k = \left\lfloor e_{k-(m-1)\tau_d};\ldots e_{k-\tau_d};e_k\right\rfloor$

and γ is an adjustable constant. The basic idea of mapping function $\varphi(\cdot)$ is to map the data $\boldsymbol{s}$ into a high-dimensional feature space, and to do linear regression in this space.

To resolve the optimal function eqs. (7) and (8),we define the Lagrangian function

$$L(w,b,e;\alpha)=J(w,e)+\sum_{k=m+1}^{N+m}\alpha_{k-m}\times\left[h_{k+1}-e_{k+1}-w^{T}\varphi\left(\boldsymbol{s}_{j,k}-\boldsymbol{e}_{k}\right)-b\right] \tag{9}$$

where α_i are Lagrange multipliers.

The resulting recurrent simulation model is described as follows[6]

$$s_{j,(k+1)\tau_d}=\sum_{l=m+1}^{N+m}\alpha_{l-m}K\left(z_l,\left[\hat{s}_{j,k-(m-1)_{\tau_d}},\ldots,\hat{s}_{j,k-\tau_d},\hat{s}_{j,k}\right]\right)+b \tag{10}$$

where $z_l=\boldsymbol{s}_l-\boldsymbol{e}_l$

The mapping function $\varphi(\cdot)$ can be paraphrased by a kernel function $K(\cdot,\cdot)$ because of the application of Mercer's theorem[4], which mean that

$$K\left(x_i,x_j\right)=\varphi\left(x_i\right)^{T}\varphi\left(x_j\right) \tag{11}$$

with RBF kernels one employs.

5 Simulation and Results

In the following procedure, we use the laser data of the Santa Fe competition D. The data set consists of laser intensity collected from a laboratory experiment[12]

The reconstructed embedding phase space has been discussed in the Sect.2. The choice of the embedding dimension m and time delay τ_d is the first problem we faced. The methods to determine the m and τ_d have been discussed in reference [13] and [14].

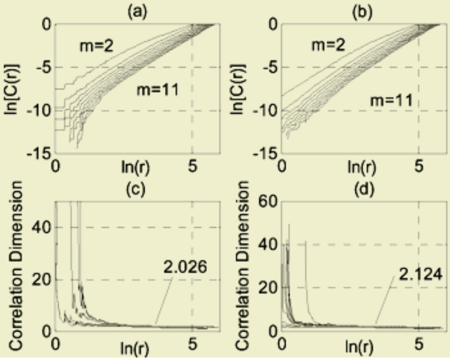


Fig. 2. Correlation dimension of Santa Fe competition estimates for embedding dimensions 2~11.(a) Correlation Integral Map (CIM) for original time series. (b) CIM for series generated from proposed method.(c) Slope (correlation dimension) estimate of CIM for original time series.(d) Slope estimate of CIM for proposed generated time series.

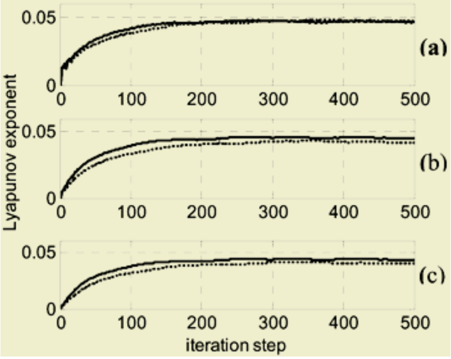


Fig. 3. Estimation of largest Lyapunov exponent for original (real line) and generated (dotted line) time series of Santa Fe competition.(a)m=2, (b)m=4, (c)m=6.

The correlation dimension and the largest Lyapunov exponents were computed using in house implementations of the Grassberger's algorithm [10] and the Wolf's algorithm [9] respectively. In Figure 2 the correlation integral map (CIM) and its slope are depicted both for the original and reconstructed time series. The value of the correlation dimension is defined as the slope of the CIM curves for at least 3 consecutive embeddings. In this case the correlation dimension for the predicted time series is 2.026 for the region 3<r<5. The correlation dimension for the time original time series is 2.124 for 3<r<5. Figure 3 shows that largest Lyapunov exponent of reconstruction time series is able to follow ones of original time series with a very small error.

6 Conclusion

Stationary wavelet combined with RLS-SVM have the ability to capture the dynamics of nonlinear dynamical systems as was demonstrated for the system of Santa Fe competition. This opinion is based on the fact that the invariants of the original and generated time series are very similar.

References

1. Brillinger, D.R.: Time series, Data Analysis and Theory. McGraw-hill, New York (1981)
2. Farmer J.D., Sidorowich, J.J.: Predicting Chaotic Time Series, Phys. Rev. Lett. **59** (1987) 845-848
3. GanPcay R.: A Statistical Framework for Testing Chaotic Dynamics Via Lyapunov Exponents. Physica D, **89** (1996) 261-266
4. Vapnik, V.. The Nature of Statistical Learning Theory. N.Y.,Springer (1995)
5. Suykens, J.A.K., Van Gestel, T., De Brabanter, J., De Moor, B., Vandewalle, J.: Least Squares Support Vector Machines. World Scientific (2002)
6. Suykens, J. A. K., Vandewalle, J.: Recurrent Least Squares Support Vector Machines. IEEE Trans. on Circuits and System-I: Fundamental Theory and Applications, **47** (2000) 1109-1114
7. Kaplan, D., Glass, L.: Understanding Nonlinear Dynamics. Springer, New York (1995)
8. Takens, F.: Detecting Strange Attractors in Fluid Turbulence. In D. Rand and L.S. Young, editors, Dynamical systems and turbulence, Springer-Verlag, Berlin (1981) 366-381
9. Wolf, A., Swift, J. B., Swinney, H. L., Vastano, J. A.: Determining Lyapunov Exponents from A Time Series. Physica D, **16** (1985) 285-317
10. Grassberger, P, Procaccia, I.: Characterization of Strange Attractors. Physical Review Letters, Vol. **50** (1983) 346–349
11. Nason, G.P., Silverman, B.W.: The Stationary Wavelet Transform and Some Statistical Applications, Lecture Notes in Statistics, Vol. 103, Springer-Verlag, Berlin Heidelberg New York (1995) 281-299
12. Weigend, A. S., Gershenfeld, N. A.: Time Series Prediction: Forecasting the Future and Understanding the Past. Reading, MA: Addison-Wesley (1994)
13. Cao, L.: Practical Method for Determining the Minimum Embedding Dimension of a Scalar Time Series. Physica D, **110** (1997) 43-50
14. Fraser, A. M., Swinney, H.L.: Independent Coordinates for Strange Attractors from Mutual Information. Phys. Rev. A, **33** (1986)1134-1140

Adding Value to System Dynamics Modeling by Using Artificial Neural Network

Changrui Ren, Yueting Chai, and Yi Liu

Department of Automation, Tsinghua University, Beijing 100084, China
rcr00@mails.tsinghua.edu.cn
{chaiyt,yiliu}@mail.tsinghua.edu.cn

Abstract. The study of system dynamics starts from model construction and simulation to understand and solve dynamical complicated problems. Traditional approaches of modeling process depend on experts' experiences and the trial-and-error procedure, so it is difficult to guarantee a useful model. Because a system dynamics model is equivalent to a specially-designed artificial neural network, both of which operate under the same numerical propagation constraints, we use the artificial neural network training algorithms and take advantage of historical data to assist system dynamics model construction. Experimental studies show that this approach is feasible.

1 Introduction

System dynamics (SD) is an approach for exploring the nonlinear dynamic behavior of a system and studying how the structure and the parameters of the system lead to behavior patterns. The essential viewpoint of SD is that feedback and delay cause the behavior of systems, i.e., that dynamic behavior is a consequence of system structure. SD was conceived and developed in the late 1950s at MIT by Jay W. Forrester [1]. Ever since then, SD has been applied to a wide range of problem domains [2]. It includes work in corporate planning and policy design, economic behavior, public management, biological and medical modeling, energy and the environment, dynamic decision making, software engineering, and so on.

Yet, despite the widespread recognition of the usefulness of SD method, there are still some issues which require further study if it is to be effective in supporting various decision making processes. SD is a model-based method, so building SD model is the foundation of further analysis and policy design. The models involved in SD include both qualitative model by causal loop diagrams (CLDs) and quantitative model by stock-and-flow diagrams. Traditionally, the model construction mainly relies on domain experts' insight and experience to solve the problem in a trial-and-error style, which is a subjective method and can not guarantee the models to be effective and efficient. It is a certain kind of art, and a time and effort consuming process. Taking a holistic view of the field of SD modeling, we can conclude that much of the work has focused on how to build CLDs or stock-and-flow diagrams with less concern for testing if these models are reasonable. That is the reason why many efforts in SD-based analyses have not met with great success.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 430–435, 2005.

In order to overcome the shortcomings of traditional modeling methods, we present a data-based approach to assist SD model construction. Based on the transformation between SD model (i.e., a stock-and-flow diagram) and artificial neural network (ANN), we derive models or parameters from historical data by using ANN's training algorithms. These models can either be used to further analyses directly or at least enhance an expert's analytical approach.

The remainder of this paper is structured as follows. At first, we give a literature review of SD modeling in Section 2. In Section 3, we will discuss the transformation between a SD model and a specially-designed ANN, and then in Section 4, we will illustrate how our approach can add value to SD model construction. Finally, in Section 5 we conclude with some closing remarks.

2 Literature Review

There is no cookbook recipe for successful SD modeling, no procedure you can follow to guarantee a useful model. Modeling is inherently creative. Individual modelers have different styles and approaches. For example, Starr's approach included two phases: problem definition, structuring and analysis [3]; Randers divided the SD modeling process into four steps: conceptualization, formulation, testing, and implementation [4]; Furthermore, Forrester presented a more detailed six-phase modeling process: describe the system, convert description to level and rate equations, simulate the model, design alternative policies and structures, educate and debate, implement changes in policies and structure [5]. Yet all successful modelers follow a disciplined process that involves the following activities: [6]

(1) Problem articulation (boundary selection)

Choose the theme, key variables, time horizon, and reference modes.

(2) Formulation of dynamic hypothesis

Develop maps of causal structure based on dynamic hypotheses, key variables, reference modes, and other available data.

(3) Formulation of a simulation model

Specify the structure and decision rules, and estimate parameters, behavioral relationships, and initial conditions.

(4) Testing

Test the robustness under extreme conditions by comparison to reference modes.

(5) Policy design and evaluation

Design policies, do what-if analysis and sensitivity analysis.

Although the above would indicate that the modeling process is linear, it is, in fact, iterative. Results of any step can yield insights that lead to revisions in any earlier step.

3 The Transformation Between SD Model and ANN

Intuitively, a SD model is much similar to an ANN, because they all store knowledge expression in the network structure and the weights of linkages. Kofjac [7] first demonstrated the possibility of interconnecting SD methodology and ANN modeling. He showed that ANNs can be easily created by SD modeling, so that can help the understanding of the "physics" of ANNs. Chen [8] further proposed a transformation method that could map a SD model to a specially designed ANN - partial recurrent network (PRN). He also proved in mathematics that they two operated under the same numerical propagation constraints.

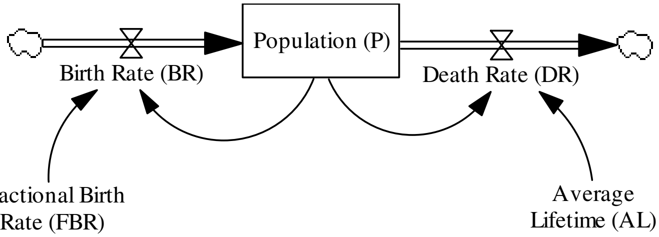


Fig. 1. A simple SD model

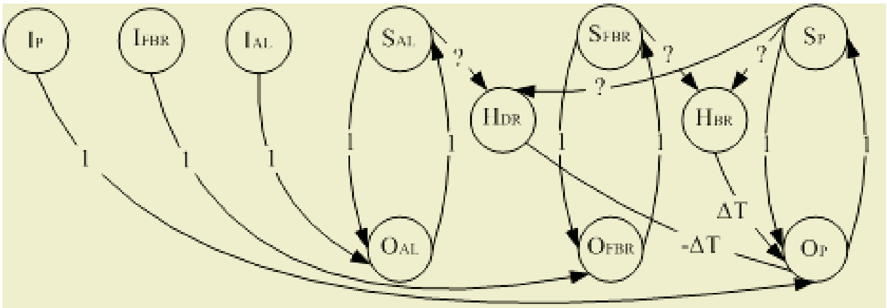


Fig. 2. Transforming the SD model into an ANN

Shown in Fig. 1 is a simple SD model (stock-and-flow diagram) for population management. In this model, there are two decision points, birth rate (*BR*) and death rate (*DR*), which control the flow into and out of a level (population, *P*). Fractional birth rate (*FBR*) and average lifetime (*AL*) are constants and propagated to *BR* and *DR* through wires. Using the algorithm FD2PRN presented by Chen [8], this model can be transformed into a PRN. The above model in ANN form will be look like Fig. 2. It is composed of four parts: the top left three nodes are input units, which feed data into the network at initialization; the bottom three nodes are output units; the top right three nodes are state units, which keep the previous values of the output units; and the middle two units are hidden units. The relationship with the original model is following. As illustrated in Fig. 2, level (*P*) and constants (*BR* and *DR*) are each mapped to three units: input I_P, output O_P, state S_P; input I_{FBR}, output O_{FBR}, state S_{FBR}; and input I_{AL}, output O_{AL}, state S_{AL}, respectively. Rate (*BR* and *DR*) and the flow are mapped to hidden units (H_{BR} and H_{DR}) and the links (from H_{BR} to O_P, from H_{DR} to O_P) respectively. The others that appear as a coefficient in a rate equation are mapped to the links from the state units to the hidden units (e.g., from S_{AL} and S_P to H_{DR}). The details of the individual relationships between the corresponding components in two models are listed in Table 1 [8].

4 ANN Based Method for SD Model Construction

As we mentioned above, SD models can be transformed into ANNs, thus we could use the ANN training algorithms to assist model construction and policy design. The modeling process can be illustrated by Fig. 3.

Table 1. The mappings between SD Model and ANN

Components for SD Model	Components for ANN
Level variable, constant (not for coefficient)	A triple of input, output, and state units
Rate (or auxiliary) variable	Hidden unit
Wire	Link from a state unit to a hidden unit
Flow	Link from a hidden unit to output unit
Level equation	A weighted sum of the values of hidden and state units connecting to an output unit via links
Rate equation (including constants as coefficients)	Any function of the values of state units connecting to a hidden unit via links
Equation for initial value	Link from an input unit to an output unit
Constant equation	Link from a state unit to an output unit

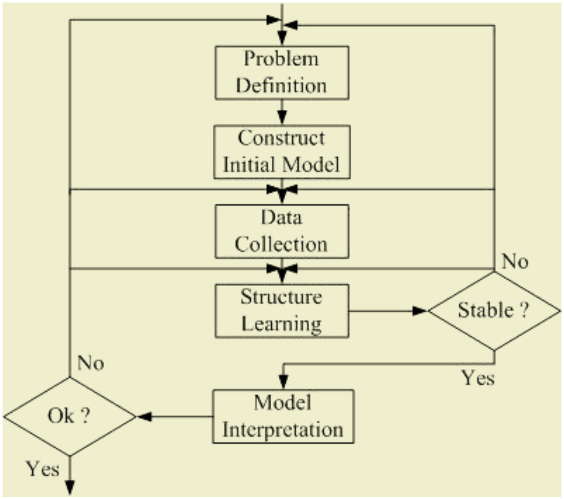


Fig. 3. An ANN-based SD modeling process

(1) Problem definition

In this approach, the causal links are derived by ANN learning, so we need only select level variables, rate variables, and the flows between levels in the SD model.

(2) Construct initial model

Based on the problem definition, we can firstly construct the model by stock-and-flow diagrams, and then map to ANNs. However, the model still lacks causal links, which can be derived by ANN learning. So we set the state level and hidden level in ANN as full connection. In the meantime, in order to reduce the complexity, we do not need learning all the causal links, because some links can be easily determined by common knowledge or decision makers' judgment. So we need only confirm weights (constant), causal links, and rate equations through ANN learning, while leaving other parameters to decision makers.

(3) Data collection

We need two kinds of data for ANN training: input and output, and the data here may be a little different from normal ANN training. So the data collection process is very difficult, and sometimes it may need special design, e.g., the use of some numerical analysis techniques.

(4) Structure learning

The structure learning can adopt the standard BP (BackPropagation) algorithm with the least parameters: η (learning rate), d_{max} (maximum permit error), and t (training ratio) [9]. If it can not reach a stable structure, we have to trace up to look for the reasons and adopt corresponding measures.

(5) Model interpretation

The structure derived by ANN learning is only reasonable in mathematics, while may not have concrete meanings. So we have to interpret the model to test the results, which is not necessary in traditional modeling methods. If the results do not make sense, we have to re-train the ANN or change the network structure and original data to obtain a satisfactory result eventually, and just as we mentioned above, the modeling process is iterative.

In the modeling process, we can take advantage of many existing methods in ANN [9]. We use the above modeling process to model the simple example in Section 3, and part of the learning effectiveness (for *BR*) is as follows (see Table 2).

Table 2. The learning process

	1	50	100	200	In Model	SD
FBR→BR	0.07521	0.12878	0.09976	0.1	0.1	
P→BR	0.08258	0.18779	0.19985	0.2	0.2	
SSE[1]	1.1021	2.3125×10^{-3}	3.3824×10^{-5}	2.8224×10^{-8}		

Given η=0.1, the result indicates that this method is at least feasible. But there are still large gaps to overcome if it were to take effect in practice, these are:

- Ability to deal with large scale models.
- Data collection problem, e.g., the countermeasures to imperfect data source.
- Improvement of specified ANN training method.

Besides model construction, the ANN approach can also assist SD policy design by fitting a desired behavior pattern. Since any ANN can learn some structures from a historical time series pattern, it can also learn a better structure from a better pattern set by designer. The current policy design methods mainly focus on adjusting single parameter or parameter combinations, while the ANN approach can optimize the policy from the holistic viewpoint.

[1] SSE: Error Sum of Squares, SSE= $\sum(y_i - \hat{y}_i)^2$.

5 Conclusions

This paper has presented a SD model construction method by mapping a SD Model to an ANN. This approach integrates the capability of ANN learning with a traditional process, which thus makes model construction more systematic and much easier for common people. In further research, we can introduce genetic algorithms (GAs) in the model building process, by encoding an ANN into a string and uses an evolution process to select a best solution. The algorithm can not only improves the ANN training, but also generates more candidate models for consideration. Moreover, the real world data always contains many noises, so further research will also be conducted in this direction.

Acknowledgements

This work was supported by the National High-Tech. Research & Development Program for CIMS, P.R. China (Grant No. 2003AA413032).

References

1. Forrester, J.W.: Industrial Dynamics. The MIT Press, Cambridge, Mass (1961)
2. Angerhofer, B.J., Angelides, M.C.: System Dynamics Modeling in Supply Chain Management: Research Review. In: Joines, J.A., Barton, R.R., Kang, K., Fishwick P.A. (eds.): Proceedings of the 2000 Winter Simulation Conference (2000) 342-351
3. Starr, P.J.: Modeling Issues and Decisions in System Dynamics. TIMS Studies in the Management Science, **14** (1980) 45-59
4. Randers, J.: Guidelines for Model Conceptualization. In: Randers, J. (eds.): Elements of the System Dynamics Method. MIT Press, Cambridge, Mass (1980)
5. Forrester, J.W.: System Dynamics, Systems Thinking, and Soft OR. System Dynamics Review, **10** (1994) 245-256
6. Sterman, J.D.: Business Dynamics: Systems Thinking and Modeling for a Complex World. Irwin/McGraw-Hill, Boston (2000)
7. Kofjac, D., Skraba, A., Kljajic, M.: Neural Network Modeling by System Dynamics Methodology-basic Concepts. The IEEE Region 8 EUROCON, **1** (2003) 424-428
8. Chen, Y.-T.: The Use of SDM-PRN Transformation for System Dynamics Model Construction and Policies Design. Doctoral Dissertation, National Sun Yat-sen University (2001) (in Traditional Chinese)
9. Zell, A., Mamier, G., Vogt, M., et al.: SNNS User Manual, Ver. 4.1. University of Stuttgart (1995)

Least Squares Wavelet Support Vector Machines for Nonlinear System Identification

Zhenhua Yu and Yuanli Cai

School of Electronic and Information Engineering
Xi'an Jiaotong University, Xi'an 710049, China
ylicai@mail.xjtu.edu.cn

Abstract. A novel admissible support vector kernel, namely the wavelet kernel satisfying wavelet frames, is presented based on the wavelet theory. The wavelet kernel can approximate arbitrary functions, and is especially suitable for local signal analysis, hence the generalization ability of the support vector machines (SVM) is improved. Based on the wavelet kernel and the least squares support vector machines, the least squares wavelet support vector machines (LS-WSVM) are constructed. In order to validate the performance of the wavelet kernel, LS-WSVM is applied to a nonlinear system identification problem, and the computational process is compared with that of the Gaussian kernel. The results show that the wavelet kernel is more efficient than the Gaussian kernel.

1 Introduction

Nonlinear system identification is a crucial but complex problem, and many efforts and attentions have been paid to its study. In recent years, neural networks [1], wavelet [2], and wavelet networks [3] have been widely applied to investigate nonlinear system identification. While classical neural networks suffer from the lack of efficient construction methods in determining the parameters of neurons and choosing the number of hidden units, and easily settle in undesirable local minima of the error surface. Wavelet theory emerges as a powerful tool for nonlinear system identification, which has exceptional property of time-frequency localization. However, the wavelet is usually limited to small dimension. The reason is that constructing and storing wavelet basis of large dimension are of prohibitive cost [4]. Wavelet networks can avoid some weaknesses of wavelet and neural networks, nevertheless the parameters selection of the network structures and the study of the training algorithms are difficult tasks.

Support vector machines (SVM) is a new universal learning machine proposed by Vapnik et al. in the framework of Structural Risk Minimization (SRM). SRM has greater generalization ability and is superior to the traditional Empirical Risk Minimization (ERM) principle adopted in many conventional neural networks. In SVM, the results guarantee global minima whereas ERM can only locate local minima. SVM uses a kernel function to map the input data into a high-dimensional feature space, and then constructs an optimal separating hyperplane in that space. SVM is a kernel based approach, which allows the use of

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 436–441, 2005.

linear, Gaussian, polynomial and RBF kernels and so on that satisfy Mercer's condition. Least squares support vector machines (LS-SVM) is an SVM version which involves equality instead of inequality constraints and works with a least squares cost function. Recently, LS-SVM has been applied to classification [5] and function estimation [6]. As the wavelet has the property of time-frequency localization and is a powerful tool for arbitrary function approximation, a wavelet kernel satisfying wavelet frames is proposed, which is an extension of the work presented in [11]. Combining LS-SVM with the wavelet kernel, the least squares wavelet support vector machines (LS-WSVM) are proposed. Finally, the LS-WSVM is applied to the nonlinear system identification to test the efficiency and validity of the wavelet kernel.

This paper is organized as follows. Section 2 briefly reviews some basic notions of LS-SVM for function estimation. Section 3 presents a wavelet kernel. Section 4 applies LS-WSVM to nonlinear system identification and compares to the SVM based on Gaussian kernel. Section 5 summarizes the results of this paper.

2 Least Squares Support Vector Machines for Nonlinear Function Estimation [7],[8]

Given a training data set of N points $\{x_k, y_k\}_{k=1}^N$ with the input data $x_k \in R^n$ and the corresponding target $y_k \in R$. In feature space SVM models take the form

$$y(x) = \omega^T \varphi(x) + b \tag{1}$$

where the nonlinear mapping $\varphi(\cdot)$ maps the input vector into a higher dimensional feature space, b is the bias and ω is a weight vector of the same dimension as the feature space. In LS-SVM for function estimate, one considers the following optimization problem

$$\min_{\omega,e} J(\omega, e) = \frac{1}{2}\omega^T \omega + \frac{1}{2}\gamma \sum_{k=1}^{N} e_k^2 \tag{2}$$

subject to the equality constraints

$$y_k = \omega^T \varphi(x) + b + e_k \ \ , k = 1, \ldots, N \tag{3}$$

here γ is the regularization parameter.

This problem can be solved by using the optimization theory. One can define the Lagrangian for this problem as follows

$$L(\omega, b, e; \alpha) = J(\omega, e) - \sum_{k=1}^{N} \alpha_k (\omega^T \varphi(x_k) + b + e_k - y_k) \ . \tag{4}$$

In this equation, the α_k's are called the Lagrangian multipliers. The saddle point can be found by setting the derivatives equal to zero

$$\begin{cases} \frac{\partial L}{\partial \omega} = 0 \rightarrow \omega = \sum\limits_{k=1}^{N} \alpha_k \varphi(x_k) \\ \frac{\partial L}{\partial b} = 0 \rightarrow \sum\limits_{k=1}^{N} \alpha_k = 0 \\ \frac{\partial L}{\partial e_k} = 0 \rightarrow \alpha_k = \gamma e_k \\ \frac{\partial L}{\partial \alpha_k} = 0 \rightarrow \omega^T \varphi(x_k) + b + e_k - y_k = 0 \end{cases} \tag{5}$$

for $k = 1, \ldots, N$. Elimination of e_k and ω through substitution in the following set of linear equations

$$\left[\begin{array}{c|c} 0 & \overrightarrow{1}^T \\ \hline \overrightarrow{1} & \Omega + \gamma^{-1} I \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline y \end{array}\right] \tag{6}$$

where $y = [y_1; \ldots; y_N]$, $\overrightarrow{1} = [1; \ldots; 1]$, $\alpha = [\alpha_1; \ldots; \alpha_N]$ and $\Omega_{kl} = \varphi(x_k)^T \varphi(x_l)$ for $k, l = 1, \ldots, N$. According to Mercer's condition, there exists a mapping φ and an expansion

$$K(x, y) = \sum_i \varphi_i(x) \varphi_i(y), \quad x, y \in R^n \quad . \tag{7}$$

The resulting LS-SVM model for function estimation becomes

$$y(x) = \sum_{k=1}^{N} \alpha_k K(x, x_k) + b \tag{8}$$

where α, b are the solutions of (6). In this paper, for the choice of the kernel function $K(\cdot, \cdot)$, one chooses Gaussian kernel, $K(x, x^{'}) = exp(-\|x - x^{'}\|^2 / 2\sigma^2)$.

3 Least Squares Wavelet Support Vector Machines

3.1 Conditions for Support Vector Kernel [9],[10]

The kernel function is a dot product in the feature space, which must satisfy the conditions in Mercer theorem.

Theorem 1. *In $L_2(R^N)$ space (R^N denotes the input space), $\forall g(x) \in L_2(R^N)$, $\int g^2(x) dx < \infty, g \neq 0$ and $K(x, x^{'}) \in L_2(R^N \times R^N)$, if*

$$\int \int_{L_2 \bigotimes L_2} K(x, x^{'}) g(x) g(x^{'}) dx dx^{'} \geq 0 \tag{9}$$

holds, $K(x, x^{'})$ is a dot product in the feature space.

Theorem 1 is a simple method for judging and constructing support vector kernel.

Theorem 2. *A translation invariant kernel $K(x, x^{'}) = K(x - x^{'})$ is an admissible support vector kernels if and only if the Fourier transform*

$$F[k](\omega) = (2\pi)^{-N/2} \int_{R^N} exp(-j(\omega \cdot x)) K(x) dx \geq 0 \tag{10}$$

holds.

3.2 Wavelet Kernels [11]

Theorem 3. *Let $\psi(x) \in L^2(R)$ be a mother wavelet, and $a \in R$ and $b \in R$ be the dilation and translation parameters respectively. if $x, x^{'} \in R^N$, then the dot product wavelet kernels are*

$$K(x, x^{'}) = \prod_{i=1}^{N} \psi(\frac{x_i - b_i}{a_i})\psi(\frac{x_i^{'} - b_i^{'}}{a^i}) \tag{11}$$

and translation invariant wavelet kernels that satisfy Theorem 2 are

$$K(x, x^{'}) = \prod_{i=1}^{N} \psi(\frac{x_i - x_i^{'}}{a_i}) \ . \tag{12}$$

Choosing an appropriate wavelet function as a wavelet kernel is a critical problem. One takes into account not only the wavelet function satisfying the Mercer's condition, but also the properties of the wavelet function. For orthonormal wavelet bases, the wavelet function must satisfy strong restrictions, and the construction of orthonormal bases is complex. Furthermore, as the training data are sparse, one cannot make full use of the advantage of orthonormal wavelet bases [4]. Another solution is to adopt wavelet frames. By relaxing the orthogonality, much more freedom on the choice of the wavelet function is gained. It is possible to generate a single-scaling frame of $L^2(R^d)$ with a single mother wavelet function. In contrast, in order to construct a single-scaling orthonormal wavelet basis of $L^2(R^d)$, $2^d - 1$ mother wavelet functions and one scaling function are needed. Therefore, the orthonormal bases are more complex than the wavelet frames. In this paper, a translation-invariant wavelet kernel is constructed by choosing a wavelet function satisfying the conditions of wavelet frames. The wavelet function is chosen as follows

$$\psi(x) = \cos(1.75x)exp(-\frac{x^2}{2}) \ . \tag{13}$$

Theorem 4. *Using the wavelet frame (13), a wavelet kernel can be constructed as*

$$K(x, x^{'}) = \prod_{i=1}^{N} \psi(\frac{x_i - x_i^{'}}{a_i}) = \prod_{i}(\cos(1.75 \times \frac{(x_i - x_i^{'})}{a_i})\exp(-\frac{\|x_i - x_i^{'}\|^2}{2a_i^2})) \tag{14}$$

which is an admissible support vector kernel.

Substituting (14) into (8), one can obtain function estimation of LS-WSVM

$$y(x) = \sum_{k=1}^{N} \alpha_k \prod_{i}^{N}(\cos(1.75 \times \frac{(x_k^i - x_k^{i'})}{a_k^i})\exp(-\frac{\|x_k^i - x_k^{i'}\|^2}{2a_k^{i2}})) + b \tag{15}$$

where x_k^i represents the ith component of the kth training example.

Now, LS-SVM can adopt the wavelet kernel as its kernel function. As LS-SVM cannot optimize the parameters of kernels, it is difficult to determine $N \times N$ parameters a_k^i, $i = 1, \ldots, N$. For the sake of simplicity, one lets $a_k^i = a$, so the number of parameters becomes 1 [11].

4 Application of LS-WSVM to Nonlinear System Identification

In this section, to validate the performance of the wavelet kernel, LS-WSVM is applied to nonlinear system identification. At the same time, the results obtained by the wavelet kernel are compared with that obtained by the Gaussian kernel.

Table 1. Results of approximation

Kernel	Parameter	Approximation error
Wavelet kernel	α =3	0.0178
Gaussian kernel	σ =3	0.0439

Example 1. Consider a SISO nonlinear system[1]. The input $u(t)$ and the output $y(t)$ are related by the difference equation as follows:

$$y(t+1) = \frac{y(t)}{1+y^2(t)} + u^3(t) \tag{16}$$

where $u(t)$is random input in interval [-2,2], $u(t) = \sin(2\pi t/25) + \sin(2\pi t/10)$. One takes 100 points as the training samples, and 80 points as testing samples. The approximation errors using the wavelet kernel and the Gaussian kernel are listed in Table 1.

Table 2. Results of approximation

Kernel	Parameter	Approximation error
Wavelet kernel	α =3	0.0312
Gaussian kernel	σ =4	0.0815

Example 2. A nonlinear system to be identified is governed by the difference equation $y(k+1) = 0.3y(k) + 0.6y(k-1) + f(u(k))$, where $f(u) = 0.6\sin(\pi u) + 0.3\sin(3\pi u) + 0.1\sin(5\pi u)$, and the input $u(k) = \sin(2\pi k/250)$. One takes 700 points as the training samples, 400 of which are taken as testing samples. Table 2 lists the approximation errors using the two kernels.

From Table 1, and Table 2, one can find the proposed wavelet kernel has better result than the Gaussian kernel. The wavelet kernel is not only a kind of multidimensional wavelet functions, but also especially suitable for local signal analysis, the signal-noise separation and the detection of jumping signals.

5 Conclusions

In this paper, a wavelet kernel is proposed. The wavelet kernel possesses the advantage of approximating arbitrary functions. A new SVM version is constructed based on LS-SVM and the wavelet kernel, namely LS-WSVM. LS-WSVM is applied to nonlinear system identification. Simulation shows that the wavelet kernel has better approximation than the Gaussian kernel, and the wavelet kernel is more suitable to process the jumping signals.

References

1. Narendra, K., Parthasarathy, K.: Identification and Control of Dynamical Systems Using Neural Networks. IEEE Trans. on Neural Networks, **1** (1990) 4–27
2. Benveniste, A., Juditsky, A., Delyon, B., Zhang, Q., Glorennec, P.Y.: Wavelets in Identification. 10th IFAC Symposim System Identification, Copenhagen, Denmark (1994)
3. Zhang, Q., Benveniste, A.: Wavelet Networks. IEEE Trans. on Neural Networks, **3** (1992) 889–898
4. Zhang, Q.: Using Wavelet Network in Nonparametric Estimation. IEEE Trans. On Neural Networks, **8** (1997) 227–236
5. Suykens, J.A.K., Vandewalle, J.: Least Squares Support Vector Machine Classifiers. Neural Processing Letters, **9** (1999) 293–300
6. Saunders, C., Gammerman, A., Vovk, V.: Ridge Regression Learning Algorithm in Dual Variables. Proceedings of the 15th International Conference on Machine Learning, Morgan Kaufmann (1998) 515–521
7. Suykens, J.A.K., Brabanter, J.De., Lukas, L., Vandewalle, J.: Weighted Least Squares Support Vector Machines Robustness and Sparse Approximation. Neurocomputing, **48** (2002) 85–105
8. Kruif, Bas.J.de., Vries, Theo.J.A.de.: Pruning Error Minimization in Least Squares Support Vector Machines. IEEE Trans. On Neural Networks, **14** (2003) 696–702
9. Mercer, J.: Functions of Positive and Negative Type and Their Connection with the Theory of Integral Equation. Transactions of the London Philosophical Society, **A-209** (1909) 415–446
10. Smola, A., Sch*ö* lkopf, B., M*ü*ller, K.R.: The Connection Between Regularization Operators and Support Vector Kernels. Neural Networks, **11** (1998) 637–649
11. Zhang, L., Zhou, W.D., Jiao, L.C.: Wavelet Support Vector Machine. IEEE Trans. On Systems, Man, and Cybernetics-Part B: Cybernetics, **34** (2003) 1–6

Wavelet Support Vector Machines and Its Application for Nonlinear System Identification*

Xiangjun Wen,Yunze Cai, and Xiaoming Xu

Automation Department, Shanghai Jiaotong University, Shanghai 200030, China
{Wenxiangjun,yzcai,xmxu}@sjtu.edu.cn

Abstract. Based on a novel multidimensional wavelet kernel constructed in Reproducing Kernel Hilbert Space (RKHS), an identification scheme with the Wavelet Support Vector Machine (WSVM) estimator is proposed for nonlinear dynamic systems. The good reproducing property of wavelet kernel function enhances the generalization ability of the system identification scheme. Two cases are presented to validate the proposed method and show its feasibility.

1 Introduction

Wavelet analysis has been frequently used in a wide range of fields, from signal processing, numerical analysis, image decomposition to automatic control [2], [3], [4], [5]. The wavelet-based neural network (wavelet networks [6], [7]) has been proposed as an alternative to the feed forward neural network. It can greatly remedy the weakness of both wavelets and neural networks and has been widely used in the fields of classification and approximation with great success. However, wavelet networks find the solution by minimizing an empirical risk (usually a mean square error) with gradient-based training method such as Back Propagation, and they often suffer from the existence of multiple local minima.

As a new methodology of learning theory, Support Vector Machines (SVM) originally introduced by Vapnik [9] within the area of the statistical theory and structural risk minimization has proven to be a more powerful approach in many areas. Kernel techniques like wavelet networks also rely on similar basis functions and their behaviors should be related. It is a valuable issue whether a better performance could be obtained if we combine the wavelet decompositions with kernel method. Actually it has caused great interest of many researchers in the last few years [8], [15]. Noting that a "good" (generalization) model of nonlinear system identification is closely related to a rather modest approximation problem, we construct a SVM estimator based on reproducing wavelet kernels in Reproducing Kernel Hilbert Space (RKHS) and develop a framework for nonlinear system identification.

* This work was supported by the national 973 key fundamental research project of China under grant 2002CB312200 and national 863 high technology projects foundation of China under grant 2002AA412010.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 442–447, 2005.

This paper is organized as follows. In the next section we give a general introduction to SVM for regression. Then in section 3 we propose a practical approach to construct wavelet kernels in RKHS. In section 4, two application cases for nonlinear black-box modeling are presented to show the applicability and the feasibility of the proposed method. Finally, we conclude in section 5.

2 SVM for Nonlinear Function Estimation

Given a training data set D of l samples independent and identically drawn (i.i.d.) from an unknown probability distribution $\mu(\mathrm{X},\mathrm{Y})$ on the product space $Z = X \times Y$:

$$D = \{z_1 = (x_1, y_1), \dots z_n = (x_l, y_l)\} \tag{1}$$

where the input data X is assumed to be a compact domain in a Euclidean space R^{d} and the output data Y is assumed to be a closed subset of R. Learning from the training data can be viewed as a multivariate function f approximation that represents the relation between the input X and output Y [1], [9]. By some nonlinear mapping Φ, input X is mapped onto a hypothesis space (feature space) in which the learning machine (algorithm) selects a certain function f.

We would like to find a function that gives a small-expected error $R[f]$. To avoid ill-posed problems, we have to look for a function f that minimizes the regularized empirical risk instead of the empirical risk according to regularization theory [13], [14].

$$R_{reg}[f] = \frac{1}{l}\sum_{i=1}^{l} C(x_i, y_i, f(x_i) + \lambda(\| f \|_H^2) \tag{2}$$

where H is a RKHS, λ a regularization parameter, and $C(\cdot)$ is a cost function such as the square loss function and classic Vapnik's ε -insensitive loss function. In this paper, we choose Vapnik's ε -insensitive loss function to minimize the empirical risk

$$R_{emp}[f] = \frac{1}{l}\sum_{i=1}^{l} \max(0, | y_i - f(x_i) | - \varepsilon) \tag{3}$$

where $\varepsilon \geq 0$. By minimizing the regularized empirical risk shown in equation (2), we can obtain Wavelet Support Vector Machines (WSVM) estimator as follows:

$$\hat{f}(x) = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) K(x, x_i) + b \tag{4}$$

where $b \in R$, α_i, α_i^* are the Lagrange multipliers, and $K(\cdot,\cdot)$ is the reproducing wavelet kernel which will be discussed in detail in section 3.

3 Reproducing Wavelet Kernel Method

RKHS is a Hilbert space with special properties. The interest of RKHS arises from its associated kernel functions. For simplicity, we give the following proposition without proof.

Proposition 1: Any finite set of wavelet frames of $\mathrm{L}^2(R)$ endowed with inner product spans a RKHS, and its Reproducing Kernel is $K(x, y) =< \Gamma_x(.), \Gamma_y(.) >_{L^2}$. Where $\Gamma_t(.) \in L^2(R)$ is a family of functions indexed by $\mathrm{t} \in X$ (*X* being any subset of *R*), which acts as the evaluation functional in t.

The proof of proposition 1 is nontrivial. Due to page limit, we present it here without proof. For a thorough discuss of building RKHS with kernel operator and wavelet frame the reader is further referred to [3], [10], [11], [12], [15], [16], [18].

Let $\phi(\mathrm{x})$ be a mother wavelet, and let a and b denote the dilation and translation factor, respectively, $a, b \in R$, then according to wavelet theory

$$\phi(x) =| a |^{-1/2} \phi(\frac{x-b}{a}) \tag{5}$$

Rewritten with multi index form, that is

$$\phi_i(x) = \phi_{j,k}(x) = a_0^{-j/2} \phi(a_0^{-j} x - kb_0) \tag{6}$$

where $\mathrm{a}_0, \mathrm{b}_0 \in R$, $\mathrm{j,k} \in \mathrm{Z}$, i denotes a multi index . It is known that when the function $\phi(\cdot)$ satisfies the necessary condition (admissibility of the mother wavelet in theorem 3.3.1 of Daubechies [3]) and suitable parameters with a_0, b_0 will lead to wavelet frames.

From proposition 1, the function as follows is chosen for reproducing kernel construction

$$\Gamma_x(.) = \sum_{i,j} \alpha_{i,j} \phi_j(x) e_i(.) \tag{7}$$

where $\alpha_{i,j} = c_j \delta_{i,j}$ is the coefficients combining the orthonormal basis $\{e_i\}$ of $L^2(R)$ with wavelet basis ϕ_j of $L^2(R)$, and c_j is a coefficient depending on the considered wavelet ϕ_j.

We can construct a wavelet kernel in RKHS as follows:

$$K(x, y) = \sum_{i,j,n} \alpha_{i,j} \alpha_{j,n} \phi_j(x) \phi_n(y) \tag{8}$$

For a common multidimensional wavelet kernel function, we can write it as the product of one-dimensional (1-D) wavelet function according to tensor products theory proposed by Aronszajn [16].

$$K_d(x,y)=\prod_{i=1}^{d}K(x_i,y_i) \tag{9}$$

In this paper, we only use a subset of Daubechies wavelet, which is compactly supported and orthonormal basis of $L^2(R)$ [18]. Moreover, we set coefficients in (10) so that the kernel in RKHS can be written as follows:

$$K(x,y)=\sum_{j=j_{\min}}^{j_{\max}}\sum_{k=k_{\min}}^{k_{\max}}\phi_{j,k}(x)\phi_{j,k}(y) \tag{10}$$

where j,k are the dialation and translation parameters of a mother wavelet function $\phi(x)$ respectively, $j_{\min}$ and $j_{\max}$ are the minimum and maximum dialations, and $k_{\min}$ and $k_{\max}$ are the minimum and maximum translations of the wavelet kernel, respectively.

4 Application Study of Nonlinear Black-Box System Identification

In this section, two simulations are presented to validate the proposed method and show its feasibility for nonlinear system identifications.

Example 1: The plant is assumed to be of the form [17]

$$\hat{y}(k+1)=f[y(k),y(k-1),y(k-2),u(k),u(k-1)] \tag{11}$$

where the unknown function *f* has the form

$$f[x_1,x_2,x_3,x_4,x_5]=\frac{x_1x_2x_3x_5(x_3-1)+x_4}{1+x_3^2+x_2^2} \tag{12}$$

We generated 200 samples {u (k), y (k)} by using the random input signal uniformly distributed in the interval [-1 1] with the form:

$$\begin{gathered}\mathrm{u(k)}=\sin(2\pi k/250),\ k\le150,\ \text{Otherwise}\\ \mathrm{u(k)}=0.8\sin(2\pi k/250)+0.2\sin(2\pi k/25).\end{gathered} \tag{13}$$

The training data set consists of 100 samples and the other 100 were used as test data. We constructed NARX model with SVM based on reproducing wavelet kernel, and we compared the results with two popular criteria, the normalized root of mean-square-error (NRMSE) and maximal-absolute-error (MAXE). The simulation results are illustrated in table 1.

Table 1. Comparison results of several algorithms

S.I. Method	RMSE (train)	MAXE (train)	RMSE (test)	MAXE (test)
RBFNN	0.00105	0.11502	0.0270	0.4602
SVM	0.00061	0.10960	0.0039	0.1873
LS-SVM	0.00067	0.0972	0.0024	0.1295
WSVM	9.1634e-5	0.010	0.0012	0.1081

Example 2: The plant to be identified is described by the second-order difference equation [17]:

$$\hat{y}(k+1)=\frac{y(k)y(k-1)[y(k)+2.5]}{1+y(k)^2+y(k-1)^2}+u(k) \tag{14}$$

We generated 200 samples {u (k), y (k)} by using the random input signal uniformly distributed in the interval [-2 2] with the form:

$$\mathrm{u(k)}=\sin(2\pi k/25)+\cos(2\pi k/10) \tag{15}$$

The training data set consists of 100 samples and the other 100 samples were used as test data. The simulation results are illustrated in table 2.

Table 2. Comparison results of several kernel algorithms

S.I. Method	RMSE (train)	MAXE (train)	RMSE (test)	MAXE (test)
SVM	9.9581e-5	0.0110	9.9860e-5	0.0108
LS-SVM	4.5549e-4	0.0638	3.9932e-4	0.0430
WSVM	9.7149e-5	0.01	9.6166e-5	0.0100

In example 1, we showed the results obtained with NARX estimator for nonlinear system identification based on RBFNN, classic SVM, LS-SVM and WSVM. For comparison, the kernel parameters for Gaussian kernel of SVM and Lease Squares SVM (LS-SVM) are chosen as the same, and the regularization parameters for SVM and WSVM also hold. In example 2, we compared the NARX estimator for nonlinear system identification based on classic SVM, LS-SVM and WSVM with optimized parameters, respectively. All in all, WSVM based on reproducing wavelet kernel has better performance and generalization ability than the other methods.

5 Conclusions

In this paper, we discussed a practical way to construct wavelet kernel in RKHS. By combining a reproducing wavelet kernel with SVM, we proposed a nonlinear system identification framework based on WSVM estimator. Because reproducing wavelet kernel based on wavelet decomposition is not only orthonormal (or approximately orthonormal), but also suitable for local signal analysis and signal-noise separation, it is not surprising that the WSVM has good generalization ability. This technique provides a practical approach for nonlinear system modeling and the simulations show that the proposed method is feasible. In general, WSVM methodology might offer a new opportunity in the area of automatic control, the online training WSVM method and parameters optimization in equation (10) still remain to be further explored for the future.

References

1. Poggio, T., Rifkin R., Mukherjee, S., Niyogi, P.: General Conditions for Predictivity in Learning Theory. Nature, **428** (2004) 419-422.
2. Mallet, S.G.: A Theory for Multiresolution Signal Decompostion:the Wavelet Representation. IEEE Trans. on pattern Analysis and Machine intelligence, **11** (1989) 674-693
3. Daubechies, I.: Ten Lectures on Wavelets. CBMS-NSF Conference Series in Applied Mathematics, **137** SIAM Ed. (1992) 117-119
4. Mallet, S.G.: Multifrequentcy Channel Decomposition of Images and Wavelets Models. IEEE Trans. Accoust. Speech, and Signal Process. **37** (1989) 2091-2100
5. Sjöberg, J., Qinghua, Z. (ed.): Nonlinear Black-box Modeling in System Identification: Mathematical Foundations. Automatica, **31** (1995) 1725-1570
6. Qinghua, Z.: Using Wavelet Network in Nonparametric Estimation. IEEE Trans. on Neural Networks, **8** (1997) 227-236
7. Qinghua, Z., Benvenisite: Wavelet Networks. IEEE Trans. on Neural Networks, **3** (1992) 889-898
8. Li, Z., Weida, Z., Licheng, J.: Wavelet Support Vector Machine. IEEE Trans. on Systems, Man, and Cybernetics—Part B: CYBERNETICS, **34** (2004) 34-39
9. Vapnik, V.: The Nature of Statistical Learning Theory. 2nd edn. New York: Springer-Verlag (1998)
10. Scholkopf, B.: A Generalized Representer Theorem. Technical Report 2000-81, Neuro-Colt2 Technical Report Series (2000)
11. Suykens, J., Horvath,G., Basu, S., Micchelli, C., Vandewalle, J. (Eds.) : Advances in Learning Theory: Methods, Models and Applications.NATO Science Series III: Computer and Systems Sciences, **190** IOS Press, Amsterdam (2003) 89-110
12. Rakotomamonjy, A., Canu, S.: Frame, Reproducing Kernel, Regularization and Learning. Technical Report PSI (2002)
13. Evgeniou,.T. Pontil,M. ,Poggio, T.: Regularization Networks and Support Vector Machines. Advances in Computational Mathematics, **13** (2000) 1-50
14. Girosi,F.,Jones,M., Poggio,T.: Regularization Theory and Neural Network Architectures. Neural Computation, **7** (1995) 219-269
15. Rakotomamonjy, A. Mary X., Canu, S.: Wavelet Kernel and RKHS. Proc. of Statistical Learning: Theory and Applications, Paris (2002)
16. Aronszajn, N.: Theory of Reproducing Kernels. Transactions of the American Society. **68** (1950) 337–404.
17. Narendra K., Parthasarathy, K.: Identification and Control of Dynamical Systems Using Neural Networks. IEEE Trans. on Neural Network. **1** (1990) 4-27
18. Mallat, S.: A Wavelet Tour of Signal Processing. 2nd edn. Academic Press (2003)

Comparative Assessment of Interval and Affine Arithmetic in Neural Network State Prediction

Marcela Jamett and Gonzalo Acuña

Universidad de Santiago de Chile, USACH
Departamento de Ingeniería Informática
Av. Ecuador 3659, Postfach 917-0124. Santiago, Chile
{mjamett,gacuna}@usach.cl

Abstract. Two set theory methods, Interval and Affine Arithmetic, are used together with feedforward neural networks (FNN) in order to study their ability to perform state prediction in non-linear systems. Some fundamental theory showing the basic interval and affine arithmetic operations necessary to forward propagate through a FNN is presented and an application to a generic biotechnological process is performed confirming that due to the way the perturbations of the input data are considered, affine FNN perform better than interval ones.

1 Introduction

Automatic control of complex non-linear systems is frequently prevented because of the lack of sensors necessary to have on-line real-time measurements of the complete state vector of the process. Developing adequate state observers is also a very hard task for that kind of systems especially if the input data have high levels of disturbances. Good predictive models have to be used in order to adequately perform this task [1], [2].

Feedforward Neural Networks (FNN) and Recurrent Neural Networks have been recently used as a good tool for modeling and prediction [3], [4]. In this work set theory is introduced as a framework for developing good FNN predictive models which are able to cope with uncertainty and disturbances of the involved variables. Set theory basically consists in the representation of input and output data as belonging to a set with finite bounds [5]. The aim of this work is to study the performance of FNN coupled with two different set theory representations of data (Interval and Affine Arithmetic) for state prediction in nonlinear systems. The study consists on comparing Interval FNN (IFNN) and Affine FNN (AFNN) by means of an application to a generic fermentation process.

This paper is organized as follows. Chapters 2 and 3 present respectively the basic interval and affine arithmetic operations necessary to forward propagate through a FNN the adequately represented input data. In Chapter 4 an application to a generic fermentation process is developed in order to compare the prediction ability of interval nets and affine ones. Finally, in Chapter 5 some conclusions are drawn.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 448–453, 2005.

2 Interval Neural Networks

Hansen [6] proposes a way to operate with interval numbers, assuming that each variable is bounded by a lower and an upper real value. An interval number is defined as X=[a,b], i.e. the set $\{X\text{: } a \leq x \leq b\}$.

Interval FNN (IFNN) are especially suited to cope with input variables uncertainties when they are represented as intervals, because they can be easily propagated through the network using interval arithmetic (IA) in order to calculate the bound for the output variables.

2.1 Output Computation

A MISO IFNN is considered, its inputs are represented as intervals. The net has been previously trained using simulation data. The first step to compute the activation is (detailed arithmetic operations can be found in [6], [7]):

$$[v_{i1}, v_{i2}] = w_i\,[x_{i1}, x_{i2}] \quad \forall i = 1, \ldots, n \tag{1}$$

Then, the activation, considering bias, is :

$$[z_1, z_2] = \sum_{i=1}^{n} [v_{i1}, v_{i2}] + b \tag{2}$$

Using a nonlinear activation function (*tansig*) :

$$f(z) = \frac{2}{1 + e^{-2z}} - 1 \tag{3}$$

Which is computed following these steps:

$$\begin{array}{ll}
\underline{\textit{Subfunctions}} & \underline{\textit{Bounds}} \\
f_1(z) = -2z & [-2z_2, -2z_1] \\
f_2(z) = e^{f_1(z)} & [e^{-2z_2}, e^{-2z_1}] \\
f_3(z) = 1 + f_2(z) & [1+e^{-2z_2}, 1+e^{-2z_1}] \\
f_4(z) = \dfrac{1}{f_3(z)} & \left[\dfrac{2}{1+e^{-2z_1}}, \dfrac{2}{1+e^{-2z_2}}\right]
\end{array} \tag{4}$$

Finally, $f(z) = f_4(z)$ -1 and the network output bounds are :

$$y = \left[\frac{2}{1 + e^{-2z_1}} - 1, \frac{2}{1 + e^{-2z_2}} - 1\right] \tag{5}$$

3 Affine Neural Networks

Another appropriated method to deal with uncertain variables is that proposed by Stolfi [5]. Affine Arithmetic (AA) avoids the lost of information when two or more variables are dependent.

In AA, the value x is represented by a first order polynomial: $x_0 + x_1\varepsilon_1 + ... + x_k\varepsilon_k$. Where $x_0, x_1,..., x_k$ are real values and $\varepsilon_1,..., \varepsilon_k$ are uncertain variables in [-1,1]. Each variable ε_i represents an uncertainty source or error in the value of x_i.

There exists a relationship between IA and AA thus allowing to work under a unique framework.

3.1 Output Computation

As for the IFNN, we considered that the net's weights have been previously identified using simulation data. The output of a MISO AFNN is computed as a nonlinear function of the weighted sum of uncertain inputs (presented in their affine form).The operations defined in AA are: $z=\alpha x+\beta$, called affine functions.

The inputs are affine variables, defined in the form:

$$x_i = x_{i0} + x_{i1}\varepsilon_1 + ... + x_{in}\varepsilon_n \tag{6}$$

And, each x_i is an element of the input vector:

$$X = [x_1 \ x_2 \ \dots \ x_m]^T \tag{7}$$

Weights and biases are considered as real values, without error. They are expressed like:

$$\begin{gathered} W = [w_1 \ w_2 \ \dots \ w_m]^T \\ B = b \end{gathered} \tag{8}$$

Then, the neural network output is obtained as:

$$Y = f\left(\sum_{i=1}^{m} w_i (x_{i0} + x_{i1}\varepsilon_1 + ... + x_{in}\varepsilon_n) + b\right) \tag{9}$$

But first, the activation is considered:

$$z = \sum_{i=1}^{m} w_i (x_{i0} + x_{i1}\varepsilon_1 + ... + x_{in}\varepsilon_n) + b \tag{10}$$

This activation corresponds to an affine function of the kind $z=\alpha x+\beta$, then z is computed in an exact way:

$$z = w_1(x_{10} + x_{11}\varepsilon_1 + ... + x_{1n}\varepsilon_n) + ... + w_m(x_{m0} + x_{m1}\varepsilon_1 + ... + x_{mn}\varepsilon_n) + b \tag{11}$$

Nonlinear Functions. Working with AA, there is no way to compute nonlinear activation functions like *tansig*, exactly. However, an approximation is proposed. The approximation method used is Min Range [5].

$$f^* = \frac{2}{1+e^{-2z}} - 1 \tag{12}$$

The Min Range method allows linearly approximating a function if it is monotonic and decreasing. So, the *tansig* function should be separated into simpler sub functions. Then, it can be solved partially.

$f_1 = e^{-2z}$ (approximated by Min Range)
$f_2 = f_1 + 1$ (linear, exactly calculated)
$f_3 = 1/f_2$ (approximated by Min Range)
$f^* = 2f_3 - 1$ (linear, exactly calculated)

Finally, the FNN output is obtained in its affine form.

4 Application

A FNN, previously trained with simulation data, is used as a one-step ahead predictor for a nonlinear system. NN predictive ability is studied when the input data are uncertain. This uncertainty is represented as intervals and affine values.

4.1 Bioprocess Simulation

A generic fermentation process as described in [8] is used in order to compare IFNN and AFNN performance. Two state variables are considered, x being the biomass concentration and s the substrate concentration of the fermentation process whose specific kinetics is modeled by a Monod law in order to predict biomass concentration, which becomes the output of model.

State:

$$X = [x \quad s]' \tag{13}$$

State equations:

$$\begin{cases} \dot{x} = \mu_{\max} \dfrac{s}{K+s} * x - D \cdot x \\ \dot{s} = -\mu_{\max} \dfrac{s}{K+s} * \dfrac{x}{Y} + D \cdot (S_{in} - s) \end{cases} \tag{14}$$

Parameters: $\mu_{\max} = 0.4$; $K = 10$; $D = 0$; $Y = 0.8$; $S_{in} = 1$; $x_0 = 0.1$; $s_0 = 10$.

The states at instant k are denoted as x_k and s_k. They represent biomass and substrate respectively. x_{k+1} is the net's output, thus becoming the one step ahead prediction of biomass concentration. Direct process simulation gives the following state dynamics:

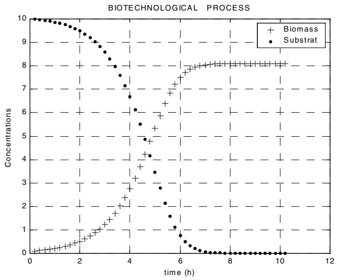


Fig. 1. Biomass and Substrate simulation.

NN Prediction. A FNN is trained with simulation data in order to predict the biomass evolution. Then, input data (biomass and substrate) is fed and propagated, using intervals and affine sets to represent them. A ±5% uncertainty is chosen (it could represent noise, perturbations, etc.). For IFNN, biomass and substrate are represented as $x=[0.95x,1.05x]$ and $s=[0.95s,1.05s]$ respectively. For AFNN, they are analogously represented as $x=x+0.05\varepsilon_1$ and $s=s+0.05\varepsilon_1$. Figures 2 and 3 show the state x evolution.

From figures 2 and 3 it can be seen that propagation of bounds with AA is better than that achieved with standard IA. This is due to a better representation of input data uncertainties considering interdependency of input variables thus avoiding overestimation of output bounds. Then, affine representation could be considered as more efficient than interval representation.

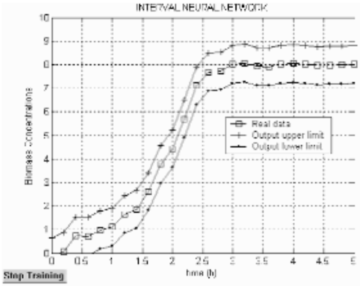


Fig. 2. IFNN output (biomass concentration) prediction, represented asan interval that contains the real data between its limits.

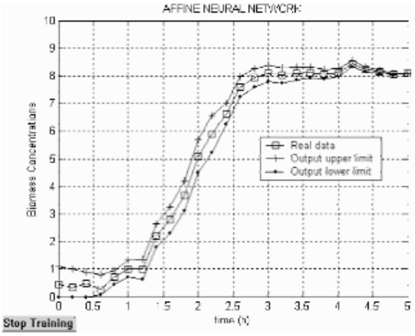


Fig. 3. AFNN output (biomass concentration) prediction, represented as theequivalent interval that contains the real data between its limits, but thinner than IFNN one.

5 Conclusions

FNN predictive capabilities are studied using set theory. The first step was the computation of a single neural output when inputs are uncertain, but bounded.

IA and AA are the tools used in this study. The first one works with interval variables as inputs, thus the output is obtained in a simple way, but the convergence cannot be guaranteed. The second method – AA – is more efficient because it avoids overestimation of the output bounds, considering the interdependency of input variables. This is verified by an application considering IFNN and AFNN as predictive models in a generic fermentation process.

Acknowledgements

The authors would like to thank partial financial support from Fondecyt (Chile) Grant 1010179 and ECOS-Conicyt (French Cooperation) Grant C99-B04. One of the authors, MJ, would like to thank Dr. Jean-Marie FLAUS and the Laboratoire d'Automatique de Grenoble, INPG-UJF, France for visits during July 2000 and March 2002.

References

1. Kavchak, M. and Budman, H.: Adaptive Neural Network Structures for Non-Linear Process Estimation and Control. Computers and Chemical Engineering. **23** (1999) 1209–1228
2. Chéruy, A. and Flaus, J.M.: Des Mesures Indirectes à l'estimation en Ligne. Capteur et mesures en Biotechnologie, chapter 12. Ed. Technique et Documentation – Lavosier, Paris France (1994)
3. Acuña, G., Latrille, E., Béal, C. and Corrieu, G.: Static and Dynamic Neural Networks Models for Estimating Biomass Concentration During Thermophilic Lactic Acid Bacteria Batch Cultures. Journal of Fermentation and Bioengineering, **85** (1998) 615–622
4. Boillereaux, L. and Flaus, Jean-Marie: A New Approach for Designing Model-Based Indirect Sensors. IEEE Transactions on Control Systems Technology, **8** (2000) 601–608
5. Stolfi, Jorge y Figuereido, Luiz: Self–Validated Numerical Methods and Applications. 21st. Brazilian Mathematics Colloquium. IMPA (1997)
6. Hansen, E.: Global Optimization Using Interval Analysis. Ed. Marcel Deker, Inc. New York, U.S.A. (1992)
7. Valdés-González, H., J-M. Flaus, G. Acuña: Moving Horizon State Estimation with Global Convergence Using Interval Techniques: Application to Biotechnological Processes. Journal of Process Control, **13** (2003) 325–336
8. Bastin, G., Dochain, D.: On-line Estimation and Adaptive Control of Bioreactors. Elsevier Science Publishing Co., Amsterdam (1990)

Identification of Duffing's Equation with Dynamic Recurrent Neural Network

Shan Liang[1], Qin Zhu[2], and Mitsuaki Ishitobi[3]

[1] Chongqing University, Chongqing 400044, China
lightsun@cqcnc.com
[2] Oyama National College of Technology, Tochigi 328-0808, Japan
zhu@mech.kumamoto-u.ac.jp
[3] Kumamoto University, Kumamoto 860-8555, Japan

Abstract. In this paper, the identification of Duffing's equation by dynamic recurrent neural network (DRNN) is discussed. It is assumed that only states of the Duffing's equation are available and the system is taken as a black box. It is shown that the DRNN can model the dynamics of the Duffing's equation. A numerical example indicates that the forced training method can be used to train the DRNN.

1 Introduction

A classic nonlinear differential equation that has been used to model the nonlinear dynamics of mechanical and electrical systems is the harmonic oscillator with a cubic nonlinearity [1]

$$\ddot{x} + \delta\dot{x} + \alpha x + \beta x^3 = \eta \cos \omega t \ . \tag{1}$$

With $\alpha = 0$, it is a model for a circuit with a nonlinear inductor and it is also a model for cantilever beam subjected to magnetic force [2]. Since this equation is a generalized expression for single degree of freedom system in electromechanical engineering, many approaches for controlling the system have been studied which are based on the assumption that its parameters are known or with uncertainty. However, the calculation or identification of the parameters δ, α and β in this equation is not easy especially when they are related to magnetic force and elastic deformation.

On the other hand, the advance in studying artificial neural network has provided a powerful modeling tools for the control of nonlinear systems. It was found that the feed-forward neural network can give a satisfactory model of the Duffing's equation [3]. Although this method is powerful, the disadvantage is that it is difficult to be included in the nonlinear control analysis using differential geometric and algebraic techniques. In this article, we show that the dynamic recurrent neural network (DRNN) can be trained to generate the state of Duffing's equation. Since DRNN is expressed by a set of differential equations, the DRNN model of the Duffing's equation can be used in design the control system within a differential geometric framework [4].

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 454–459, 2005.

2 DRNN and Its Application in Modelling Duffing's Equation

2.1 Dynamic Recurrent Neural Network

The architecture of dynamic neural network is shown in Fig. 1. For a dynamic neuron, its output is described by a differential equation [5][8]

$$\dot{X}_i = -X_i + \sum_{j=1}^{N} W_{ij}\sigma(X_j) + \gamma_i u \quad . \tag{2}$$

A DRNN is constructed by dynamic neurons with forward and backward connections which is shown in Fig. 1(b). The following lemma shows that it can also be used to model the Duffing's equation.

Lemma 1. *The Duffing's system with control input u*

$$\ddot{x} + \delta\dot{x} + \alpha x + \beta x^3 = \eta\cos\omega t + u \quad , \tag{3}$$

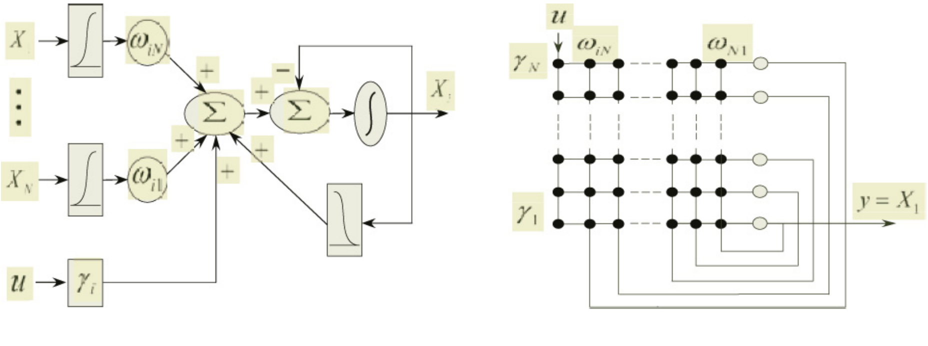
can be approximated by the dynamic recurrent neural network

$$\dot{\boldsymbol{X}} = -T\boldsymbol{X} + W\sigma(\boldsymbol{X}) + \boldsymbol{\Gamma}U \quad , \tag{4}$$

where $\boldsymbol{X} \in R^n$, $T \in R^{N\times N}$, $W \in R^{N\times N}$, $\sigma(\boldsymbol{X}) = [\sigma(X_1), \cdots, \sigma(X_N)]^T$, $\boldsymbol{\Gamma} \in R^{N\times 1}$ and $U = \eta\cos\omega t + u$. The number of units $N \geq 2$.

Proof. The Duffing's equation (3) in which the exciting term $\eta\cos\omega t$ is the time function can be rewritten in compact form by introducing a new control input term $U = \eta\cos\omega t + u$

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}U \quad , \\ y &= h(\boldsymbol{x}) \quad . \end{aligned} \tag{5}$$



(a) Dynamic neuron model (b) DRNN

Fig. 1. Architecture of dynamic neural network

Notice that the system (5) is a nonlinear-control affine system. It has been proved that a DRNN can approximated control affine systems [5]. In the same way, the system (5) is expressed as

$$\begin{aligned} \dot{\boldsymbol{x}} &= -A\boldsymbol{x} + \boldsymbol{f}_0(\boldsymbol{x}) + \boldsymbol{g}U \quad , \\ y &= h(\boldsymbol{x}) \quad , \end{aligned} \tag{6}$$

where $\boldsymbol{f}_0(\boldsymbol{x}) = A\boldsymbol{x} + \boldsymbol{f}(\boldsymbol{x})$ and $A \in R^{2\times 2}$. It is known that the function $\boldsymbol{f}_0(\boldsymbol{x})$ can be approximated by a feed forward neural network as [6]

$$\boldsymbol{f}_0(\boldsymbol{x}) = C\boldsymbol{\sigma}(D\boldsymbol{x} + \theta) \quad . \tag{7}$$

Taking (7) into (6)

$$\begin{aligned} \dot{\boldsymbol{x}} &= -A\boldsymbol{x} + C\boldsymbol{\sigma}(D\boldsymbol{x} + \theta) + \boldsymbol{g}U \quad , \\ y &= h(\boldsymbol{x}) \quad . \end{aligned} \tag{8}$$

Define a new variable $\boldsymbol{\xi} = D\boldsymbol{x} + \theta$ and substituting it into (8) and then (8) into $\dot{\boldsymbol{\xi}} = D\boldsymbol{x}$, the following equations is obtained

$$\begin{aligned} &\begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{\xi}} \end{bmatrix} = -\begin{bmatrix} A & 0 \\ DA & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\xi} \end{bmatrix} + \begin{bmatrix} 0 & C \\ 0 & DC \end{bmatrix} \begin{bmatrix} \boldsymbol{\sigma}(\boldsymbol{x}) \\ \boldsymbol{\sigma}(\boldsymbol{\xi}) \end{bmatrix} + \begin{bmatrix} \boldsymbol{g} \\ D\boldsymbol{g} \end{bmatrix} U \quad , \\ &y = h(\boldsymbol{x}) \quad . \end{aligned} \tag{9}$$

Since (9) can be rewritten in the expression of DRNN, then we have

$$\dot{\boldsymbol{X}} = -T\boldsymbol{X} + \sum_{j=1}^{N} W_{ij}\sigma(\boldsymbol{X}_j) + \boldsymbol{\Gamma}U \quad , \tag{10}$$

$$y = h(\boldsymbol{x}) \quad , \tag{11}$$

where

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\xi} \end{bmatrix} \quad , \qquad T = \begin{bmatrix} A & 0 \\ DA & 0 \end{bmatrix} \quad , \qquad W = \begin{bmatrix} 0 & C \\ 0 & DC \end{bmatrix} \quad , \qquad \boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{g} \\ D\boldsymbol{g} \end{bmatrix} \; .$$

with $\boldsymbol{X} \in R^N, T \in R^{N\times N}, W \in R^{N\times N}, \boldsymbol{\Gamma} \in R^{N\times 1}, \boldsymbol{\sigma}(\boldsymbol{x}) = \{\boldsymbol{\sigma}(\boldsymbol{x}_1), \cdots, \boldsymbol{\sigma}(\boldsymbol{x}_N)\}$ and the total number of neurons is $N \geq 2$.

Q.E.D

Furthermore the network expressed in (4) with $T = I$ and no restriction on the weight matrix W is known as the generalized Hopfield network. It is expressed as

$$\dot{\boldsymbol{X}} = -\boldsymbol{X} + W\sigma(\boldsymbol{X}) + \boldsymbol{\Gamma}U \quad . \tag{12}$$

In this study, the network in (12) will be used since it is more general and can be trained with less time.

2.2 Learning Algorithm

There are several methods available for training DRNN [7]. Here the forced training method used in the study is briefly described as follows. The advantage of the method is that it may reduce the learning time by adding an additional force to neuron model.

Suppose J_i is additional training force to ith neuron which has teaching signal Q_i. Then the input-output relation of the neuron becomes

$$\frac{dX_i}{dt} = -X_i + \sum_{j=1}^{N} W_{ij}\sigma(X_j) + \gamma_i U + J_i \quad . \tag{13}$$

The basic idea is that, J_i is selected such that the output X_i is forced to approach the teaching signal Q_i. The J_i increases or decreases with the increasing or decreasing of error $|X_i - Q_i|$. Therefore, for $X_i = Q_i$ at the time $t \geq t_1$, we need $X_i(t_1) = Q_i(t_1)$ and

$$\frac{dX_i}{dt} = \frac{dQ_i}{dt} \quad . \tag{14}$$

Thus the additional force can be calculated by

$$J_i = \frac{dQ_i}{dt} + Q_i - \sum_{j=1}^{N} W_{ij}\sigma(Q_j) - \gamma_i U \quad . \tag{15}$$

Then the objective of learning is to find an optimal value of W and Γ that minimizes the energy function E_F which is defined as

$$E_F(t_1, t_2) = \int_{t1}^{t2} \left(\sum_{i=1}^{N_t} \frac{J_i^2}{2} \right) dt \quad , \tag{16}$$

where N_t is the number of neurons with teaching signal. Thus the W and Γ can be calculated with back-propagation algorithm [8][9].

Since $E_F(t_1, t_2)$ is the function of weight W_{ij}, its variation ΔW_{ij} is chosen such that the energy function E_F decreases. Then

$$\Delta E = E(W + \Delta W) - E(W) \approx \sum_{i \in Cj} \frac{\partial E}{\partial W_{ij}} \Delta W_{ij} \leq 0 \quad . \tag{17}$$

Suppose κ is a positive and small number, we chose

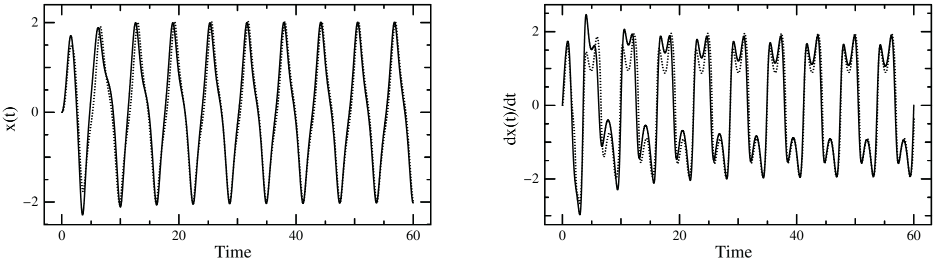
$$\Delta W_{ij} = -\kappa \frac{\partial E}{\partial W_{ij}} \quad . \tag{18}$$

Therefore, the weighting matrix can be renewed by

$$\Delta W_{ij}(n+1) = -\kappa \frac{\partial E}{\partial W_{ij}} + \alpha \Delta W_{ij}(n) \quad , \tag{19}$$

$$\Delta \gamma_i(n+1) = -\theta \frac{\partial E}{\partial \theta_i} + \beta \Delta \theta_i(n) \quad , \tag{20}$$

where n is the number for renew of the weight W, α, θ and β are small positive number for the learning process.



(a) $x(t)$

(b) $\dot{x}(t)$

Fig. 2. Comparison of output of Duffing's equation and its DRNN model ($-$: output of Duffing's eq.; $\cdots$: output of DRNN)

3 A Numerical Example

It is shown in Lemma 1 that the Duffing's equation can be modeled by DRNN. As an example, the following system was identified with a DRNN.

$$\ddot{x} + 0.5\dot{x} + 3x^3 = \cos t \quad . \tag{21}$$

It was assumed that, the system was a black box and only the state $x(t)$ and the forcing function $\cos t$ were available. Therefore the displacement and the forcing term were selected as the teaching signals. The input-output data which were generated by (21) were sampled at the sampling frequency $2\pi/50$ during the time $0 \sim 400$.

The DRNN with five neurons and $\sigma(\boldsymbol{X}_j) = \tanh(\boldsymbol{X}_j)$ were selected to approximate the Duffing's equation. Figure 2 shows the output of (21) and the output of the trained DRNN model. In Figure 2(a), the solid line is the displacement $x(t)$ which is generated by direct integration of equation (21) and the dotted line is the displacement $x(t)$ given by the trained DRNN model. The comparison of $\dot{x}(t)$ given by (21) and by the trained DRNN model is shown in Figure 2(b). Although the state $\dot{x}(t)$ is not included in the teaching signal, the trained DRNN mode provides the approximation of $\dot{x}(t)$ satisfactorily.

4 Concluding Remarks

It is shown that the Duffing's equation can be modeled by dynamic recurrent neural network. The forced training method can be used in training the DRNN. Since the output of the DRNN is described by a differential equation, the DRNN model can be used as an alterative of the real system for design of the control system within a differential geometric framework.

Acknowledgment

The work described in this paper was partly supported by Project 60375024 of NSFC.

References

1. Moon, F., C.: Chaotic and Fractal Dynamics, John Willey & Sons, Inc. (1992)
2. Zhu, Q., Tani, J., Ohtomo, K.: Chaotic Vibration and Its Control of a Cantilever Beam Subjected to Magnetic Force, Advanced Computational and Design Techniques in Applied Electromagnetic Systems, S. -y. Hahn (Editor), Elsevier, (1995) 333–336
3. Zhu, Q., Ishitobi, M.: Identification of System with Chaotic Response by Neural Network (in Japanese). Proc. of JSME Conference, No. 968-2, (1996) 114–115
4. Slotine, J., J., E., Li, W.: Applied Nonlinear Control, Prentice Hall (1993)
5. Delgado, A., Kambhampati C, Warwick, K.: A Dynamic Recurrent Eural Network for System Identification and Control. IEE Proc. - Control Theory and Appl., **142** (1995) 307–314
6. Funahashi, K., I.: Approximation of Dynamic Dystems by Contimuous Time Recurrent Neural Networks. Neural Networks, **6** (1993) 801–806
7. Pearlmutter, B., A.: Gradient Calculations for Dynamic Recurrent Neural Networks: A Survey. IEEE Trans. on Neural Networks, **6** (1995) 1212–1228
8. Aihara, I., et al: Chaos in Neural Systems (in Japanese), Press of Tokyo Denki Univ., (1993)
9. Kumpati, S., N.: Identification and Control of Dynamical Systems Using Neural Networks. IEEE Trans. on Neural Networks, **1** (1990) 4–27

An Intelligent System for Dynamic System State Forecasting

Wilson Wang

Mechanical Engineering, Lakehead University
Thunder Bay, Ontario P7B 5E1, Canada
Wilson.Wang@Lakeheadu.ca

Abstract. In this paper, an adaptive neuro-fuzzy (NF) forecasting system is proposed, and its robustness is investigated experimentally. After the NF predictor is initially trained using a data set from the Mackey-Glass differential equation, it is implemented for two applications, an online gear system condition monitoring and a material fatigue testing to forecast future states of a fatigue crack propagation trend in test specimens. From the forecasting tests and simulation analyses, it is found that the developed NF system is a very reliable prognostic scheme; it can capture system dynamic behavior quickly, and track system responses accurately.

1 Introduction

State forecasting of a dynamic system entails the current and previous conditions to forecast the future states of the dynamic system. In a machine, for example, a reliable prognostic system can predict the fault propagation trends, and provide an accurate alarm before a fault reaches critical levels so as to prevent machinery performance degradation, malfunctions, and even catastrophic failures. In practice, the temporal patterns utilized can be obtained from vibration features, acoustic signals, or the debris in the lubrication oil. Vibration-based monitoring, however, is a well-accepted approach due to its ease of measurement and analysis, and accordingly it is used in this study.

Currently, several vibration-based techniques have been proposed in the literature for time series prediction [1]. The classical approaches are mainly based on the use of stochastic models; generally, these autoregressive approaches are rather difficult to implement to forecast the response of complex dynamic systems. Recent interests focus on the use of different types of neural networks (NNs) [2]. Research results show that the predictor based on recurrent NNs perform better than those based on the general feedforward NNs [3],[4]. The disadvantages of NNs, however, are that their internal layers are opaque to users, and the convergence of learning is usually slow and not guaranteed. A solution to overcome these limitations is to use synergetic systems, for example, neuro-fuzzy (NF) schemes, in which the NNs provide fuzzy systems with learning abilities whereas fuzzy logic provides NNs with structural framework and high-level fuzzy thinking and reasoning [5]. From the author's prior

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 460–465, 2005.

investigation [6], it has been found that when an NF scheme is properly trained, it performs even better than recurrent NNs in time series forecasting. The aim of this paper is to assess the robustness of the NF predictor by implementing it for different type of prognostic applications in different environments.

In this paper, a brief description of an NF system is given. This system is implemented for gear system prognosis and material fatigue testing. Results show that the developed NF predictor is a reliable and robust predictor. It can capture system dynamic behavior quickly and then track system characteristics accurately.

2 Neuro-Fuzzy Predictor

NF predictor is a neural network-based fuzzy prediction system. The forecast reasoning is performed by fuzzy logic, whereas fuzzy system parameters are trained by using NNs. In this case, four time steps, $\{x_{t-3r}\ x_{t-2r}\ x_{t-r}\ x_t\}$, are chosen for one-step prediction (i.e., r =1). If two membership functions (MFs), *Small* (*S*) and *Large* (*L*), are assigned to each input variable, 16 rules are formulated for this forecasting operation. For notational simplicity, these rules are represented in a general form:

$$\Re_j: \text{ If } (x_{t-3r}\ is\ A_1^j) \text{ and } (x_{t-2r}\ is\ A_2^j) \text{ and } (x_{t-r}\ is\ A_3^j) \text{ and } (x_t\ is\ A_4^j)$$

$$\text{then } x_{t+r} = c_1^j x_{t-3r} + c_2^j x_{t-2r} + c_3^j x_{t-r} + c_4^j x_t + c_5^j . \qquad (1)$$

where x_{t+r} is the forecasted future state value, A_i^j are the MFs, j = 1, 2, …, 4, j = 1, 2, …, 16. The corresponding NF network architecture is shown in Fig 1.

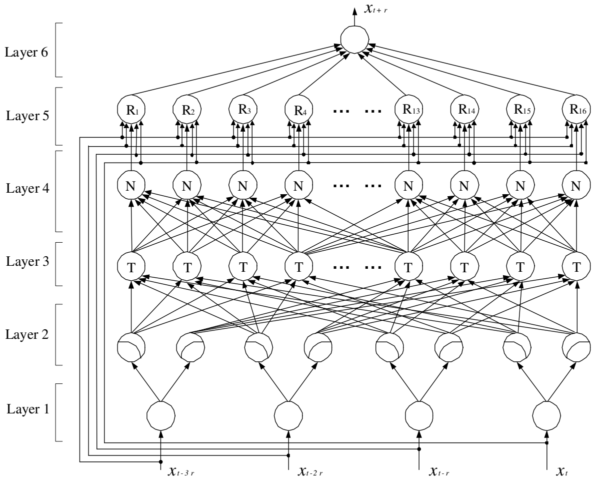


Fig. 1. The network architecture of the NF forecasting scheme

This is a six-layer feedforward network. The links, with unity weights, represent the flow direction of signals between nodes. The nodes in layer 1 only transmit input signals to the next layer directly. In layer 2, each node acts as an MF, *Small* or *Large*.

Through tests, sigmoid functions are chosen as MFs in this case, which are expressed as

$$\mu_{A_k^j}(x_k) = \frac{1}{1 + exp(-b_k^j(x_k - m_k^j))} \,. \tag{2}$$

$k = 1, 2, \ldots, 4$; $j = 1, 2, \ldots, 16$. The parameters m_k^j and b_k^j are determined by training. In layer 3, each node performs a fuzzy *T*-norm operation. If max-product operator is applied, the firing strength of rule $\Re_j$ is obtained by

$$\mu_j = \mu_{A_1^j}(x_{t-3r})\mu_{A_2^j}(x_{t-2r})\mu_{A_3^j}(x_{t-r})\mu_{A_4^j}(x_t)\,. \tag{3}$$

All the nodes in layer 3 form the forecasting rule base. The rule firing strengths (3) are normalized in layer 4. After linear combination in layer 5, the predicted output, x_{t+r}, is determined in layer 6 using centroid defuzzification; that is,

$$x_{t+r} = \frac{\sum_{j=1}^{16} \mu_j(c_1^j x_{t-3r} + c_2^j x_{t-2r} + c_3^j x_{t-r} + c_4^j x_t + c_5^j)}{\sum_{j=1}^{16} \mu_j} \,. \tag{4}$$

The above stated NF system has 96 (16 premise and 80 consequent) parameters to be optimized in training. In order to improve the training efficiency and eliminate the possible trapping due to local minima, a hybrid learning algorithm is utilized in this case [6]. In the forward pass of each training epoch, the consequent parameters, (c_1^j, c_2^j, c_3^j , c_4^j , c_5^j), are trained using the extended Kalman filter approach, whereas the premise MF parameters are updated by the gradient method in the backward pass.

3 Gear System Monitoring

Gear systems (like gearboxes) are widely used in industry; typical applications include airplanes, automobiles, and power turbines. The developed NF predictor is implemented in this study for online gear system condition forecasting. The tests are conducted by using the same experimental setup as illustrated in [7]. The analysis is based on a monitoring index from the continuous wavelet amplitude; the details of its derivation can be found in [8]. All the signal processing techniques utilized in this study and the NF prognostic system are coded in MATLAB environment. In this case, the NF predictor is only trained using a data set from the Mackey-Glass differential equation [6], which is a benchmark series frequently utilized to evaluate forecasting models. The test is conducted at the gear speed ranging from 400 rpm to 1200 rpm. To simulate the practical working conditions [9], the load level and rotation speed are randomly modified, for short time spans, in the range of ± 20% of the specifications. Every 10 minutes, the NF predictor is automatically applied to forecast future states of the health condition of the gear of interest.

A pair of new gears is first mounted and tested. After about 155 samples (about 26 hours), one tooth in the gear of interest is filed in the middle with about 15% of the tooth surface area removed to simulate a spalling damage. Then the test resumes and continues until the damaged tooth is partly broken off about 16 hours later. Fig. 2 illustrates the online forecasting result. It is clear that the prediction matches the actual state accurately during the period of the gear's healthy state (early section). After the simulated tooth fault is introduced, the predictor takes only a few samples to catch up the new dynamic characteristics of the system, and then exactly follows the fault propagation trend until the damaged tooth is broken off.

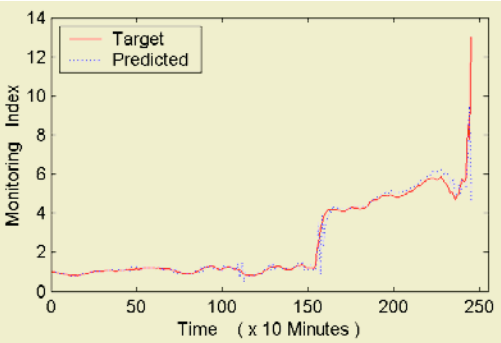


Fig. 2. Filed gear tooth prognostic result

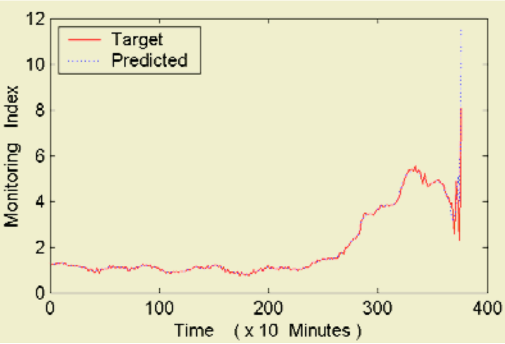


Fig. 3. Cracked gear tooth prognostic result

Next, another pair of healthy gears is mounted and tested. After 230 time samples (about 38 hours), a transverse cut with 10% of root thickness is initiated at one tooth of the gear of interest to simulate a fatigue crack. Then the test resumes, and the one-step prediction continues until the damaged gear tooth is broken off. Fig. 3 graphs the real-time forecasting results. It can be seen that the NF predictor works accurately during the test. It captures the system dynamic response perfectly except for some minor fluctuations prior to tooth failure; that is because during that period, the gear meshing stiffness becomes unsteady.

4 Fracture Test Forecasting

The generality of the developed NF prognostic scheme is evaluated in this section, by applying it to other cases represented by different scales and monitoring indices. A material fatigue property test is usually a very time consuming procedure. A reliable forecasting scheme is very useful to effectively plan the test processes and significantly shorten the experimental time. Here the NF system, initially trained by a Mackey-Glass data set, is applied to forecast the fatigue crack propagation trend. The test is conducted on a specimen with an initial crack. Fig. 4 shows the forecasting result of a crack propagation trend represented by a relative voltage measurement. Initially (up to 4×10^6 cycles), the predictor does not capture the system response reasonably. That is due to the errors induced by the measurement noise. This error can be corrected by updating the NF scheme using representative training data sets.

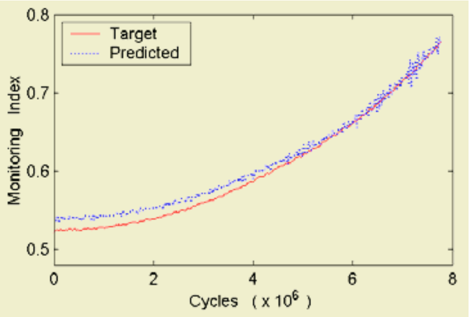


Fig. 4. Forecasting result for the crack propagation trend in material specimen I

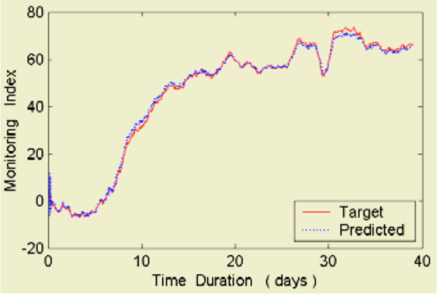


Fig. 5. Forecasting result for the crack propagation trend in material specimen II

Figure 5 illustrates the material fatigue crack propagation test for a different specimen. It is seen that, even without further training using representative data sets, the developed NF predictor can also effectively capture the system characteristics and then track the system dynamic response accurately. Without doubt, the forecasting performance will be further improved if the NF predictor is properly trained by using representative data sets corresponding to specific applications.

5 Conclusions

In this paper, an adaptive neuro-fuzzy system is developed for dynamic system state forecasting. Its forecasting performance is evaluated by online tests corresponding to gear system condition monitoring and material fatigue propagation forecasting. It has been found that even though the NF prognostic system is initially trained only using a general Mackey-Glass data set yet, it can capture and track system dynamic behavior effectively and accurately. In addition, the developed NF predictor is also a robust forecasting scheme. It can be implemented, directly, to other applications represented by different monitoring indices and scales. Advanced researches are currently undertaken to develop novel online/offline training techniques to further improve the performance of the developed NF predictors regarding the rapid convergence as well as multi-step forecasting operations.

References

1. Rhmadi, M.: Foundations of Time Series Analysis and Prediction Theory. John Wiley & Sons, Inc (2001)
2. Y, A., El-Shoura, S., Shaheen, S., El-Sherif, M.: A Comparison Between Neural-network Forecasting Techniques-case Study: River Flow Forecasting. IEEE Transactions on Neural Networks, **10** (1999) 402-409.
3. Nr, J., Martin, R. and Atlas, L.: Recurrent Neural Networks and Robust Time Series Prediction. IEEE Transactions on Neural Networks, **5** (1994) 240-254
4. Atherton, D.: Prediction of Machine Deterioration Using Vibration Based Fault Trends and Recurrent Neural Networks. Journal of Vibration and Acoustics, **121** (1999) 355-362
5. Sun, C. and Mizutani, E.:Neuro-fuzzy and Soft Computing. Prentice Hall, Inc. (1997)
6. Wang, W., Golnaraghi, F., and Ismail, F.: Prognosis of Machine Health Condition Using Neuro-fuzzy Systems. Mechanical Systems and Signal Processing, **18** (2004) 813-831
7. Wang, W., Ismail, F., and Golnaraghi, F.: Assessment of Gear Damage Monitoring Techniques Using Vibration Measurements. Mechanical Systems and Signal Processing, **15** (2001) 905-922
8. Wang, W., Ismail, F., and Golnaraghi, F.: A Neuro-fuzzy Approach for Gear System Monitoring. IEEE Transactions on Fuzzy Systems, **12** (2004) 710-723
9. Wang, W., Golnaraghi, F., and Ismail, F.: Condition Monitoring of a Multistage Printing Press. Journal of Sound and Vibration, **270** (2004) 755-766

Sequential Extraction Algorithm for BSS Without Error Accumulation

Qiang Liu and Tianping Chen[⋆]

Laboratory of Nonlinear Science, Institute of Mathematics
Fudan University, Shanghai 200433, China
tchen@fudan.edu.cn

Abstract. Blind source separation (BSS) is an emerging research field in both theory and applications. In this paper we propose a kurtosis maximization algorithm–Sequential Extraction Algorithm, which can extract the source signals sequentially. This approach is based on an algorithm for separating one signal (Algorithm 1) and some technique to eliminate the accumulating errors, which often occur in the sequential extraction steps. In Algorithm 1, a new criterion to judge whether the separated signal is an original source signal, is proposed. In Sequential Extraction Algorithm, we propose a new approach to eliminate accumulating errors, which is caused in the sequential extraction process. This approach is based on the cost function involved in this algorithm, and thus, is different from those available in literature.

1 Introduction

Blind source separation is a technique to recover a number of originally independent, random or not, signals only from linear mixture of them. Compared with the other techniques for BSS, Kurtosis Maximization Algorithm (KMA) is much easier in computation.

Since KMA has many equilibria, criteria to judge whether the recovered signals are source signals become increasingly significant. In [1], the authors pointed out the relation between local maximal points of the cost function given in their paper and source signals. But three facts undermine applications of their discovery: Fist, it is complicated to determine whether the equilibrium point is a local maximal point in computation; Second, in their paper, in extracting second source signal in sequential extraction, it is assumed the errors are gaussian in their paper. However, this assumption is inappropriate. Because after subtracting the first recovered signal, in the remainders there is some element of the first source signal, which will seriously affect separation of other source signals; Thirdly, error accumulation (errors in the separated signals will accumulate greatly in later separation) will badly affect the recovery of remaining signals.

In this paper, we focus on the three points mentioned above and propose a new Sequential Extraction Algorithm. First, we will present an easier criterion

[⋆] Corresponding author

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 466–471, 2005.

to determine whether the recovered signal is one of the original source signals. If it is not a source signal, the criterion also suggests a way to proceed extraction. Then we present a new approach to tackle the accumulating errors.

2 Problem Formulation

Let $s_i(k)$ $(i = 1, \cdots, n)$ be n independent source signals denoted by $S(k) = [s_1(k), \cdots, s_n(k)]^T$. The observed linear mixture $X(k) = [x_1(k), \cdots, x_n(k)]^T$ is represented as

$$X(k) = AS(k) \tag{1}$$

where $A = \{a_{ij}\}_{n\times n} = [a_1, \cdots, a_n]$ is an unknown nonsingular matrix. In this paper, we assume: (1) signals are real, $E\{S\} = 0$, and $E\{SS^T\} = I$; (2) A is an orthogonal matrix.

3 Main Results

3.1 Algorithm for Extracting One Source Signal

In this subsection, we propose a new criterion to determine whether a separates signal is a source signal. By this criterion, we propose **Algorithm 1**.

We take absolute value of kurtosis as our cost function. Kurtosis of signal s_i is defined as

$$kurt(s_i) = E\{s_i^4\} - 3 \tag{2}$$

Let

$$y = w^T X = v^T S \tag{3}$$

where $v^T = w^T A$.

Definition 1 $\bar{w}$ is a **separating** vector, if $\bar{w}^T X$ is one of the source signals.

We separate one signal by maximizing

$$J_{kma}(w) = |kurt(y)| = |kurt(w^T X)| = |kurt(v^T S)| \tag{4}$$

under constraint $\|w\| = 1$.

Following algorithm is a natural gradient algorithm on the the unit ball:

1. Pick initial weight w;
2. Calculate $grad\big(J_{kma}(w)\big) = sign(kurt(w^T X))[I_n - ww^T]E\{(w^T X)^3 X\}$;
3. Update w by
   $w = w + \eta \cdot grad\big(J_{kma}(w)\big)$, and $w = w/\|w\|$;

We can prove

Theorem 1 Above algorithm is stable at any separating vector $\bar{w}$.

Now, we will present a new criterion to determine whether a separates signal is a source signal. Before given the criterion, we need two lemmas.

Lemma 1 (see [1]) An equilibrium point $\bar{v}$ is a local maximal point of the following extremal problem:

$$maximizing\ J_{kma}(v) = |kurt(y)| = |kurt(v^T S)|,\ under\ \|v\| = 1 \tag{5}$$

if and only if there is an e_i such that $\bar{v} = e_i$.

We introduce following two Hessin-like matrices, which are essential in our criterion.

$$H(w) = \frac{\partial\{grad\big(J_{kma}(w)\big)\}}{\partial w^T} = sign(kurt(w^T X))\Big[3E\{(w^T X)^2 X X^T\} \\ -E\{(w^T X)^4\}I - 4wE\{(w^T X)^3 X^T\}\Big] \tag{6}$$

$$L(v) = \frac{\partial\{grad\big(J_{kma}(v)\big)\}}{\partial v^T} = sign(kurt(v^T S))\Big[3E\{(v^T S)^2 S S^T\} \\ -E\{(v^T S)^4\}I - 4vE\{(v^T S)^3 S^T\}\Big] \tag{7}$$

It can be seen that

$$H(w) = AL(v)A^T \tag{8}$$

Lemma 2 If $\bar{v}$ is an equilibrium point for $grad\big(J_{kma}(v)\big)\}$. Then, $L(\bar{v})$ is a symmetric matrix. Therefore, $H(\bar{w})$, where $\bar{w} = A\bar{v}$, is symmetric, too.

Let $H(\bar{w}) = QDQ^T$ be the Schur decomposition of $H(\bar{w})$, where $Q = [q_1, \cdots, q_n]$ and $D = diag\{d_1, \cdots, d_n\}$. The new criterion is presented by the following theorem.

Theorem 2 Equilibrium $\bar{w}$ is not a separating vector, if and only if $H(\bar{w})$ has an eigenvalue $d_k > 0$ such that $\|q_k \times \bar{w}\| \neq 0$, where q_k is the eigenvector corresponding to the eigenvalue $d_k > 0$, where $q_k \times \bar{v}$ is the outer product from q_k to $\bar{v}$.

Based on previous theoretical analysis, we propose following algorithm to extract one source signal:

Algorithm 1:

```
1. Pick initial weight $w$, and calculate $grad\big(J_{kma}(w)\big)$;
2. While $||grad\big(J_{kma}(w)\big)|| > \lambda$, update $w$ by
   $w = w + \eta \cdot grad\big(J_{kma}(w)\big)$, and $w = w/\|w\|$, and calculate $grad\big(J_{kma}(w)\big)$;
3. Assume we obtain equilibrium $\bar{w}$;
4. Calculate $H(\bar{w})$;
5. Shur decomposition of $H(\bar{w})$ is $H(\bar{w}) = QDQ^T$ where $Q = [q_1, \cdots, q_n]$, and
   $D = diag\{d_1, \cdots, d_n\}$;
6. If there is $d_k > 0$, and $\|q_k \times \bar{w}\| > \epsilon$, proceed; if not, separation is ended;
7. $w_1 = \bar{w} + \eta_1 \cdot q_k$, and let initial vector $w = \frac{w_1}{\|w_1\|}$, go back to obtain another
   equilibrium.
```

3.2 Sequential Extraction Algorithm

In this subsection, we present an algorithm to extract source signals sequentially. The key point is how to eliminate errors accumulated in the extraction process.

Sequential Extraction Algorithm is:

1. Calculate $\hat{A} = E\{X[y_1, \cdots, y_{p-1}]\}$, and $X - E\{X[y_1, \cdots, y_{p-1}]\}[y_1, \cdots, y_{p-1}]^T$. Denote $\bar{X} = [x_1^p, \cdots, x_n^p]^T = X - E\{X[y_1, \cdots, y_{p-1}]\}[y_1, \cdots, y_{p-1}]^T$;
2. Rearrange rows of $\hat{A}$ according to the descending order of the norm;
3. Pick $\hat{X}^{p-1} = \{x_i^p \mid x_i^p \in \bar{X}, i \notin M\}$, where M is the first $p-1$ linearly independent rows in $\hat{A}$, and whitening $\hat{X}^{P-1}$ by $X^{P-1} = C_{p-1}\hat{X}^{p-1}$;
4. Extract $\bar{y}_p$ from X^{p-1} by Algorithm 1;
5. Take $w = E\{\bar{y}_p X\}$ as initial weight in Algorithm 1, extract y_p from X.

Remark If we can extract $\bar{y}_p$ from X^p accurately, step 5 is unnecessary. However, in practice, due to the accumulating errors it is impossible to extract $\bar{y}_p$ from X^p accurately. Thus, we use step 5 to guarantee the extracted signal y_p is one of the source signals accurately.

To ensure that $\hat{X}^P$ is linearly independent, we need following proposition, which is used in Step 3 to define new mixture.

Proposition $A = \begin{pmatrix} A_1 & A_2 \\ A_3 & A_4 \end{pmatrix}$ is orthogonal, where A_1 and A_4 are square matrixes. If A_1 is nonsingular, A_4 is nonsingular.

4 Numerical Illustrations

In this section, we will show the effect of Sequential Extraction Algorithm by two simulation experiments. The source signals involved in this section are from Matlab toolbox or generated by Matlab tool.

Example 1: Four source: three are audio signals from matlab toolbox: chirp, laughter, and splat. The forth is $s_4 = 3sin(n_4(t) - 0.5)$ ($n_4(t)$ is a random vector in [0,1]). Of course, we whiten these signals first. And the corresponding kurtoses of the whitened signals are 3.9315, 1.5960, 18.4904 and -1.3084. Mixture matrix is:

$$A = \begin{pmatrix} -0.5711 & -0.5598 & -0.2505 & -0.5456 \\ 0.2503 & -0.1383 & -0.9107 & 0.2980 \\ 0.5109 & 0.3150 & -0.1636 & -0.7829 \\ -0.5917 & 0.7538 & -0.2847 & -0.0233 \end{pmatrix} \tag{9}$$

By the Sequential Extraction Algorithm proposed before, we get

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} -0.0010 & 0.0065 & -0.9999 & -0.0004 \\ -0.9998 & 0.0045 & -0.0014 & 0.0149 \\ -0.0331 & 0.9992 & 0.0112 & -0.0152 \\ 0.0066 & -0.0067 & 0.0038 & -0.9999 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{pmatrix} \tag{10}$$

It is clear that the four source signals are recovered successfully. However, if we remove Step 5 from Sequential Extraction Algorithm, and don't define new mixture with Step 3. the resulting first three signals are:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} == \begin{pmatrix} -0.0010 & 0.0065 & -0.9999 & -0.0004 \\ -0.9996 & 0.0051 & 0.0010 & 0.0172 \\ -0.8729 & 0.2291 & 0.0025 & -0.4391 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{pmatrix} \tag{11}$$

Then there is no need to extract the forth signal.

Example 2: In this simulation, s_i, $i = 1, 2, 3, 4$ are the same with those in previous simulation. And $s_5 = 7sin(n_5(t) - 0.8)$ ($n_5(t)$ is also a random vector in [0,1]). We whiten this signal first. And mixing matrix is

$$A = \begin{pmatrix} -0.6191 & -0.1506 & -0.3954 & -0.3167 & -0.5808 \\ -0.1406 & -0.4084 & 0.5564 & -0.6676 & 0.2410 \\ 0.1899 & -0.5034 & -0.6869 & -0.1351 & 0.4695 \\ -0.5058 & 0.5974 & -0.1478 & -0.1729 & 0.5792 \\ -0.5522 & -0.4475 & 0.2008 & 0.6370 & 0.2207 \end{pmatrix} \tag{12}$$

By Sequential Extraction Algorithm, We get

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{pmatrix} == \begin{pmatrix} -0.0011 & 0.0067 & -0.9997 & -0.0006 & -0.0216 \\ -0.9999 & 0.0043 & -0.0026 & 0.0129 & 0.0093 \\ -0.0078 & 0.0134 & -0.0062 & 0.9998 & -0.0104 \\ 0.0382 & -0.9987 & -0.0121 & 0.0154 & -0.0293 \\ -0.0006 & 0.0124 & 0.0157 & 0.0027 & -0.9998 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{pmatrix} \tag{13}$$

5 Conclusions

In this work, we propose a novel algorithm, which can be used to extract source signals sequently and avoid error accumulate. It also provides some technique to verify whether or not the resulting signals are real source signals.

Acknowledgements

This work is supported by National Science Foundation of China 69982003 and 60074005.

References

1. Ding, Z., and Nguyen, T. : Stationary Points of Kurtosis Maximization Algorithm for Blind Signal Separation and Antenna Beamforming. IEEE Trans. Signal Processing, **48** (2000) 1587-1596
2. Cardoso, J. : Source Separation Using Higher Order Moment. Proc. ICASSP (1983) 2109-2112

3. Tong, L., Soon, V., Huang, Y., and Liu, R. : AMUSE: A New Blind Identification Algorithm. Proc. ICASSP (1990) 1784-1787
4. Cao, X., and Liu, R. : General Approach to Blind Source Separation. IEEE Trans. Signal Processing **44** (1996) 562-571
5. Stoica, P., and Viberg, M. : Maximum Likelihood Parameter and Rank Estimation in Reduced-rank Multivariate Linear Regressions. IEEE Trans. Signal Processing **44** (1996) 3069-3078
6. Belouchrani, A., Abed-Meraim, K., Cardoso, J., and Moulines, E. : A Blind Source Separation Technique Using Second-order Statistics. IEEE Trans. Signal Processing, **45** (1997) 434-444
7. Shalvi, O., and Weinstein, E. : New Criteria for Blind Deconvolution of Nonminimum Phase Systems(channels). IEEE Trans. Inform. Theory, **36** (1990) 312-321
8. Ding, Z., and Nguyen, T. : Signal Separation and Extraction via Kurtosis Maximization. Proc. IEEE Workshop Digital Signal Process, Bryce Canyon, UT (1998)
9. Kung, S. : Independent Signal Analysis in Hybrid Mixture: KuicNet Learning Algorithm and Numerical Analysis. Proc. Int. Symp. Multimedia Inform. Process Taipei, Taiwan, China (1997) 368-381
10. Pham, D. : Blind Separation of Instantaneous Mixture of Sources Based on Order Statistics. IEEE Trans. Signal Processing, **48** (2000) 363-375
11. Amari, S., Chen, T., and Cichocki, A. : Stability Analysis of Learning Algorithms for Blind Source Separation. Neural Network, **10** (1997) 1345-1351
12. Amari, S., Chen, T., and Cichocki, A. : Nonholonomic Orthogonal Learning Algorithms for Blind Source Separation. Neural Computation, **12** (2000) 1463-1482
13. Boscolo, R., Pan, H., and Roychowdhurg, V. : Independent Signal Analysis Based on Nonparametric Density Estimation. IEEE Trans. Neural Networks, **15** (2004) 55-65
14. Chen, T., Amari, S., and Lin, Q., : A Unified Algorithm for Principal and Minor Signals Extraction. Neural Networks, **11** (1998), 385-390
15. Douglas, S., Amari, S., and Kung, S. : Gradient Adaptation under Unit-norm Constraints. Proc. IEEE Workshop Stat. Signal Array Processing, Portland, OR. (1998) 144-147
16. Amari, S., Cichocki, A., and Yang, H. : A New Learning Algorithm for Blind Signal Separation. Adv. Neural Inform. Proc. Sys. 8. MIT Press,Cambridge, MA (1996) 757-763
17. Kung, S., and Mejuto, C. : Extraction of Independent Signals from Hybrid Mixture: KuicNet Learning Algorithm and Applications. Proc. Int. Conf. Acoust., Speech, Signal Processing, Vol.2, Seattle, WA (1998) 1209-1212
18. Diamantaras, K., and Kung, S. : Principal Component Neural Networks: Theory and Applications. Wiley, New York (1996)
19. Douglas, S., and Cichocki, A. : Neural Networks for Blind Decorrelation of Signals. IEEE Trans. Signal Processing, **45** (1997) 2829-2842
20. Douglas, S., and Amari, S. : Natural Gradient Adaptation. Unsupervised Adaptive Filtering, Vol. I: Blind Source Separation, S. Haykin, Ed. Wiley, New York (2000) 13-61

A Learning Framework for Blind Source Separation Using Generalized Eigenvalues*

Hailin Liu[1] and Yiuming Cheung[2]

[1] Faculty of Applied Mathematics, Guangdong University of Technology, China
lhl@scnu.edu.cn
[2] Department of Computer Science
Hong Kong Baptist University, Hong Kong, China
ymc@comp.hkbu.edu.hk

Abstract. This paper presents a learning framework for blind source separation (BSS), in which the BSS is formulated as generalized Eigenvalue (GE) problem. Compared to the typical information-theoretical approaches, this new one has at least two merits: (1) the unknown unmixing matrix directly works out from the GE equation without time-consuming iterative learning; (2) The correctness of the solution is guaranteed. We give out a general learning procedure under this framework. The computer simulation shows validity of our method.

1 Introduction

At present, many authors engage in blind source separation (BSS) or independent component analysis (ICA) research work, and a lot of studying literature (for a review, see [1]) have been published. The most basic form of BSS can be stated as follows: Suppose there are n channels of source signals with at most one Gaussian source signal, denoted as $s_1(t), s_2(t), ..., s_n(t)$, which are statistically independent each other. The sources are instantaneously and linearly mixed by an unknown full-rank square matrix $\mathbf{A}$ and observed as:

$$\mathbf{x(t)} = \mathbf{As(t)}, \tag{1}$$

where $\mathbf{s(t)} = [s_1(t), s_2(t), ..., s_n(t)]^T, \mathbf{x(t)} = [x_1(t), x_2(t), ..., x_n(t)]^T$, and T is a transpose operation of a matrix. The objective of an ICA approach is to recover $s(t)_i^\prime s$ up to a constant scale and any permutation of indices through a set of observations $\mathbf{x(t)}$ by finding out a de-mixing matrix $\mathbf{W}$ such that

$$\mathbf{y(t)} = \mathbf{Wx(t)}, \tag{2}$$

where $\mathbf{y(t)} = [y_1(t), y_2(t), ..., y_n(t)]^T$ is a recovered signal of $\mathbf{s(t)}$.

In the literature, one approach initiated from the seminal work of Stone [6] is to formulate the BSS as generalized Eigenvalue (GE) problem. Compared to the

* This work was supported by the Research Grant Council of Hong Kong SAR under Project HKBU 2156/04E.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 472–477, 2005.

typical information-theoretic based approaches, e.g., INFOMAX [2], negentropy [3], cumulant [4], and ML [5], the GE-based ones have at least two merits: (1) the unknown un-mixing matrix directly works out from the GE equation without time-consuming iterative learning; (2) The correctness of the solution can be guaranteed.

In Stone's paper [6], a metric named *Temporal Predictability* has been presented as a logarithm of a ratio of two prediction error terms. The numerator is the summation of long-term prediction errors of a $\boldsymbol{y}$'s component, while the numerator is the summation of its short-term prediction errors. Essentially, his work is based on the conjecture that, *given any set of statistically independent source signals, the temporal predictability of any signal mixture is less than (or equal to) that of any of its component source signals*. Unfortunately, although a number of experiments have reported its success, some empirical studies have found that this conjecture is not totally correct, as pointed out in our recent paper [7]. Under the circumstances, we have proposed a new metric called Independency Metric, through which a new BSS algorithm with global convergence is presented [7].

In this paper, we further present a general BSS learning framework formulated as GE problems. We have given out a generalized contrast function, whereby a general BSS learning procedure is obtained with those algorithms in [7, 8] as its particular examples. We have analyzed the global convergence property of such a learning, and shown that it guarantees to acquire a correct BSS solution.

2 Contrast Function Extracting Source Signals

In instantaneous linear mixture model Eq.(1), suppose there exist two operators g and h, for any real number k_i and k_j, satisfying the following relationship:

$$g(k_is_i(t)+k_js_j(t)) = k_i^2g(s_i(t)) + k_j^2g(s_j(t)), \forall i,j \in \{1,2,...,n\}, i \neq j. \quad (3)$$

$$h(k_is_i(t)+k_js_j(t)) = k_i^2h(s_i(t)) + k_j^2h(s_j(t)), \forall i,j \in \{1,2,...,n\}, i \neq j. \quad (4)$$

$$g(k_is_i(t), k_js_j(t)) = 0, h(k_is_i(t), k_js_j(t)) = 0, \forall i,j \in \{1,2,...,n\}, i \neq j. \quad (5)$$

Hence, we have

$$g(\mathbf{x}(t)) = \mathbf{A} \cdot diag([g(s_1(t)), g(s_2(t)), ..., g(s_n(t))])\mathbf{A}^T. \quad (6)$$

For the recovered signal $\mathbf{y}(t)$ given by Eq.(2), we can then obtain:

$$g(\mathbf{y}(t)) = \mathbf{WA} \cdot diag([g(s_1(t)), g(s_2(t)), ..., g(s_n(t))])(\mathbf{WA})^T. \quad (7)$$

We know that each component y of $\mathbf{y}$ is a linear mixture of n sources with:

$$y = \mathbf{wx}, \quad (8)$$

where $\mathbf{w}$ is a n-dimensional row vector. We then define a general form of contrast function of BSS making use of generalized eigenvalue as

$$L(\mathbf{wx}) = \frac{g(\mathbf{wx})}{h(\mathbf{wx})} = \frac{\mathbf{w}g(\mathbf{x})\mathbf{w}^T}{\mathbf{w}h(\mathbf{x})\mathbf{w}^T}, \tag{9}$$

For the new redundancy reduction metric, similar to the proof in [7], we can get the following theorem:

Theorem 1. For the source signals $s_1, s_2, ..., s_n$, suppose there exist two operators g and h satisfying Eq.(4) and Eq.(5) so that $L(s_i)$'s are not equal each other, i.e.,

$$\frac{g(s_i)}{h(s_i)} \neq \frac{g(s_j)}{h(s_j)}, i \neq j.$$

Denote

$$L(s_{i_0}) = max\{L(s_1), L(s_2), ..., L(s_n)\}. \tag{10}$$

For any mixing signal y described in Eq.(8), we then have

$$L(y) \leq L(s_{i_0}). \tag{11}$$

If and only if $y = ks_{i_0}$, where k is any non-zero real number, then

$$L(y) = L(s_{i_0}). \tag{12}$$

In the above theorem, operators h and g may have various forms. For instance, $h(s(t)) = var(s(t)), g(s(t)) = var(\int_0^t s(\tau)d\tau)$ (used in [7]); $h(s(t)) = E(s^2(t))$, $g(s(t)) = \sum_{i=0}^{t} E(s^2(i))$ (used in [8]).

3 Globally Optimal Analysis of BSS Algorithm

3.1 Equivalent Form About Gradient of Contrast Function

Since $L(s_j)$'s are not equal each other, without loss of generality, we assume that

$$L(s_1) > L(s_2) > ... > L(s_n). \tag{13}$$

According to Theorem 1 and Eq.(9), we therefore have

$$Q(\mathbf{w}) = logL(\mathbf{wx}) \leq logL(s_1). \tag{14}$$

This implies that the source signal s_1 can be extracted through solving following optimization problem:

$$\max_{\mathbf{w}\neq 0} Q(\mathbf{w}). \tag{15}$$

The objective function in Eq.(15) can transform

$$Q(\mathbf{w}) = logL(\mathbf{wx}) = log\frac{\mathbf{w}g(\mathbf{x})\mathbf{w}^T}{\mathbf{w}h(\mathbf{x})\mathbf{w}^T}. \tag{16}$$

We let $\|\mathbf{z}\|_2$ be a norm of vector $\mathbf{z}=(z_1, z_2, ..., z_n)$, and $\|\mathbf{z}\|_2=\sqrt{z_1^2 + z_2^2 + ... + z_n^2}$. In Eq.(16),

$$Q(\mathbf{w}) = \log((\frac{\mathbf{w}}{\|\mathbf{w}\|_2})g(\mathbf{x})(\frac{\mathbf{w}}{\|\mathbf{w}\|_2})^T)/((\frac{\mathbf{w}}{\|\mathbf{w}\|_2})h(\mathbf{x})(\frac{\mathbf{w}}{\|\mathbf{w}\|_2})^T), \tag{17}$$

Therefore, from Eq.(16) and (17), we obtain

$$\max_{\mathbf{w}\neq 0} Q(\mathbf{w}) = \max_{\|\mathbf{w}\|_2=1} Q(\mathbf{w}) \tag{18}$$

Since $h(\mathbf{x}) = \mathbf{A}h(\mathbf{s})\mathbf{A}^T, g(\mathbf{x}) = \mathbf{A}g(\mathbf{s})\mathbf{A}^T$, $\mathbf{A}$ is a nonsingular square matrix, and both h($\mathbf{s}$) and g($\mathbf{s}$) are two diagonal matrices, we know that h($\mathbf{x}$) and g($\mathbf{x}$) are real symmetrical and positive definite matrices. Therefore, Q($\mathbf{w}$) is a logarithm of ratio of two positive definite quadratic forms, and it is continuous and differentiable. Since the set $\{\mathbf{w}|\|\mathbf{w}\|_2 = 1\}$ is a closed set, according to Eq.(18), we know that the objective function Q($\mathbf{w}$) in (15) exists global maximum and global minimum. Since Q($\mathbf{w}$) is differentiable, all optimal solutions in (15) must be stable point.

With some mathematical computations, we can finally obtain the gradient of Q($\mathbf{w}$):

$$\begin{aligned}\bigtriangledown Q(\mathbf{w}) &= \frac{2g(\mathbf{x})\mathbf{w}^T}{\mathbf{w}g(\mathbf{x})\mathbf{w}^T} - \frac{2h(\mathbf{x})\mathbf{w}^T}{\mathbf{w}h(\mathbf{x})\mathbf{w}^T} = \frac{2}{\mathbf{w}g(\mathbf{x})\mathbf{w}^T}\{g(\mathbf{x})\mathbf{w} - \frac{\mathbf{w}g(x)\mathbf{w}^T}{\mathbf{w}h(\mathbf{x})\mathbf{w}^T}h(\mathbf{x})\mathbf{w}^T\} \\ &= \frac{2}{\mathbf{w}g(\mathbf{x})\mathbf{w}^T}\{g(\mathbf{x})\mathbf{w}^T - L(\mathbf{wx})h(\mathbf{x})\mathbf{w}^T\}.\end{aligned} \tag{19}$$

If and only if $\bigtriangledown Q(\mathbf{w}) = 0$, we obtain

$$g(\mathbf{x})\mathbf{w}^T = L(\mathbf{wx})h(\mathbf{x})\mathbf{w}^T. \tag{20}$$

3.2 Generalized Eigenvalue Problem

Definition 1. Suppose that $\mathbf{A}$ is an $n \times n$ real symmetrical matrix and $\mathbf{B}$ is an $n \times n$ real symmetrical and positive definite matrix, the following eigenvalue problem:

$$\mathbf{Ax} = \lambda\mathbf{Bx} \tag{21}$$

is called generalized eigenvalue problem; the number λ satisfying Eq.(20) is called eigenvalue of matrix A relative to matrix B; the nonzero solution relative to λ is called eigenvector belonging to λ.

Note that both $g(\mathbf{x})$ and $h(\mathbf{x})$ are real symmetrical and positive definite matrices. Therefore, Eq.(20) is a generalized eigenvalue problem, where $L(\mathbf{wx})$ is an eigenvalue about the problem and $\mathbf{w}^T$ is an eigenvector corresponding to eigenvalue $L(\mathbf{wx})$. Furthermore, solving Eq.(15) actually becomes a problem solving generalized eigenvalue vector. Suppose $\mathbf{w}_0$ is a stable point of Q($\mathbf{w}$) in (15) and is also an eigenvector corresponding to eigenvalue $L(\mathbf{w}_0^T\mathbf{x})$ in Eq.(20). From

Eq.(16), we then have $Q(\mathbf{w}_0) = logL(\mathbf{w}_0\mathbf{x})$. If $L(\mathbf{w}_0\mathbf{x})$ is the largest eigenvalue in Eq.(20), $\mathbf{w}_0$ must be the global optimal solution in Eq.(15). Thus, through solving an eigenvector corresponding to the largest eigenvalue in Eq.(20), we can recover the source signals.

3.3 Method of Recovering All Source Signals

Theorem 2. Under the condition of Theorem 1, $\hat{\mathbf{w}}x$ is a recovered signal of source signals if and only if it is a stable point of optimization problem Eq.(15).

Proof: Sufficiency of the condition. Suppose that $\hat{\mathbf{w}}$ is a stable point of optimization problem Eq.(15). From Eq.(20), we obtain

$$g(\mathbf{x})\hat{\mathbf{w}}^T = L(\hat{\mathbf{w}}\mathbf{x})h(\mathbf{x})\hat{\mathbf{w}}^T. \tag{22}$$

Let $\mathbf{w}_i = (\overbrace{0, \ldots, 0, 1}^{i}, 0, \ldots, 0)\mathbf{A}^{-1}$(i=1,2,. . . ,n), then all eigenvalues are $L(\mathbf{w}_1\mathbf{x})$, $L(\mathbf{w}_2\mathbf{x}), ..., L(\mathbf{w}_n\mathbf{x})$, in generalized eigenvalue problem Eq.(20). assume $L(\hat{\mathbf{w}}x) = L(\mathbf{w}_{i_1}\mathbf{x})$ where $1 \le i_1 \le n$, and $\hat{\mathbf{w}}$ is an eigenvector corresponding to eigenvalue $L(\mathbf{w}_{i_1}\mathbf{x})$. Because the linear uncorrelated eigenvector corresponding to eigenvalue is exclusive in Eq.(20) and $\mathbf{w}_{i_1}$ is a eigenvector corresponding to eigenvalue $L(\mathbf{w}_{i_1}\mathbf{x})$, we have $\hat{\mathbf{w}} = c_1\mathbf{w}_{i_1}$, in which c_1 is a non-zero constant. Hence, we obtain $\hat{\mathbf{w}\mathbf{x}} = (c_1\mathbf{w}_{i_1})\mathbf{x} = c_1 s_{i_1}$.

Necessity of the condition. If $\hat{\mathbf{w}}\mathbf{x}$ is a recovered signal of source signals, there exists a source signal s_{i_2} $(1 \le i_2 \le n)$ such that $\hat{\mathbf{w}}\mathbf{x} = c_2 s_{i_2}$, in which c_2 is non-zero constant. Similar to the proof of Eq.(36), from Eq.(19), we have

$$g(\mathbf{x})\hat{\mathbf{w}} = L(\hat{\mathbf{w}}\mathbf{x})h(\mathbf{x})\hat{\mathbf{w}} \tag{23}$$

According to Eq.(19) and Eq.(20), $\hat{\mathbf{w}}$ is a stable point of optimization problem Eq.(15). □

Corollary 1. $\hat{\mathbf{w}}_i x$ is a recovered signal of source signal s_i if and only if $\hat{\mathbf{w}}_i$ is an eigenvector corresponding to the eigenvalue in Eq.(20).

Proof: On the basis of above proof and Eq.(19), it can be seen that this result is true. □

According to Theorem 2, de-mixing matrix W can be obtained through solving eigenvector corresponding to eigenvalue $L(s_1), L(s_2), ..., L(s_n)$ in Eq.(20).

4 Simulation Results

In our computer simulation, we let h(s) be signal s and the operator g be,

$$g(\mathbf{s}(t)) = \begin{cases} var(\sum_{i=1}^{t} \mathbf{s}(i)), t \le q; \\ var(\sum_{i=t-q+1}^{t} \mathbf{s}(i)), t > q. \end{cases} \tag{24}$$

We used three independent source signals: a subgaussian signal s_1(a cosine signal), a supergaussian signal s_2(a speech sound), and a gaussian signal s_3 generated using the randn procedure in Matlab. In this example, mixing matrix $\mathbf{A}$

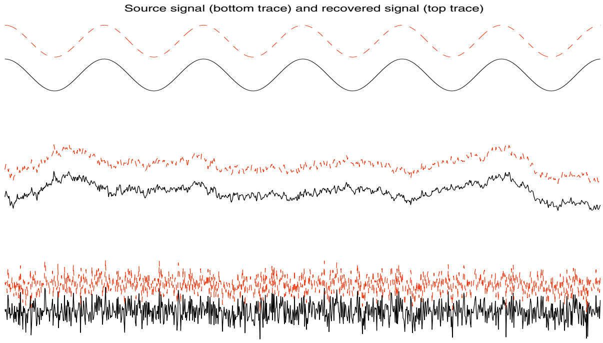


Fig. 1. Separating Mixture signals of Super-Guassian Signal, Sub-Gaussian Signal and Guassian Signal

was generated at random and the number of samples was 5,000. Each source signal (solid line) and its corresponding recovered signal (dot line) acquired by our proposed algorithm are shown in Figure 1.

5 Conclusion

In this paper, we have presented a general BSS learning procedure using general Eigenvalues. Such a learning not only acquires the BSS solution in the one step without the time-consuming iterative learning as used in those information-theoretic based algorithms, but also makes a correct BSS solution guaranteed. The computer simulations have shown the success of our method.

References

1. Hyvarinem, A., Oja, E.: Independent Component Analysis: Algorithms and Applications. Neural Networks, **13**(2000) 411–430
2. Bell, A.J., Sejnowski, T.J.: An Information Approach to Blind Separation and Blind Deconvolution. Neural Computation, **7** (1995) 1129-1159
3. Girolami, M., Fyfe, C. : Extraction of Independent Signal Sources Using a Deflationary Exploratory Projection Pursuit Network with Lateral Inhibition. IEE Proceedings on Vision, Image and Signal Processing Journal, **14** (1997) 299-306
4. Hyvarinem, A., Oja, E.: Independent Component Analysis: Algorithms and Applications. Neural Networks, **13** (2000) 411-430
5. Pham, D.T. , Garrat, P., Jutten, C.: Separation of A Mixture of Independent Sources Through a Maximum Likelihood Approach. Proceedings of EUSIPCO, (1992) 771-774
6. Stone, J.V., : Blind Source Separation Using Temporal Predictability. Neural Computation, **13** (2001) 1559-1574
7. Cheung, Y.M., Liu, H.L.,: A New Approach to Blind Source Separation with Global Optimal Property. Proceedings of the IASTED International Conference of Neural Networks and Computational Intelligence. Grindelwald, Switzerland (2004) 137–141
8. Parra, L., Sajda, P., : Blind Source Separation via Generalized Eigenvalue Decomposition. Journal of Machine Learning Research, **4** (2003) 1261-1269

Post-nonlinear Blind Source Separation Using Neural Networks with Sandwiched Structure

Chunhou Zheng[1,2], Deshuang Huang[1], Zhanli Sun[1,2], and Li Shang[1,2]

[1] Intelligent Computing Lab, Institute of Intelligent Machines, Chinese Academy of Sciences
P.O. Box 1130, Hefei, Anhui 230031, China
[2] Department of Automation, University of Science and Technology of China
{zhengch,dshuang}@iim.ac.cn

Abstract. This paper proposes a novel algorithm based on informax for post-nonlinear blind source separation. The demixing system culminates to a neural network with sandwiched structure. The corresponding parameter learning algorithm for the proposed network is presented through maximizing the joint output entropy of the networks, which is equivalent to minimizing the mutual information between the output signals in this algorithm, whereas need not to know the marginal probabilistic density function (PDF) of the outputs as in minimizing the mutual information. The experimental results about separating post-nonlinear mixture stimulant signals and real speech signals show that our proposed method is efficient and effective.

1 Introduction

The separation of independent sources from mixed observed data is an challenging signal processing problem, especially when the mixing is nonlinear. Most of the blind separation algorithms are based on the theory of the independent component analysis (ICA) when the mixture model is linear [1],[2]. However, in general real-world situation, nonlinear mixture of signals is generally more prevalent. For nonlinear demixing, many difficulties occur and the linear ICA is no longer applicable because of the complexity of nonlinear parameters.

Even so, several researchers have studied the difficult problems of the nonlinear blind source separation and derived a few efficient demixing algorithms [2],[3],[6],[7]. Despite these attempts to solve the blind source separation problem in nonlinear mixtures, there still exist the problems of lacking solid theoretical background. In fact, the problem of nonlinear BSS is possibly a collection of problems that should be addressed on a case-by-case basis [5].

In this paper, we shall thoroughly investigate a special but important instance of nonlinear mixtures, i.e., post-nonlinear mixtures. We employ a network composed of two sets of multilayer perceptrons and a linear network to separate sources in post-nonlinear mixtures. Furthermore, this new method can adaptively estimate the nonlinearities that are needed at the outputs of the network used in informax, which allows the method to deal with the components with a wider range of distributions.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 478–483, 2005.

2 Post-nonlinear Mixtures

An important special case of the general nonlinear mixing model consisting of so called post-nonlinear mixtures introduced by Taleb and Jutten [6], can be seen as a hybrid of a linear stage followed by a nonlinear stage. In the post-nonlinear mixtures model, the observations $\mathbf{x} = (x_1, x_2, \cdots, x_n)^T$ have the following specific form:

$$x_i = f_i\left(\sum_{j=1}^{n} a_{ij} s_j\right) i = 1, \cdots, n \,. \tag{1}$$

Here, the nonlinear functions f_i are assumed to be invertible. Even though the PNL model partitions complex nonlinear mixtures into a linear mixing and a nonlinear transformation, and ignores the nonlinearity of cross-channel mixing, the model is actually general enough.

The most important thing when dealing with nonlinear mixtures is the separability issue. In literature [6], it has been shown that the post-nonlinear mixtures are separable for the distributions having at most one Gaussian source, with the same indeterminacies as the linear mixtures if **A** has at least two nonzero entries on each row or column.

3 Unsupervised Learning of Separating System

3.1 Contrast Function

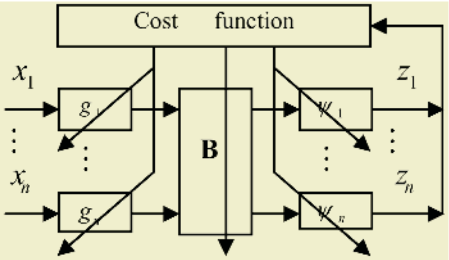


Fig. 1. Structure of ICA system proposed in this paper

The separating system of the PNL proposed in this paper is shown in Fig.1, where blocks ψ_i are auxiliary module, being used only during the optimization phase, the cost function is the output entropy $H(\mathbf{z})$. Clearly, in this system, a linear demixing matrix is sandwiched between two nonlinear networks.

Assume that each function ψ_i is the cumulative probability function (CPF) of the corresponding component y_i, then, a simple calculation will show that z_i will be uniformly distributed in [0, 1]. Consequently, $H(z_i) = 0$. Therefore, we have:

$$I(\mathbf{y}) = I(\mathbf{z}) = \sum_i H(z_i) - H(\mathbf{z}) = -H(\mathbf{z}) \tag{2}$$

It has been proved in literature [7] that given the constraints placed on the functions ψ_i, z_i is bounded to [0, 1], and its maximal entropy will correspond to a uniform distribution in that interval. Besides, given that ψ_i is also constrained to be an increasing function, it must be the CPF of y_i, yielding a z_i uniformly distributed in [0, 1]. Therefore, maximizing the output entropy will lead the functions ψ_i to become the estimate of the CPFs of the corresponding components y_i. The basic problem that we have to solve is to optimize the networks (formed by the blocks of g_i, **B** and ψ_i) by maximizing the output entropy $H(\mathbf{z})$.

3.2 Algorithm Architecture

With respect to the separation structure of this paper, the joint probabilistic density function (PDF) of the output vector $\mathbf{z}$ can be calculated as:

$$p(\mathbf{z}) = \frac{p(\mathbf{x})}{\left|\det(\mathbf{B})\right| \prod_{i=1}^{n} \left|g'_i(\boldsymbol{\theta}_i, x_i)\right| \prod_{i=1}^{n} \left|\psi'_i(\boldsymbol{\varphi}_i, y_i)\right|} \tag{3}$$

which leads to the following expression of the joint entropy:

$$H(\mathbf{z}) = H(\mathbf{x}) + \log\left|\det(\mathbf{B})\right| + \sum_{i=1}^{n} E\left(\log\left|g'_i(\boldsymbol{\theta}_i, x_i)\right|\right) + \sum_{i=1}^{n} E\left(\log\left|\psi'_i(\boldsymbol{\varphi}_i, y_i)\right|\right) \tag{4}$$

The minimization of $I(\mathbf{y})$, which is equal to maximizing $H(\mathbf{z})$ here, requires the computation of its gradient with respect to the separation structure parameters $\mathbf{B}$, $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$.

Since $H(\mathbf{x})$ does not contain any parameters of the separating system, it becomes null when taking a gradient with respect to the parameters. We thus have the following gradient expressions:

$$\frac{\partial H(\mathbf{z})}{\partial \mathbf{B}} = \frac{\partial \log\left|\det(\mathbf{B})\right|}{\partial \mathbf{B}} + \frac{\partial\left(\sum_{i=1}^{n} E\left(\log\left|\psi'_i(\boldsymbol{\varphi}_i, y_i)\right|\right)\right)}{\partial \mathbf{B}} \tag{5}$$

$$\frac{\partial H(\mathbf{z})}{\partial \boldsymbol{\theta}_k} = E\left(\frac{\partial \log\left|g'_k(\boldsymbol{\theta}_k, x_k)\right|}{\partial \boldsymbol{\theta}_k}\right) + \frac{\partial\left(\sum_{i=1}^{n} E\left(\log\left|\psi'_i(\boldsymbol{\varphi}_i, y_i)\right|\right)\right)}{\partial \boldsymbol{\theta}_k} \tag{6}$$

$$\frac{\partial H(\mathbf{z})}{\partial \boldsymbol{\varphi}_k} = E\left(\frac{\partial \log \left| \psi'_k(\boldsymbol{\varphi}_k, y_k) \right|}{\partial \boldsymbol{\varphi}_k} \right) \tag{7}$$

Of course, their computation depends on the structure of the parametric nonlinear mapping $\mathbf{g}$ and ψ. In this paper, we use multilayer perceptron (MLP) to model the nonlinear parametric functions $g_k(\boldsymbol{\theta}_k, x_k)$ and $\psi_k(\boldsymbol{\varphi}_k, y_k)$.

4 Experimental Results

This section will present two simulating experiments of post-nonlinear blind separation to illustrate the effectiveness and efficiency of our proposed algorithm.

In the first experiment, the source signals consist of a sinusoid signal and a funny curve signal [2], i.e., $\mathbf{s}(t) = [\sin(t/3), ((\text{rem(t,23)-11})/9)^5]^T$, which are shown in Fig.2 (a). The two source signals are first linearly mixed with the (randomly chosen) mixture matrix:

$$\mathbf{A} = \begin{bmatrix} 0.6234 & 0.3103 \\ 0.3125 & -0.7961 \end{bmatrix}$$

Then, the two nonlinear distortion functions

$$f_1(u) = \frac{1}{2}(u + \frac{1}{5}u^3) \tag{8}$$

$$f_2(u) = \frac{3}{5}u + \tanh(u) \tag{9}$$

are applied to each mixture for producing a PNL mixture, thus

$$[x_1(t), x_2(t)]^T = [f_1(u_1), f_2(u_2)]^T \tag{10}$$

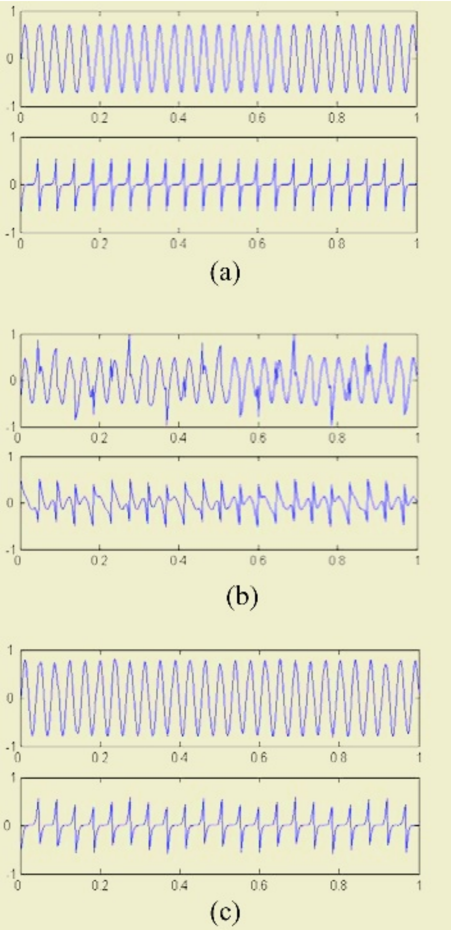
The mixture signals $\mathbf{x}(t)$ are shown in Fig.2 (b). And, Fig.2 (c) shows the separated signals.

To compare the performance of our proposed method with other ones, we also use linear ICA algorithm (FastICA [4]) and MISEP method [7] to conduct the related experiments based on the same data. The correlations between the two recovered signals separated by three methods and the two original sources are reported in Table.1. Clearly, the separated signals using the method proposed in this paper is more similar to the original signals than the others.

To test the validity of the algorithm proposed in this paper ulteriorly, we also have carried out the related experiments using real-life speech signals. In this experiment two speech signals (with 20000 samples, sampling rate 8kHz, obtained from http://www.ece.mcmaster.ca/~reilly/kamran/id18.htm) are post-nonlinearly mixed by:

$$\text{A} = \begin{bmatrix} 0.6234 & 0.3103 \\ 0.3125 & -0.7961 \end{bmatrix}$$

$$f_1(u) = f_2(u) = \tanh(u) \tag{11}$$



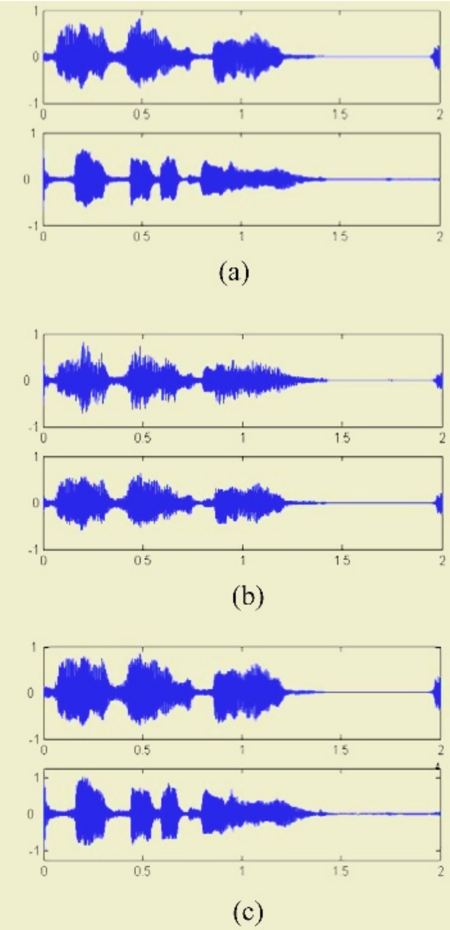


Fig. 2. The three set of signals shown. (a) Source signals. (b) Mixed signals. (c) Separated signals

Fig. 3. The three set of speech signals shown. (a) Source signals. (b) Mixed signals.(c) Separated signals

The experimental results are shown in Fig.3 and Table.1, which conform to the conclusion drawn from the first experiment.

5 Conclusions

We proposed in this paper a novel algorithm for post-nonlinear blind source separation, which is based on informax criterion differing from the one in literature [8]. This new method works by optimizing a network with a specialized architecture, using the output entropy as the objective function, which is equivalent to the mutual information criterion but needs not to calculate the marginal entropy of the output. Finally, the experimental results showed that this method is competitive to other existing ones like FastICA and MISEP.

Table 1. The correlations between two original sources and the two recovered signals achieved by three different methods

Experiment	NO.1		NO.2	
	c_1	c_2	c_1	c_2
FastICA	0.9327	0.9416	0.9450	0.9300
MISEP	0.9864	0.9515	0.9878	0.9460
Method in this paper	0.9974	0.9838	0.9978	0.9980

*c_i : Correlation between the ith recovered signal and the ith original source

The weakness of the proposed method, however, lies in the fact that it can only be used to resolve the PNL problem. Future works will focus on adjusting the structure of the network to resolve more general nonlinear BSS problem.

References

1. Comon, P.: Independent Component Analysis, a New Concept?. Signal Processing, **36** (1994) 287– 314
2. Hyvärinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. J. Wiley, **3** (2001)
3. Hyvärinen, A., Pajunen, P.: Nonlinear Independent Component Analysis: Existence and Uniqueness Results. Neural Networks, **12** (1999) 429–439
4. Hyvärinen, A.: Fast and Robust Fixed-point Algorithms for Independent Component Analysis. IEEE Trans. Neural Networks, **10** (1999) 626–634
5. Jutten, C., Zadeh, M.B., Hosseini, S.: Three Easy Ways for Separating Nonlinear Mixtures?. Signal Processing, **84** (2004) 217–229
6. Taleb,A., Jutten, C.: Source Separation in Post- nonlinear Mixtures. IEEE Trans. Signal Processing, **47** (1999) 2807–2820
7. Almeuda, L.B.: Linear and Nonlinear ICA Based on Mutual Information –the MISEP Method. Signal Processing, **84** (2004) 231-245
8. Bell, A., Sejnowski, T.: An Information-maximization Approach to Blind Separation and Blind Deconvolution. Neural Computation, **7** (1995) 1129–1159

A Novel Approach for Underdetermined Blind Sources Separation in Frequency Domain*

Ming Xiao[1,2], Shengli Xie[1], and Yuli Fu[1]

[1] School of Electrics & Information Engineering, South China University of Technology
Guangzhou, Guangdong 510640, China
[2] Department of Electrics & Information Engineering, Maoming College
Maoming, Guangdong 525000 , China

Abstract. In this paper, we discussed the separation of n sources from m linear mixtures when the underlying system is underdetermined, that is, when $m<n$. The underdetermined blind sources separation has two steps. In matrix-recovery step, we defined a characteristic of the signals as the durative-sparsity and proposed a novel approach called as a searching-and-averaging-based method in frequency domain. This approach tells us how to search some data points that are very close to the basis lines along the direction of basis vectors a^j and how to use them to estimate the mixing matrix. In source-recovery step, we used Bofill and Zibulevsky's shortest-path algorithm. Finally, the separation results were obtained using their short-time Fourier transforms.

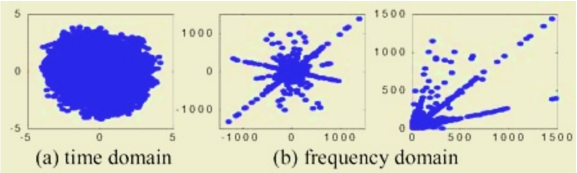
1 Introduction

Recent blind source separation based on independent component analysis (ICA) has many potential applications including speech recognition systems, telecommunications, and medical signal processing. The standard formulation of ICA requires that sources are more than sensors [1],[2].

The column a^j ($j = 1,2,\cdots,n$) of the mixing matrix A is a basis vector. The line along its direction is called as a basis line. When the sensors are fewer than the sources, the underlying system is underdetermined. If the sources are sparse, the underdetermined blind source separation (BSS) problem can be solved by a two-step approach, which has been proposed recently by Bofill and Zibulevsky [4]. The first step is to estimate mixing matrix, and the second is to estimate sources. In the matrix-recovery step, usually use k-means, a potential-function-based method [4] and etc. The goal of these approaches is to estimate the basis lines either by external optimization or by clustering. In source-recovery step, often use a shortest-path algorithm [4],[5] and a method of solving a low-dimensional linear programming problem for each of data points [3].

[*] The work is supported by the Guang Dong Province Science Foundation for Program of Research Team (grant 04205783), the National Natural Science Foundation of China (Grant 60274006), the Natural Science Key Fund of Guang Dong Province, China (Grant 020826), the National Natural Science Foundation of China for Excellent Youth (Grant 60325310) and the Trans-Century Training Program, the Foundation for the Talents by the State Education Commission of China, and the SRF for ROCS, SEM.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 484–489, 2005.



(a) time domain (b) frequency domain

Fig. 1. Scatter plot x_2 vs. x_1 of six flutemelodies mixed into two mixtures with equally spaced angles in the (a) time, (b) frequency domains

Several experiments with music signals show that their time-domain representation is not sparse enough, and their frequency-domain representation is sparse enough (see Fig.1). In Fig.1 (b), after DFT of the mixtures, the left is the scatter plot of the real and image, and the right is the scatter plot of the module. So the mixing matrix can be estimated in the sparser frequency domain.

Matrix-recovery either by external optimization or clustering needs many samples. Because most data points in mixture space would not lie on the basis lines (see Fig. 1(b)), the difference between the angles of the estimated and actual basis vectors would likely be very large. In order to estimate the basis vectors accurately and reduce the computational complexity, we propose a searching-and-averaging-based method in the frequency domain.

2 Matrix-Rrecovery Step

Let x^t be an m-dimensional column vector corresponding to output of m sensors at a given discrete time instant t. Let s^t be an n-dimensional column vector of underlying source signals at discrete time instant t and let A be the $m \times n$ mixing matrix. When $m < n$, the underlying system of equation: $x^t = As^t$ is underdetermined, where A and s^t are unknown. Very often the data in the time domain don't satisfy the requirement of sparsity. However, if x^t is discrete Fourier transformed, the data in frequency domain have higher sparsity. So x^t and s^t are respectively DFT-transformed into X^k and S^k . The DFT being linear, the mixing matrix is preserved and the underlying system of equation can be rewritten as $X^k = A \cdot S^k$, where $X^k = (X_1^k, X_2^k, \cdots, X_m^k)$. The blind source separation, then, is performed in frequency domain. For our purpose, a useful formulation of this system is obtained by decomposing A into its columns a^j :

$$X^k = \sum_{j=1}^{n} a^j S_j^k \quad for \quad k = 0,1,,\cdots,N-1 . \tag{1}$$

Following from Eq. (1), if only one of sources (say, source j) was different from zero, then all X^k 's would be proportional to a^j and all data points in mixture space would be aligned along the direction of this basis vector. When the sources are sparse, smaller coefficients are more likely and thus, for a given data point k , if one of the sources is significantly larger, the remaining ones are likely to be close to zero. For the given data point k , (1) is written as $X^k \approx a^j S_j^k$.

Now we only consider the module of complex X^k and S^k in the sparse case, (1) is also written as

$$|X^k| \approx |a^j| \cdot |S_j^k|. \tag{2}$$

where | • | denotes the module of a complex. Let $X_{\mathrm{mod}} = (|X^0|,|X^0|,\cdots,|X^{N-1}|) \in R^{m\times N}$ be the data matrix of the module of the complex X^k in frequency domain.

In this paper, only use the data matrix X_{mod} to estimate mixing matrix A.

When $m=2$, we project $|X^k|$ onto unit circle and get $u^k := |X^k| / \|(|X^k|)\|$, where $\|\cdot\|$ denotes the module of a vector. We can write $U=(u^0,u^1,\cdots,u^{N-1})$. Then set $d^k = u^k - u^{k+1}$ and get $D=(d^0,d^1,\cdots,d^{N-1})$. If d^k equals zero, the vector $|X^k|$ and $|X^{k+1}|$ will lie on the same line.

Now define the durative-sparsity of module in frequency domain.

Definition 1. If the modules $|S_j^k|$ of only one of the source signals, for more than three consecutive data points, are nonzero or far bigger than the others, get Eq. (2). The characteristic is called as durative-sparsity of module in frequency domain. The frequency interval Δk for having durative-sparsity is called as single-signal frequency interval (SSFI). The others are called as multi-signal frequency interval (MSFI).

Real and image of the DFT coefficients of the source signals usually don't have the characteristic.

Definition 2. Give positive whole number Δ, if $d^k = 0$ when $k = k_0, k_0+1,\cdots,k_1$ and $k_1 - k_0 + 1 \ge \Delta$, and $d^k \ne 0$ when $k = k_0 - 1, k_1 + 1$, the frequency interval from k_0 to k_1+1 is called as *single-signal frequency interval* (SSFI). The others are called as *multi-signal frequency interval* (MSFI). Let Δ be the minimum length of SSFI.

In fact, very often the vector d^k is not equal but approximate to zero.

Definition 3. Give error $\varepsilon > 0$ and positive whole number Δ, if $|d_1^k| + |d_2^k| < \varepsilon$ when $k = k_0, k_0+1,\cdots,k_1$ and $k_1 - k_0 + 1 \ge \Delta$, and $|d_1^k| + |d_2^k| \ge \varepsilon$ when $k = k_0 - 1, k_1 + 1$, the frequency interval from k_0 to k_1+1 is called as SSFI. The other intervals are called as MSFI.

Set $\Omega = (\omega^0, \omega^1, \cdots, \omega^{N-1})$, where $\omega^k = (1,1)^T$ if k lies in SSFI, and $\omega^k = (0,0)^T$ if k lies in MSFI. Let $X^* = X_{\mathrm{mod}} \odot \Omega$ be the data matrix for SSFI, where $X^* = (X^{*0}, X^{*1}, \cdots, X^{*N-1})$, $X^{*k} = (X_1^{*k}, X_2^{*k})^T$, and $\odot$ denotes array multiply.

Because all X^{*k} lie in first quadrant of Cartesian coordinates (see the right of Fig.1 (b)), the column of mixing matrix estimated by X^* is $|a^j|$. In order to estimate a^j, we must correct the matrix X^* as following:

$$\begin{cases} \bar{X}_1^k = X_1^{*k},\ \bar{X}_2^k = X_2^{*k} & when\ \ 0 \le |\varphi_1(k) - \varphi_2(k)| \le \dfrac{\pi}{2}\ or\ \dfrac{3\pi}{2} \le |\varphi_1(k) - \varphi_2(k)| \le 2\pi \\ \bar{X}_1^k = -X_1^{*k},\ \bar{X}_2^k = X_2^{*k} & when\ \ \dfrac{\pi}{2} \le |\varphi_1(k) - \varphi_2(k)| \le \dfrac{3\pi}{2} \end{cases} . \tag{3}$$

Where $\varphi_1(k)$ and $\varphi_2(k)$ respectively denote the phase of complex X_1^k and X_2^k.

Definition 4. The matrix $\bar{X} = (\bar{X}^0, \bar{X}^1, \cdots, \bar{X}^{N-1})$ obtained by (3) is called as the corrected data matrix for SSFI.

Definition 5. Project vector $\bar{X}^k$ onto unit circle and get $\theta^k = \bar{X}^k / \| \bar{X}^k \|$. The matrix $\Theta = (\theta^0, \theta^1, \cdots, \theta^{N-1})$ is called as the matrix of searching the basis vector, where $\theta^k = (\theta_1^k, \theta_2^k)^T$.

If the DFT coefficients of more than one of the sources, for a given frequency interval Δk, have the same quantity in Eq. (1), the vectors $\bar{X}^k$ lie on the same line. The line is called a pseudo-basis line. However, the data points, which is close to a pseudo-basis line, is almost impossible to have more than two frequency intervals.

Then give $\delta > 0$ and search all the nonzero vectors θ^k almost on the same basis line as following:

$$|\theta_1^{k_i} - \theta_1^{k_j}| + |\theta_2^{k_i} - \theta_2^{k_j}| < \delta . \tag{4}$$

If the nonzero vectors θ^k, for a group of data points $k_1, k_2, \cdots, k_l$, satisfy the requirement of (4), give N_0 and estimate the column $\hat{a}^j$ as:

$$\hat{a}^j = \frac{1}{l} \sum_{i=1}^{l} \theta^{k_i} \qquad when \quad l \geq N_0 . \tag{5}$$

Let N_J be the number of the nonzero vectors θ^k, which is approximately equal to $\hat{a}^j$. Let N_{total} be sum of all the N_J ($j = 1, 2, \cdots, n$).

Above approach may be called as *searching-and-averaging-based method* in the frequency.

3 Source-Recovery Step

Bofill and Zibulevsky first proposed the shortest-path algorithm in [4],[5]. In this paper, we use the algorithm to recover the DFT coefficients of the source signals in frequency domain. Then the source signals are obtained by inverse discrete Fourier transform (IDFT).

4 Experiments and Results

The experiments were conducted on the following sets of signals: a SixFlutes data set and a SixFluteMelodies data set[4]. The two data sets were originated from http://www.ac.upc.es/homes/pau/. These sounds in the SixFlutes data set were so steady (the spectra varied so little over time) that the whole signal could be processed with a single FFT with 32768. The mixtures of using dynamic sources were processed in frames of length L samples and they were multiplied by a Hanning window. A “hop” distance d was used between the starting points of successive frames, leading to an overlap of $L-d$ samples between consecutive frames. Each frame was transformed with a standard FFT of length L. The SixFluteMelodies data set was processed with $L = 8192$ and $d = 3276$.

In all the experiments, a mixing matrix was formed with equally spaced angles, the other two mixing matrixes were randomly formed as table 1.

Table 1. The mixing matrixes A_1, A_2, and A_3

a^j	A_1	A_2	A_3
(a_{11}, a_{12})	$(\cos 15^0, \sin 15^0)$	(-0.787, -0.617)	(0.966, 0.259)
(a_{21}, a_{22})	$(\cos 45^0, \sin 45^0)$	(-0.988, -0.157)	(0.695, 0.719)
(a_{31}, a_{32})	$(\cos 75^0, \sin 75^0)$	(0.352, 0.936)	(0.242, 0.970)
(a_{41}, a_{42})	$(\cos 105^0, \sin 105^0)$	(0.972, -0.236)	(-0.406, 0.914)
(a_{51}, a_{52})	$(\cos 135^0, \sin 135^0)$	(0.615, 0.789)	(-0.891, 0.454)
(a_{61}, a_{62})	$(\cos 165^0, \sin 165^0)$	(0.688, -0.726)	(-0.995, 0.105)

Experiment SixFlutes was repeated three times using the mixing matrixes A_1, A_2 and A_3. Experiment SixFluteMoldies was repeated twice using the mixing matrixes A_1 and A_2. Set parameters $\Delta = 4$, $\delta = 0.2$ and ε (see table 1). After using a searching-and-averaging-based method to search the basis lines, the blurred basis lines in figure 1(c) have been distinct in figure 2, and their projections Θ on unit circle only lie at the six points.

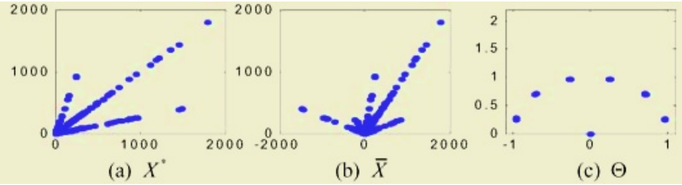


Fig. 2. Scatter plot x_2 vs. x_1 of X^*, $\bar{X}$ and Θ in the experiment SixFluteMelodies I

In the matrix-recovery step, the error in the estimated matrix was measured by the difference between the angles of the estimated and actual basis vectors (columns of the mixing matrix). In source-recovery step, a reconstruction index is SNR ..

Table 2. Parameter Experiments

Experiments	ε	N	N_{total}	N_1	N_2	N_3	N_4	N_5	N_6
SixFlutes I	0.015	32768	174	28	16	22	10	54	44
SixFlutes II	0.020	32768	134	18	16	48	22	14	16
SixFlutes III	0.020	32768	258	54	24	32	12	58	78
SixFluteMelodies I	0.010	40960	400	66	76	188	18	10	42
SixFluteMelodies II	0.010	90112	710	190	110	334	38	10	28

In table 2, because the number N_{total} of the nonzero θ^k for matrix Θ in all the experiments doesn't exceed 710 samples, the approach in the letter is more stable and also faster than k-means or the potential-based method in [4].

Table 3. The difference between the angles of the estimated and actual basis vectors

Experiments	$\hat{a}^1\, vs.a^1$	$\hat{a}^2\, vs.a^2$	$\hat{a}^3\, vs.a^3$	$\hat{a}^4\, vs.a^4$	$\hat{a}^5\, vs.a^5$	$\hat{a}^6\, vs.a^6$
SixFlutes I	0.56^0	0.25^0	0.30^0	0.29^0	0.11^0	0.44^0
SixFlutes II	0.22^0	0.51^0	0.11^0	0.05^0	0.46^0	0.12^0
SixFlutes III	0.57^0	0.02^0	0.19^0	0.50^0	0.15^0	0.18^0
SixFluteMelodies I	0.03^0	0.01^0	0.14^0	0.55^0	0.13^0	0.28^0
SixFluteMelodies II	0.11^0	0.14^0	0.02^0	0.57^0	0.85^0	0.17^0

In the table 3, the mixing matrix was estimated accurately. The result was more exact than Bofill and Zibulevsky's in [4].

Table 4. *SNR* reconstruction indices (dB) for the different experiments (see text)

SixFlutes I	51.89	51.87	49.01	45.91	49.74	54.59
SixFlutes II	43.23	45.04	50.40	48.17	43.42	58.36
SixFlutes III	51.66	52.82	50.78	48.42	49.64	52.88
SixFluteMelodies I	22.10	19.53	14.70	18.43	26.14	28.64
SixFluteMelodies II	19.38	21.38	25.25	25.23	18.38	26.71

In the table 4, *SNR* of our recovered source signals are better than Bofill's and Zibulevsky's in [4].

5 Conclusion

In the paper, we discussed a problem of underdetermined blind sources separation (i.e., BSS with fewer mixtures than sources). The main result of this paper is a searching-and-averaging-based method in the frequency domain. In this novel algorithm, mainly using the durative-sparsity of the modules of the DFT coefficients of the sources, the basis vectors were estimated in a matrix of hundreds nonzero data points. Therefore, this algorithm has lower computational complexity and higher accuracy in the recovered matrix. Several experimental results show the availability of our proposed algorithm.

References

1. Li, Y., Wang, J.: Sequential Blind Extraction of Instantaneously Mixed Sources. IEEE Trans on Signal Processing, **50** (2002) 997-1006
2. Cardoso, J.F.: Blind Signals Separation: Statistical Principles. Proc. IEEE, **86** (1998) 1129-1159
3. Li, Y., Cichocki, A., Amari, S.: Analysis of Sparse Representation and Blind Source Separation. Neural Computation, **16** (2004) 1193-1234
4. Bofill, P., Zibulevsky, M.: Underdetermined Blind Source Separation Using Sparse Representations. Signal Processing, **81** (2001) 2353-2362
5. Theis, F.J., Lang, W.E., Puntonet, C.G.: A Geometric Algorithm for Overcomplete Linear ICA. Neurocomputing, **56** (2004) 381-398

A Neural Network Blind Separation Method Based on Special Frequency Bins

Anqing Zhang[1,2], Xuxiu Zhang[1], Tianshuang Qiu[1], and Xinhua Zhang[2]

[1] School of Electronics and Information Eng., Dalian Univ. of Tech.
Dalian, Liaoning 116024, China.
zhanganq@mail.dlptt.ln.cn
[2] Dalian Naval Academy, Dalian, Liaoning 116018, China

Abstract. It is a usual approach to separate the convolutive mixtures blindly in frequency domain. Many blind separation algorithms proposed for instantaneous mixtures were employed to separate signals in each frequency bin. These approaches must consider all frequencies, and correct the permutation/amplitude of output signals, resulting in a huge computation. In this paper we propose a neural network blind separation approach with a few special frequency bins, which have line spectra or some foreknowing characteristics. The approach can separate convolutive signals effectively with a reduced computation, suitable for the application on real time. The validity and performance of the proposed approach is demonstrated by computer simulation with speech and ship radiating underwater acoustic signals. The comparison between the proposed method and all frequency bins algorithms is performed on their calculation complexity and separation effect.

1 Introduction

Received signals of an array are formed from multi-sources through multi-channels. It is the blind separation problem of convolutive signal mixtures to estimates original signals without priori-knowledge about sources and transmission channel. This kind of blind separation is applied in several areas, such as data communication, speech processing, and ship radiating underwater acoustic processing. Most of the solutions [1], [2] to the MIMO convolutive problem are achieved in time domain based on SOS or HOS, they adaptively adjust demixing filters weights of recurrent neural networks (NN). Unfortunately, BSS approaches in time-domain are poor for long tap signals of convolutive mixtures, with a huge computation. Several authors [3], [4] have proposed many frequency-domain blind separation methods which convert the general problem of convolutive mixtures into the problem of instantaneous mixtures. But, the existed frequency-domain approaches need to consider all frequency bins of signals and correct the permutation/amplitude indeterminacy involves in all frequency bins. Therefore, such methods are not suitable for real time application. However, in practice most signals are band-limited, even with a line-spectrum. Sometimes, the main power of a target signal concentrated in a few particular frequency bins, and then the important property of the target signal is embodied in them. So it is important and possible to separate signals with these particular frequency bins. In this paper we

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 490–495, 2005.

present a new frequency-domain approach developed in the context of unsupervised learning of neural networks and based on a few particular frequency bins. These particular bins can be confirmed by method of line enhancement. The proposed algorithm is significant on not only separating effective for convolutive mixtures but also reducing computational cost. The criterion is proposed to restore the correct order for demixing output signals of difference frequencies. The validity and performance of the proposed approach is demonstrated by computer simulation for both speech and ship radiating underwater acoustic signals.

2 Blind Separation of Convolved Mixtures in Frequency Domain

Assume that N signals s_i are ordered in a vector $\mathbf{S}(t)=[s_1(t),...,s_N(t)]^T$, where t is the time index, the sources are real-valued, non-Gaussian distributed and statistically independent. The M dimension observation vector is $\mathbf{X}(t)=[x_1(t),...,x_M(t)]^T$, $M \ge N$. If mixture system is FIR and is causal, then the observation vector is as follows

$$\mathbf{X}(t)=\sum_{\tau=0}^{P}\mathbf{A}(\tau)\mathbf{S}(t-\tau) \quad , \quad P<\infty . \tag{1}$$

Where $\mathbf{A}(\tau)$ is an unknown $M \times N$ mixture matrix whose elements are FIR polynomials, P is the maximal order of delay. It is instantaneous mixture if $P=1$.

In blind source separation, we are interested in finding a corresponding demixing system, where the output signals $Y(t)$ are described by

$$\mathbf{Y}(t)=\sum_{\tau=0}^{Q}\mathbf{W}(\tau)\mathbf{X}(t-\tau) . \tag{2}$$

Where $\mathbf{W}(\tau)$ is $N \times M$ matrix of linear filters representing the separation system, $\mathbf{Y}(t)=[y_1(t),..., y_M(t)]^T$, a particular output $y_i(t)=\sum_{j=1}^{M} w_{ij}(t) * x_j(t)$, the * is the linear convolution operator, and Q is the order of the FIR filter in the separation system.

In frequency-domain, the convolutive mixture (1) and separation system outputs (2) take the form as

$$\mathbf{X}(\omega)=\mathbf{A}(\omega)\mathbf{S}(\omega), \qquad \mathbf{Y}(\omega)=\mathbf{W}(\omega)\mathbf{X}(\omega) \quad , \quad \forall \omega . \tag{3}$$

Where $\mathbf{X}(\omega)$, $\mathbf{S}(\omega)$, $\mathbf{A}(\omega)$ represent the observation, the sources and the mixing coefficient in frequency domain, respectively. Note that the observation vector $\mathbf{X}(\omega)$ at each frequency bin corresponds to an instantaneous mixture of the sources $\mathbf{S}(\omega)$. Therefore, in order to recover the sources at each frequency, we use a MIMO blind separation system and obtain the demixing matrix $\mathbf{W}(\omega)$ which satisfies equation (4). The sources are optimally recovered when each output in vector $\mathbf{Y}(\omega)$ extracts a single and different source.

$$\mathbf{Y}(\omega)=\mathbf{W}(\omega)\mathbf{X}(\omega)=\mathbf{D}(\omega)\,\mathbf{P}(\omega)\mathbf{S}(\omega) \quad , \quad \forall \omega . \tag{4}$$

where $\mathbf{D}(\omega)$ is a diagonal matrix and $\mathbf{P}(\omega)$ is a permutation matrix. Since the separation matrix at each frequency is independently obtained, the sources may be recovered in a different order and with different amplitudes in some frequency bins. So, we must solve the problem to have the permutation and amplitude consistent respectively, since the sources in time-domain are recovered combining the outputs in all different frequencies by the inverse Fourier transform. On the other hand, we do our best to reduce calculation time of the blind separation algorithm, so as to implement it in real time, such as array signal processing in sonar system.

3 Blind Separation Approach of NN Based on Special Frequency Bin

3.1 Characteristic Frequencies Determination

In general, the most relevant information about the source signals is contained in the waveforms of the source signals in some frequencies. Otherwise, we may want to estimate only one or several desired frequency components with particular statistical features or properties, but discard uninteresting components or noises. Special frequencies are chosen by means of spectrum analysis. Firstly, we utilize the adaptive line spectrum enhancement to process observation signals of array sensors. Secondly, characteristic frequencies are confirmed by searching line spectrum frequency or local maximum for each channel. Lastly, set a threshold to choose special frequencies for BSS. With this method, we can maintain those frequencies whose powers are higher. Assuming special frequencies chosen are $\omega_k, k \in \{1, 2, \cdots, g\}$.

Based special frequencies BSS system is composed of three parts, including the special frequency selecting and STFT stage, BSS stage, and time-domain signal recovery stage. To recover the sources from convolutive mixtures, we propose to use the system shown in Fig.1.

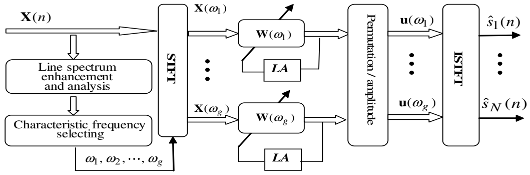


Fig. 1. Separation system of NN configuration based on special frequencies

In the first stage, we apply the STFT to move windows of observations. Sequence of the observation is split into non-overlapped segments with T samples, i.e., $\mathbf{x}_i(t_\tau) = [x_i(t_\tau), x_i(t_\tau + 1), \cdots, x_i(t_\tau + T - 1)]^T$, where t_τ denotes the window's position. We compute the L-point DFT ($L \geq K$) of each window, i.e. $\mathbf{x}_i[\omega_k, t_\tau] = \sum_{m=0}^{T-1} x(t_\tau + m)e^{j\omega_k m}$. where $t_\tau = \tau T, \tau = 0,1,2,\cdots$, $k = 0,\cdots,g$, $\omega_k = 2\pi k / L$ denotes the frequency bin. So equation (1) and (2) are expressed as equation (3) in frequency-domain. Note that we take the observations corresponding to some special frequency bins.

3.2 Blind Separation Approach of NN and Time-Domain Signals Recovering

In frequency-domain, BSS also allows us to estimate FIR filters as demixing filters in each special frequency. We can execute different separation approaches based on Neural Network. In this paper, we apply JADE algorithm [5] to find weights of separation network, but needing to perform modification, matrix transposition to Hermitian transposition. The other algorithms can be used also. So, Bell's and Amari's rules transform to formula $\Delta\mathbf{W} \propto \left[\mathbf{W}^{-1}\right]^H - 2\cdot\varphi(\mathbf{y}(z))\mathbf{x}(z)^H$ and $\Delta\mathbf{W} \propto [\mathbf{I} - \varphi(\mathbf{y}(z))\mathbf{y}(z)^H]\cdot\mathbf{W}$.

Since at each frequency we are applying an algorithm for instantaneous mixtures, it is possible that the permutation matrix $P(z)$ be different for each one of these frequencies. When this occurs, that can lead us to a wrong reconstruction of the spectrum of the recovered sources at output of the separation. We propose a method, which we select a benchmark order of output signals at first special frequency bin (ω_1), and compute the cross-correlation between the benchmark and the outputs in other frequency bins ($\omega_k, k\in\{2,3,\cdots,g\}$), finding and adjusting a consistent rank of output signals. The adjusting criterion is maximum cross-correlation. Then, we cluster the outputs corresponding to the same source.

$$\mathbf{U}_i = \{\ \ u_i(\omega_1) = y_i(\omega_1),\quad u_i(\omega_2) = \max_{\substack{y_l(\omega_2)\\ l=1,\cdots,N}} \left|E[y_i(\omega_1)y_l^*(\omega_2)]\right|, \cdots, u_i(\omega_g) = \max_{\substack{y_l(\omega_g)\\ l=1,\cdots,N}} \left|E[y_i(\omega_1)y_l^*(\omega_g)]\right|\ \ \}\,. \tag{5}$$

With this operation, the permutation matrixes satisfy : $\mathbf{P} = \mathbf{P}(\omega_1) = \mathbf{P}(\omega_2) = \cdots = \mathbf{P}(\omega_g)$

In order to solve the amplitude indeterminacy, the simplest and probably most popular constraint is to simply set the diagonal elements of $\mathbf{W}(\omega_k)$ as one. However, it is still possible that the recorded noise gets amplified at several frequencies. Alternatively, the norm of the rows of $\mathbf{W}(\omega_k)$ can be constrained to unity. This guarantees that the noise does not get amplified, at the expense of a slight signal distortion. In this paper, we force that all outputs $u_i(\omega_k)$ in $\mathbf{U}_i$ have the same amplitude as those of $u_i(\omega_1)$ .Let us consider $u_i(\omega_k) = \delta_{ii}(\omega_k)s_i(\omega_k)$, Set $u_i(\omega_1) = \delta_{ii}(\omega_1)s_i(\omega_1)$, then we compute the other outputs $u_i(\omega_k) \Leftarrow \delta_{ii}(\omega_1)s_i^*(\omega_k)u_i(\omega_k)/|\delta_{ii}^*(\omega_k)|$, ($k\in\{2,3,\cdots,g\}$) . It is easy to show that $u_i(\omega_k) = \delta_{ii}(\omega_1)s_i(\omega_k)$ has the same amplitude as $u_i(\omega_1)$.

The last stage of the separation system is to recover the sources in time-domain. So, the inverse STFT is applied to the outputs $\mathbf{u}(\omega_k)$, $k\in\{1,2,\cdots,g\}$ for recovering the sources

$$\hat{s}_i(t_\tau + m) = \frac{1}{L}\sum_{l=0}^{L-1} u_j[\omega_l, t_\tau]\exp(-j\frac{2\pi lm}{L}), \quad m = 0,\ldots,Q-1\,. \tag{6}$$

where $t_\tau = \tau Q$ with $\tau = 1, 2, \ldots$ is the window position.

4 Simulations and Analysis

In this section we demonstrate the effect of the proposed approach using two groups of sources, the first group is the recorded speech and music signals, the other is two ship radiating underwater acoustic signals. In our experiments, we synthesize observations by mixing matrixes whose coefficients are showed in Fig. 2.

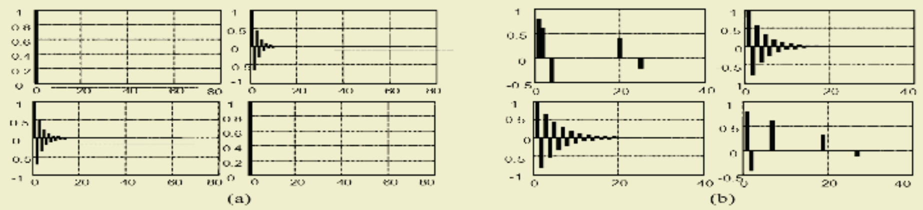


Fig. 2. Coefficient of mixing matrix: (a) coefficient of mixing matrix for speech (b) coefficient of mixing matrix for underwater acoustic

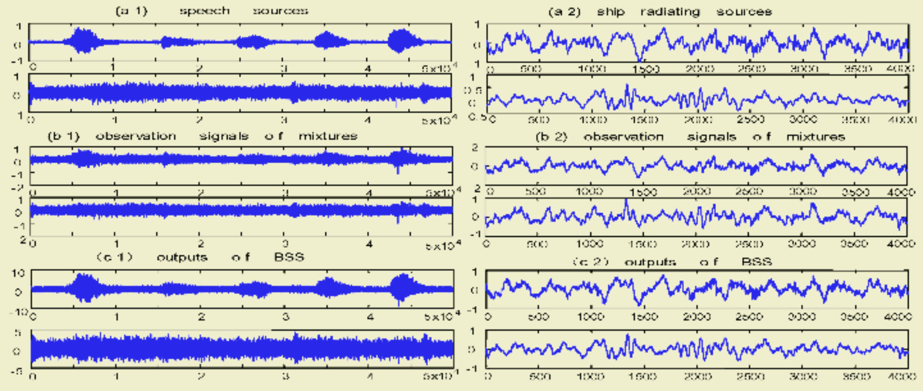


Fig. 3. The blind separation results : (a1,b1,c1) for speech and (a2,b2,c2) for ship radiating signals

Fig. 3(c1) shows the separation results of speech. The parameters of the separation system are the DFT of $L = 2048$, $T = 64$, and the total points of speech are 50000 with sample frequency of 8 kHz. Convolved mixing matrixes coefficients are showed in Fig.2 (a). According to the spectrum analysis of observations, we have chosen the frequency bins $\omega_1 = 16$ and $\omega_2 = 29$, and perform the proposed algorithm. It is clearly that our approach is able to successfully recover the original sources. The separation coefficients are 0.91 and 0.96, respectively.

Fig. 3(c2) shows the separation results of recorded ship radiating underwater acoustic signals. The parameters of the separation system are the DFT of $L = 1024$, $T = 32$, and the total points of speech are 40000 with sample frequency of 25 kHz. Convolved mixing matrixes coefficients are showed in Fig.2 (b). According to the spectrum analysis of ship radiating signals, we choose the frequency bins $\omega_1 = 6$, $\omega_2 = 7$ and $\omega_3 = 10$. The separation coefficients are 0.94 and 0.88, respectively.

Table 1 denotes the computation cost for frequency-domain BSS using a few particular frequency bins and all frequency bins, respectively. Row (A) of Table 1 is the case only for special frequency bins, and row (B) of Table 1 is the case for all frequency bins. Simulation conditions are Intel Pentium 4 CPU of 2.2GHz and MATLAB6.5 software system. In Table 1, s and m denote the second and minute respectively.

Table 1. Computation cost for frequency-domain BSS

	voice signals		ship radiating underwater acoustic	
	L=2048,T=64	L=1024,T=64	L=1024, T=32	L=512, T=32
(A)	0.435 (s)	0.361 (s)	0.145 (s)	0.107 (s)
(B)	7.248 (m)	4.562(m)	3.768(m)	1.882(m)

From the results in Table 1, we conclude that the proposed BSS approach is more practical than the traditional algorithm in frequency-domain. The proposed approach can not only separate convolutive mixture signals effectively, but also reduce the computational cost greatly. Otherwise, it avoids the drawback of existed algorithms that fails to separate mixed signals at some frequency bins, which results in the degeneration of the separation performance.

5 Conclusions

The neural network blind separation based on a few special frequency bins is applied for convolutive mixtures of signals. This proposal is motivated by the fact that the computation cost is huge when all frequency bins are considered for BSS in frequency-domain, and the separation performance is degraded when the poor performance of blind separation happens for existed algorithms in some frequency bins. Our proposed approach can (1) guarantee the effective separation for the dominant components of signals, and (2) reduce the computation. We confirmed in simulation experiment that the proposed approach is effective.

References

1. Parra, L., Spence, C.: Convolutive Blind Source Separation Based on Multiple Decorrelation. In Proc. of NNSP'98, Cambridge, UK (1998) 23-32
2. Kawamoto, M., et al.: A Method of Blind Separation for Convolved Non-stationary Signals. Neurocomputer, **22** (1998) 157-171
3. Smaragdis, P.: Blind Separation of Convolved Mixtures in the Frequency Domain. In Int. Workshop on Indepence and Artificial Neural Networks , New York (1998)
4. Dapena, A., et al.: Inversion of the Sliding Fourier Transform Using only Two Frequency Bins and Its Application to Source Separation. Signal processing, **83** (2003) 453-457,
5. Cardoso, J.F., Souloumiac, A.: Blind Beamforming for Non-Gaussian Signals. IEE-Proceedings-F, **140** (1993) 362-370

Application of Blind Source Separation to Time Delay Estimation in Interference Environments

Gaoming Huang[1,2], Luxi Yang[2], and Zhenya He[2]

[1] Naval University of Engineering, Wuhan, Hubei 430033, China
[2] Department of Radio Engineering, Southeast University, Nanjing, Jiangshu 210096, China
redforce@sohu.com, {lxyang,zyhe}@seu.edu.cn

Abstract. An important signal parameter estimation problem is time delay estimation. The problem of time delay estimation in interference environments is investigated in this paper, with the application of Blind Source Separation (BSS). This novel method can reduce the complexity of estimation and improve the performance of time delay estimation. The important contribution of this new estimation method is that it can reduce the affection of interference to time delay estimation. Simulation result confirms the validity and practicality of the proposed approach, the results of time delay estimation in interference environments are more accurate and stable based on this new method.

1 Introduction

Time delay is caused by different transmitting distance to different receiver units in the receiver array, which is an important parameter to represent a signal. The estimation of time delay has a broad application in radar, sonar, biomedicine and telecommunication, etc. The basic method of time delay estimation includes GCC, generalized bispectrum estimation, adaptive estimation, etc [1]. It has absorbed a great of attention in early time [2]. There are many new estimation methods in recent years [3],[4]. The main task of time delay estimation is to estimate the time difference of the target signals, which is often polluted by interference.

In recent years, particularly after Herault and Jutten's work [5], the research of BSS has become a hotspot in signal processing. A lot of BSS algorithms have been proposed [6],[7]. BSS have been applied in our recent works, such as direction finding [8] and passive location [9], etc. The performance is improved remarkably. BSS can effectively separate or extract source signals only by the mixing measurements, which is just the processing that time delay estimation in interference environments requires. Hereby we proposed applying BSS to time delay estimation in interference environments, which based on mutual information methods [10],[11]. Then stable and accurate estimation results can be obtained after a cross correlation. Structure of this paper can be described as follows: In the Section 2, we will analyze the issue of time delay estimation and pose the problem. In the Section 3, the time-delay estimation algorithm based on BSS will be analyzed. Then some experiments of the algorithm proposed in this paper are conducted in the Section 4. Finally a conclusion is given.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 496–501, 2005.

2 Problem Formulation

The model of time delay between the signals received by separate antennas is shown as Fig. 1.

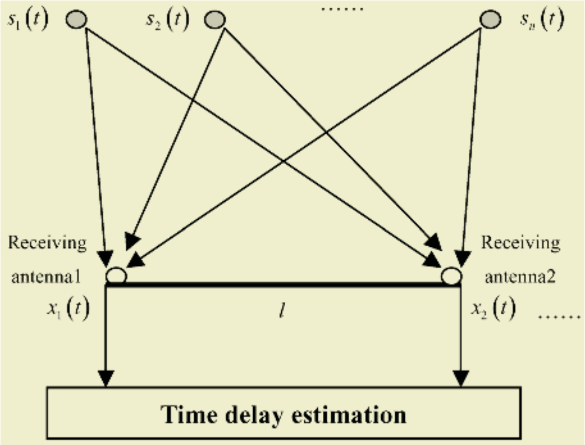


Fig. 1. Signal receiving model

Supposing there are m receiving antennas, the mutual distance is l. If have $n(n \le m)$ narrow band signals, $\boldsymbol{r}(\theta_i)$ is the antenna respond to narrow band signal on direction θ_i, then $\boldsymbol{r}(\theta_i)=[1,e^{j\phi_i},...,e^{j(m-1)\phi_i}]^T$, where $\phi_i = 2\pi l \sin\theta_i/\lambda$, λ is the wavelength. The general receiving signal model could be described as:

$$\boldsymbol{x}(t)=\boldsymbol{H}\cdot\sum_{i=1}^{n}\boldsymbol{r}(\theta_i)s_i(t-\tau_{ij})+\boldsymbol{n}(t)=\boldsymbol{A}\cdot\boldsymbol{s}(t)+\boldsymbol{n}(t). \tag{1}$$

where $\boldsymbol{x}(t)$ is the receiving signals of antenna. $\boldsymbol{A}$ is a $m\times n$ matrix, which is the product of responding function $\mathbf{R}=[\mathbf{r}(\theta_1),...,\mathbf{r}(\theta_n)]$ and the mixing matrix $\boldsymbol{H}$ during the signal transmitting process. $\boldsymbol{s}(t)$ are the source signals including source signals, interference signals etc. τ_{ij} are the delays from source i to sensor j. $\boldsymbol{n}(t)$ are $m\times 1$ dimension noise signals. The signals are assumed as mutually independent and independent with noise in the following analysis. The key problem of time delay estimation is how to eliminate or reduce the affection of interference, which is the main problem that this paper wants to solve.

3 Time Delay Estimation Algorithm

The general time delay estimation method is to apply correlation, but the estimation results will be strongly affected by interference. The new time delay estimation algorithm includes two steps: Blind separation [10],[11] and cross correlation.

3.1 Blind Separation Processing

In order to describe this method clearly, here we set two receive antennas and two sources. One source is the target echo source; the other is an interference source. The receiving model can be described as:

$$\begin{cases} x_1(n) = a_{11}s_1(n) + a_{12}s_2(n-\tau_1) + n_1(n) \\ x_2(n) = a_{21}s_1(n-\tau_2) + a_{22}s_2(n) + n_2(n) \end{cases}. \tag{2}$$

The noise in the Eq. (2) can be reduced to a very little level by the property of spectrum correlation. Most of the signals in communication, navigation, radar and sonar are all have the property of spectrum correlation, but stationary noise does not have this property. Here we suppose that the noise has been cancelled. For the separation we use a feed back network as Fig.2.

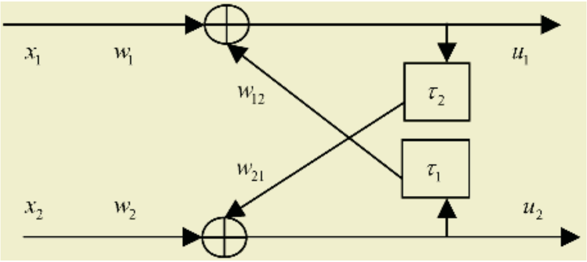


Fig. 2. Separation network for delayed mixtures

In Eq. from this network is written as:

$$\begin{cases} u_1(n) = w_1 x_1(n) + w_{12}u_2(n-\tau_1) \\ u_2(n) = w_2 x_1(n) + w_{22}u_2(n-\tau_2) \\ y_1(n) = g(u_1(n)) \\ y_2(n) = g(u_2(n)) \end{cases}. \tag{3}$$

The weights can be learned by minimizing the mutual information between the components of $\boldsymbol{y} = g(u(n))$, $g(\cdot)$ is a nonlinear function, which is equal to maximize the entropy of $\boldsymbol{y}$, $H = -E[\log(p_y(y))]$, where $p_y(y)$ is the pdf for $\boldsymbol{y}$. p_y can be written as $p_y = p_x / \boldsymbol{J}$, where $\boldsymbol{J}$ is the Jacobian for the separation network. As we cannot have any information about p_x, so maximize the entropy of $\boldsymbol{y}$ is equal to maximize $\log(\det \boldsymbol{J})$.

$$\begin{aligned} \det(\boldsymbol{J}) &= \frac{\partial y_1}{\partial x_1} \cdot \frac{\partial y_2}{\partial x_2} - \frac{\partial y_1}{\partial x_2} \cdot \frac{\partial y_2}{\partial x_1} = \frac{\partial y_1}{\partial u_1} \cdot \frac{\partial u_1}{\partial x_1} \cdot \frac{\partial y_2}{\partial u_2} \cdot \frac{\partial u_2}{\partial x_2} - \frac{\partial y_1}{\partial u_1} \cdot \frac{\partial u_1}{\partial x_2} \cdot \frac{\partial y_2}{\partial u_2} \cdot \frac{\partial u_2}{\partial x_1} \\ &= \frac{\partial y_1}{\partial u_1} \cdot \frac{\partial y_2}{\partial u_2} \cdot \left(\frac{\partial u_1}{\partial x_1} \cdot \frac{\partial u_2}{\partial x_2} - \frac{\partial u_1}{\partial x_2} \cdot \frac{\partial u_2}{\partial x_1} \right) \end{aligned}. \tag{4}$$

Suppose $y_1' = \frac{\partial y_1}{\partial u_1}$, $y_2' = \frac{\partial y_2}{\partial u_2}$, $D = \left(\frac{\partial u_1}{\partial x_1} \cdot \frac{\partial u_2}{\partial x_2} - \frac{\partial u_1}{\partial x_2} \cdot \frac{\partial u_2}{\partial x_1} \right)$, then Eq. (4) can be written as: $\det(\boldsymbol{J}) = y_1' \cdot y_2' \cdot D$, log it as: $\log\det(\boldsymbol{J}) = \log(y_1') + \log(y_2') + \log(D)$, where $D = \left(\frac{\partial u_1}{\partial x_1} \cdot \frac{\partial u_2}{\partial x_2} - \frac{\partial u_1}{\partial x_2} \cdot \frac{\partial u_2}{\partial x_1} \right) = w_1 w_2 - 0 = w_1 w_2$, the objective function can be written as:

$$Q = \log(y_1') + \log(y_2') + \log(w_1 w_2). \tag{5}$$

At first we begin with Δw_1, which can be obtained by the partial derivatives of Eq. (5) as:

$$\begin{aligned} \Delta w_1 \propto \frac{\partial Q}{\partial w_1} &= \frac{1}{y_1'} \cdot \frac{\partial y_1'}{\partial w_1} + \frac{1}{y_2'} \cdot \frac{\partial y_2'}{\partial w_1} + \frac{1}{D} \cdot \frac{\partial D}{\partial w_1} \\ &= \frac{1}{y_1'} \cdot \frac{\partial y_1'}{\partial y_1} \cdot \frac{\partial y_1}{\partial u_1} \cdot \frac{\partial u_1}{\partial w_1} + \frac{1}{y_2'} \cdot \frac{\partial y_2'}{\partial y_2} \cdot \frac{\partial y_2}{\partial u_2} \cdot \frac{\partial u_2}{\partial w_2} + \frac{1}{w_1 w_2} \cdot \frac{\partial (w_1 w_2)}{\partial w_1} \end{aligned} . \tag{6}$$

Denoting $\hat{y}_i = (\partial y_i' / \partial y_i)$, which depends on the cdf used. Eq. (6) can be written as:

$$\Delta w_1 \propto \hat{y}_1 x_1 + 1/w_1 . \tag{7}$$

Similar calculation as in Eq. (6), other parameters give their adaptation rules:

$$\begin{cases} \Delta w_2 \propto \hat{y}_2 x_2 + 1/w_2 \\ \Delta w_{12} \propto \hat{y}_1 u_2 (n - \tau_1) \\ \Delta w_{21} \propto \hat{y}_2 u_2 (n - \tau_2) \\ \Delta \tau_1 \propto -\hat{y}_1 \cdot w_{12} \cdot \dot{u}_2 (n - \tau_1) \\ \Delta \tau_2 \propto -\hat{y}_2 \cdot w_{21} \cdot \dot{u}_1 (n - \tau_2) \end{cases} . \tag{8}$$

Where $\dot{u}_1(n-\tau_2) = \partial u_2 / \partial \tau_2$,$\dot{u}_2(n-\tau_1) = \partial u_1 / \partial \tau_1$. The nonlinear functions shown in the table 1 are proved very effective.

The merits of different nonlinear function may be different. As the experiment results, g_3 is a universality function. Consider about the universality of signals, here we choose g_3 as the nonlinear function.

Table 1. Nonlinear function

No.	$y_i = g(u_i)$	y_i'	$\hat{y}_i$
1	$\tanh(u_i)$	$1 - y_i^2$	$-2y_i$
2	u_i^3	$3u_i^2$	$2/u_i$
3	$1/(1+e^{-u_i})$	$y_i(1-y_i)$	$1-2y_i$

3.2 Correlation Computation

Sometime the delays converge to a wrong value. In order to obtain the stable and accurate results, a cross correlation between the separate signals and mixtures is very important.

$$\begin{cases} r_{x_1u_2}(\tau)=\sum\limits_{n=-\infty}^{\infty} x_1(n)u_2(n-\tau) \\ r_{x_2u_1}(\tau)=\sum\limits_{n=-\infty}^{\infty} x_2(n)u_1(n-\tau) \end{cases} . \tag{9}$$

When τ varying from $-\infty$ to $+\infty$, $r_{x_1u_2}(\tau), r_{x_2u_1}(\tau)$ are the functions of τ. When $\left|r_{x_1u_2}(\tau)\right|$ reach the maximal value at τ_1, then τ_1 is the time difference of the two signals, τ_2 can be obtained by the similar calculation.

4 Simulations

In order to verify the validity of this time delay estimation algorithm applied in this paper, here a series of experiments have been conducted. The background of the experiments are assumed as: there are two separate receiving antennas, the time delay are $\tau_1 = 23$ and $\tau_2 = 26$. The interference source signals is a FM noisy jamming, which can be generated using the corresponding steps from [12]. The first step is generating Gaussian noisy voltage, and then using a six-apices ellipse filter to form noisy bandwidth. At last taking the outputs from filter to a monofier, the outputs represent the outputs of noisy FM jamming signal. The target signal is a FM signal as: $s_{FM}(t)=\cos(2\pi f_0 t+\phi(t))$, where f_0 is the carrier frequency, $\phi(t)$is a modulating component. BSS of delayed sources was conducted at first. The separation results are shown as in Fig. 3.

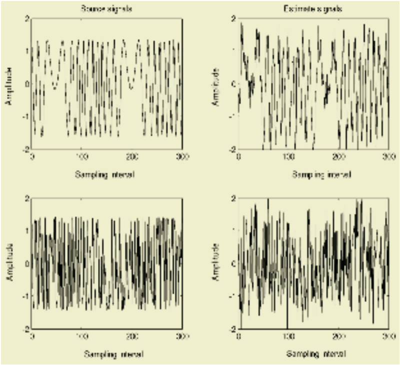


Fig. 3. Blind separation results

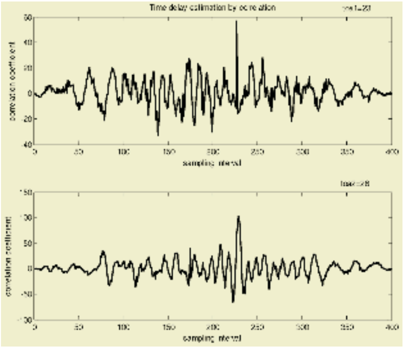


Fig. 4. Time delay estimation results

Then a cross correlation between the receiving signals and separation signals in the interference background is conducted. From the result as shown in Fig.4, we can see that the extremum is distinct, which is just the perfect delay value. A lot of experiments have been conducted by the changing the impact of additive noise. The results show that the estimation of delay time after BSS processing is more accuracy and stable. The anti-jamming property is also excellent.

5 Conclusions

This paper proposes a novel time delay estimation method in interference environments, which applies BSS method based on mutual information. This new time delay estimation approach can effectively overcome the sensitivity to interference. The important contribution of this novel method proposed by this paper is that it can improve the applicability of time delay direction finding. It will play an important role in military and civilian affairs.

Acknowledgements

This work was supported by NSFC (60496310, 60272046), National High Technology Project (2002AA123031) of China, NSFJS (BK2002051) and the Grant of PhD Programmers of Chinese MOE (20020286014).

References

1. Bjorklund, S. and Ljung, L.: A Review of Time-delay Estimation Techniques. Decision and Control, 2003 Proceedings. 42nd IEEE Conference on, **3** (2003) 2502 - 2507
2. Knapp, C.H. and Carter, G.C.: The Generalized Correlation Method for Estimation of Time Delay. IEEE Trans. Acoust. on Speech, Signal Processing, **24** (1976) 320-327
3. Cheng, Z and Tjhung, T.T.: A New Time Delay Estimator Based on ETDE. IEEE Trans. on Acoust., Speech, Signal Processing, **51** (2003) 1859-1869
4. Yi, M., Wei, P., Xiao, X.C. and Tai, H. M.: Efficient EM Initialization Method for Time-delay Estimation. Electronics Letters, **39** (2003) 935-936
5. Jutten, C. and Herault, J.: Blind Separation of Sources, Part I: An Adaptive Algorithm Based on Neuromimetic. Signal Processing, **24** (1991) 1-10
6. Sánchez, A.: Frontiers of Research in BSS/ICA. Neurocomputing, **49** (2002) 7-23
7. Hyvärinen, A., Karhunen, J. and Oja, E.: Independent Component Analysis. J. Wiley (2001)
8. Huang, G.M., Yang, L.X. and He, Z.Y.: Blind Source Separation Using for Time-Delay Direction Finding. ISNN2004, LNCS (2004) 660-665
9. Huang, G.M., Yang, L.X. and He, Z.Y.: Application of Blind Source Separation to a Novel Passive Location. ICA2004, LNCS (2004) 1134-1141
10. Bell, A. and Sejnowski, T.: An Information Maximization Approach to Blind Separation and Blind Deconvolution. Neural Computation, **6** (1995) 1004-1034
11. Torkkola, K.: Blind Separation of Delayed Sources Based on Information Maximization. Proc. IEEE ICASSP1996, (1996) 3509-3512
12. Schleher, D.C.: Electronic Warfare in the Information Age. Artech House (1999)

Blind Identification and Deconvolution for Noisy Two-Input Two-Output Channels

Yuanqing Li[1], Andrzej Cichocki[2], and Jianzhao Qin[3]

[1] Institute for Infocomm Research, Singapore 119613
[2] Laboratory for Advanced Brain Signal Processing
RIKEN Brain Science Institute, Wako Shi, Saitama 3510198, Japan
[3] Institute of Automation Science and Engineering
South China University of Technology, Guangzhou 510641, China

Abstract. This paper discusses blind identification and deconvolution of two-input two-output channels corrupted by noises based on second-order statistics. First, the identifiability of channel is analyzed. By constructing an new criterion, the channel parameters can be identified precisely in the present of noises. Second, the cost function of identification is established and the corresponding algorithm is presented. Next, a feedback model is used for deconvolution, and several important problems, such as the effect of noises in the blind deconvolution of mixed sources and the stability of deconvolution model, are discussed. At last, simulation results are given to illustrate the theoretical results of this paper.

1 Introduction

Blind separation of independent sources from their mixtures has been received considerable attention in recent years. This class of techniques can be used in many real-world applications, such as communications, medical signal processing, speech recognition, image restoration [1], to name a few. The objective of blind separation is to recover source signals from their mixtures. Independent Component Analysis (ICA) deals with the instantaneous mixture case, and Dynamical Component Analysis (DCA) deals with the convolutive mixture case. Although high-order statistics is often used in existing papers e.g., [7], as pointed in [5], second-order statistics is sufficient to solve some DCA problems. There are other papers establish the criteria of separation based on second-order statistics, e.g., [4], [6]. In [5], the noise is not considered in channel parameter identification. It is inevitable that there exist noises in a physical channel. Noise is considered in [6], but is regarded as a source signal and is attenuated through adding sensor. In [4], single-input single-output channel is studied, also the noise is considered, but it is not easy to subtract the corresponding noise correlation matrices from the observation correlation matrices when eigenvalue λ_d in [4] is not large enough.

In this paper, the similar problems as these in [5] are discussed, but the noises in channel are considered. A new criterion or cost function based on second-order statistics is chosen which makes it possible to identify the channel parameters

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 502–507, 2005.

precisely under noisy condition. After the parameters are identified, a feedback model is used for deconvolution, and the errors resulted by noises are analyzed. The stability of deconvolution model is also studied, which is related to the choose of initial values of the model.

2 Problem Formulation

This paper discusses the blind identification and deconvolution of signals from two-input two-output channels corrupted with noises. That is,

$$\begin{cases} y_1(i) = x_1(i) + \sum\limits_{j=I_1}^{I_2} b_1(j)x_2(i-j) + n_1(i), \\ y_2(i) = x_2(i) + \sum\limits_{j=I_1}^{I_2} b_2(j)x_1(i-j) + n_1(i), \end{cases} \tag{1}$$

where $x_1(i), x_2(i)$ are source signals which are assumed to be stationary stochastic series, $b_1(i), b_2(i), i = I_1, \cdots, I_2$ are coefficients of FIR filters B_1, B_2, respectively, of which the lengths are supposed to be equal in this paper; n_1, n_2 are two independent white noises with power density σ_1^2, σ_2^2 respectively; $y_1(i), y_2(i)$ are the outputs of the channel. The object of this paper are to identify the channel filters $B_1(z), B_2(z)$, and extract x_1, x_2 from the observable y_1, y_2.

The following (2) is our blind identification model which is used to separate the source signals from the observable mixtures in [5],

$$\begin{cases} s_1(i,\theta) = y_1(i) - \sum\limits_{j=J_1}^{J_2} d_1(j)y_2(i-j), \\ s_2(i,\theta) = y_2(i) - \sum\limits_{j=J_1}^{J_2} d_2(j)y_1(i-j), \end{cases} \tag{2}$$

where $d_1(i), d_2(i)$ are the coefficients of FIR filters D_1, D_2 respectively which are to be determined; that is, $D_1(z) = \sum\limits_{j=J_1}^{j=J_2} d_1(j)z^{-j}$, $D_2(z) = \sum\limits_{j=J_1}^{j=J_2} d_2(j)z^{-j}$.

Without loss of generality, the following assumptions are made,

C1 The power densities of noises n_1, n_2 can be determined; i.e., σ_1^2, σ_2^2 are known. **C2** $I_1 = J_1, I_2 = J_2$, and $0 \leq I_1 < I_2$.

3 Parameter Identifiability and Identification Algorithm

In this section, the cross correlation will be used to identify the parameters in Eqn. (1). At first, two notations are defined,

$$\theta = [d_1(I_1), \cdots, d_1(I_2), d_2(I_1), \cdots, d_2(I_2)]^T,$$
$$\theta_0 = [b_1(I_1), \cdots, b_1(I_2), b_2(I_1), \cdots, b_2(I_2)]^T.$$

According to (1),(2), and the independency of sources, the Z-transform of $R_{s_1s_2}(l)$, the cross correlation of s_1, s_2, can be obtained as

$$\begin{aligned}R_{s_1s_2}(z) = (B_2(z^{-1}) - D_2(z^{-1}))(1 - D_1(z)B_2(z))R_{x_1x_1}(z) + (B_1(z) - D_1(z))\\ (1 - D_2(z^{-1})B_1(z^{-1}))R_{x_2x_2}(z) - D_2(z^{-1})\sigma_1^2 - D_1(z)\sigma_2^2.\end{aligned} \tag{3}$$

Set

$$\begin{aligned}P_{12}(z) &= R_{s_1s_2}(z) + D_2(z^{-1})\sigma_1^2 + D_1(z)\sigma_2^2\\ &= (B_2(z^{-1}) - D_2(z^{-1}))(1 - D_1(z)B_2(z))R_{x_1x_1}(z) +\\ &\quad (B_1(z) - D_1(z))(1 - D_2(z^{-1})B_1(z^{-1}))R_{x_2x_2}(z).\end{aligned} \tag{4}$$

It is easy to see that if $B_1(z) = D_1(z), B_2(z) = D_2(z)$, then x_1, x_2 can be separated using (2), and $P_{12}(z) = 0$. Thus $P_{12}(z) = 0$ is used as a criterion of identification in Z-domain in this paper. Since $P_{12}(z)$ here is similar to that in (**??**) in [5], through the similar discussion to that in [5], we can find only $B_1(z) = D_1(z)$, $B_1(z) = D_1(z)$ is the true solution of the criterion equation.

Set

$$\bar{R}_{s_1s_2}(l) = R_{s_1s_2}(l) + \sum_{j=I_1}^{I_2} d_2(j)\delta(l+j)\sigma_1^2 + \sum_{j=I_1}^{I_2} d_1(j)\delta(l-j)\sigma_2^2. \tag{5}$$

Obviously, $P_{12}(z) = Z(\bar{R}_{s_1s_2}(l))$, thus it can be obtained the following theorem.

Theorem 1. The two-input two-output system (1) with causal channels is parameter identifiable using only the correlation $\bar{R}_{s_1s_2}(l)$.

Next, we present the identification algorithm. Take the following cost function,

$$V(\theta) = \sum_{l=-U}^{U} [\bar{R}_{s_1s_2}(l)]^2, \tag{6}$$

where U is a sufficiently large positive integer. Noting that x_1, x_2 can not be observed and only y_1, y_2 are observable, thus $\bar{R}_{s_1s_2}(l)$ in (6) should be rewritten in y_1, y_2 instead of x_1, x_2.

It is not difficult to find that for all l, when $\theta = \theta_0$, $\bar{R}_{s_1s_2}(l) = 0$, and that θ_0 is a global minimum of $V(\theta)$. From the cost function in (6), a stochastic gradient-type of algorithm can be obtained,

$$\theta(n) = \left[d_1^{(n)}, d_2^{(n)}\right]^T = \left[d_1^{(n-1)} d_2^{(n-1)}\right]^T - \mu(n)\varDelta V(\theta(n-1)). \tag{7}$$

We can use the following Gauss-Newton algorithm to increase the speed of convergence,

$$\theta(n) = \left[d_1^{(n)}, d_2^{(n)}\right]^T = \left[d_1^{(n-1)} d_2^{(n-1)}\right]^T - \mu(n)\bar{H}^{-1}\varDelta V(\theta(n-1)), \tag{8}$$

where the modified Hessian matrix

$$\bar{H} = 2\sum_{l=-U}^{U} \left[\frac{\partial \bar{R}_{s_1s_2}(l)}{\partial d_1}^T, \frac{\partial \bar{R}_{s_1s_2}(l)}{\partial d_2}^T\right]^T \left[(\frac{\partial \bar{R}_{s_1s_2}(l)}{\partial d_1})^T, (\frac{\partial \bar{R}_{s_1s_2}(l)}{\partial d_2})^T\right]. \tag{9}$$

4 Deconvolution Model and Analysis of Errors and Stability

In this section, the deconvolution model is presented firstly, then, the errors resulted from noises in deconvolution and stability of deconvolution model is studied. Suppose that the channel parameters have been identified; that is $b_1 = d_1, b_2 = d_2$. The following feedback model is used for deconvolution,

$$\begin{cases} y_1(i) = x_1^*(i) + \sum\limits_{j=I_1}^{I_2} d_1(j)x_2^*(i-j), \\ y_2(i) = x_2^*(i) + \sum\limits_{j=I_1}^{I_2} d_2(j)x_1^*(i-j), \end{cases} \tag{10}$$

where x_1^*, x_2^* represent the approximate sources extracted from the observables y_1 and y_2.

Since the precise source signals are x_1, x_2 in (1), thus there would happen errors when (10) is used to realize deconvolution because which are resulted from noises. Let the errors be

$$z(i) = [z_1(i), z_2(i)]' = x^*(i) - x(i). \tag{11}$$

It is easy to find from (1), and (10) or (11) that,

$$\begin{cases} z_1(i) + \sum\limits_{j=I_1}^{I_2} b_1(j)z_2(i-j) = n_1(i), \\ z_2(i) + \sum\limits_{j=I_1}^{I_2} b_2(j)z_1(i-j) = n_2(i). \end{cases} \tag{12}$$

Suppose that $z(i) = 0$ for $i \le 0$. Rewrite (12) in matrix,

$$z(i) + \sum_{j=I_1}^{I_2} B(j)z(i-j) = n(i), \tag{13}$$

where $B(j) = \begin{bmatrix} 0 & b_1(j) \\ b_2(j) & 0 \end{bmatrix}$. In this paper, the norm $||B(j)|| = [b_1^2(j) + b_2^2(j)]^{\frac{1}{2}}$.

We present the following two theorems about error analysis and stability and omit their proofs.

Theorem 2. If $(||B(I_1)|| + \cdots + ||B(I_2)||) < 1$, then the errors resulted from noises do not accumulate when (10) is used to realize deconvolution.

Theorem 3. If $\sum\limits_{j=I_1}^{I_2} ||B(j)|| < 1$, the system of (10) is asymptotically stable.

Theorem 3 shows that the difference between different solutions will tends to zero exponentially though different initial values lead to different solution of (10).

5 Simulation Results

In this section, simulation results are presented to illustrate the theoretical results of this paper. Though the cost function (6) is constructed through modifying the cost function (48) in [5], the two cost functions are the same if they are rewritten in x_1 and x_2, thus the simulations in [5] are suitable for the identification in the case of this paper. In [5], in order to identify the channel parameters, 200 realizations are run, and then the corresponding mean values are obtained. No new simulation on blind identification is given here.

After the channels are identified, (i.e., $b_1 = d_1$, $b_2 = d_2$ are obtained), then the deconvolution model (10) is used to extract sources from the observables y_1 and y_2. Simulations are shown in Figs. 1 and 2. Note that the first two subplots of Figs. 1 and 2 represent the sources x_1 and x_2, the next two subplots represent the mixture signals y_1, y_2 deteriorated by noises, the ensuing two subplots represent the extracted sources x_1^*, x_2^*, the last two represent the errors $x_1^* - x_1$, $x_2^* - x_2$. The noise n_1 is white noise valued in [-0.1,+0.1], the noise n_2 is white noise valued in [-0.05, +0.05]. In the simulations shown in Fig. 1, the deconvolution model (10) is used, and the channel parameters are: $b_1(1) = 0.1$, $b_1(2) = 0.3$, $b_2(1) = 0.15$, $b_2(2) = 0.2$, under which, the condition of Theorem 2 or Theorem 3 is satisfied, and the initial values are set to be ones. In Fig. 2, the separation model (2) is used to extract sources x_1, x_2, where the channel parameters are the same as those in Fig. 1. By comparing the simulation results in Fig. 2 and those in Fig. 1, one can see that the results from the deconvolution model (10) are better.

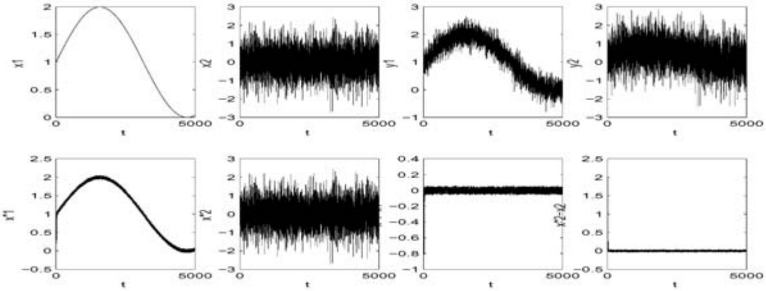


Fig. 1. Deconvolution 1, Initial values set to be 1s

6 Concluding Remarks

In this paper, blind identification and deconvolution of two-input two-output channels are studied based on second-order statistics in noisy environments. Through constructing a new criterion or cost function, the precise channel parameters can be identified theoretically. When identification is completed, a feedback model is used for deconvolution, based on which, the analysis results of errors resulted from noises and stability are presented. If the condition of Theorem

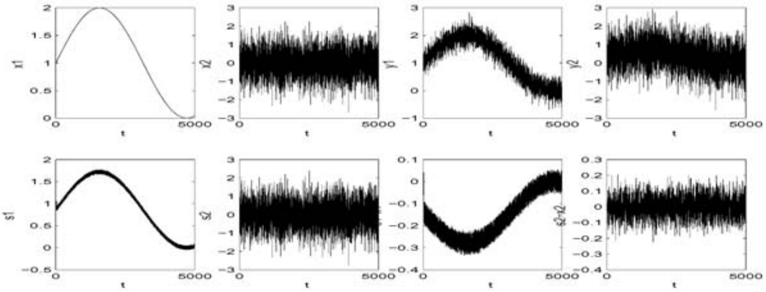


Fig. 2. Deconvolution 2

2 or Theorem 3 is satisfied, through using the deconvolution model in this paper, the separation and deconvolution of mixture signals can be performed one time, and the satisfactory results have been obtained in simulations.

Acknowledgement

This study was supported by the National Natural Science Foundation of China under Grants 60475004 and 60325310, Guangdong Province Science Foundation for Research Team Program under grant number 04205789, the Excellent Young Teachers Program of MOE, P.R. China.

References

1. Kundur, D., Hatzinakos, D.: A Novel Blind Deconvolution Scheme for Image Restoration Using Recursive Filtering. IEEE Trans. on Signal Processing, **46** (1998) 375-390
2. Jutten, C., Herault, J.: Blind Separation of Sources, Part I: An Adaptive Algorithm Based on Neuromimetic Architecture. Signal Processing, **24** (1991) 1-10
3. Bell, A. J., Sejnowski, T. J.: An Information-Maximization Approach to Blind Separation And Blind Deconvolution. Neural Computation, **7** (1995) 1004-1034
4. Tong L., Xu, G.H., Kailath, T.: Blind Identification And Equalization Based on Second-order Statistics: A Time Domain Approach. IEEE Trans. on Information Theory, **40** (1994) 340-349
5. Lindgren, U. A., Broman, H.: Source Separation Using A Criterion Based on Second-order Statistics. IEEE Trans. on Signal Processing, **46** (1998) 1837-1850
6. Sahlin, H., Broman, H.: Separation of Real-world Signals. Signal Processing, **64** (1998) 103-133
7. Inouye, Y., Hirano, K.: Cumulant-based Blind Identification of Linear Multi-input Multi-output Systems Driven by Colored Inputs. IEEE Trans. on Signal Processing, **45** (1997) 1543-1552

A Novel Blind Deconvolution Method for Single-Output Chaotic Convolution Mixed Signal*

Xiefeng Cheng[1], Yong Zhang[1], Zhiquan Feng[1], Ju Liu[2], and Huibo Hu[2]

[1] School of Control Science and Engineering, Jinan University
Jinan 250022, China
{cse_cxf,cse_zhangy,ise_fzq}@ujn.edu.cn
[2] School of Information Science and Engineering, Shandong University
Jinan 250100, China
{Juliu,Huhuibo}@sdu.edu.cn

Abstract. According to the chaotic characteristics, a deconvolution filter for single-output chaotic convolution mixed signal is designed by the linear predication error analysis. Then corrected estimated error from the output data of this filter by the singular spectrum analysis. This blind deconvolution algorithm aiming to special object has the practical meaning to expand the application of the blind signal separation and ulterior research on the character of this special signal. And some simulation results verified the effectiveness and adaptability of this proposed method.

1 Introduction

Chaos is a ubiquitous natural phenomenon in the complex non-linear systems, such as heart system, neuron network, ordinary differential equations, and planar non-linear mapping. So the extraction and the separation of the chaotic signals under the various the conditions are the very important topics in the signal processing. If the mechanism of the dynamics of the chaos system has been known, the method separating the chaotic signal from the mixed signals is to find a time sequence not only to meet the known rule of the chaos dynamics but also closes to the obtained observation sequence. But if the mechanism of the dynamics is unknown, the common method is phase space reconstitution [1].

Obviously, knowing the chaotic dynamics mechanism ahead, the request can not meets for the blind separation. And after convolution with the system function, the chaotic signal is not homeomorphous with the original chaotic dynamic equation. So the original chaotic signals can no be directly recovered by adopting the phase space reconstitution. Comparing with the instantaneous mixture, the solution to the convolution mixture is more difficulty, and is nearer the reality.

In this paper, according to the similar random characteristic of the chaos and the bind signal processing technique, the deconvolution filter on chaotic signal is designed based on the prediction error analysis. Then corrected estimated error from the

* This work was supported by National Natural Science Foundation of Chain (6037031).

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 508–513, 2005.

output data of this filter by the singular spectrum analysis. So the blind deconvolution of the source signals and the transmission function in single-output chaotic convolution system can be realized by using this novel method. And some simulation results verified the effectiveness and adaptability of this method.

2 Description of the Method

Let $s(n)$ be the chaotic signal, $h(n)$ be the impulse response of the channel, and then the convolution model can be expressed as

$$y(n)=\sum_{n=0}^{\infty}h(n)s(t-n) \tag{1}$$

Assume that the length of the observed chaotic convolution mixed signal $y(n)$ is N, then design a P-order deconvolution filter and let $\hat{y}(n)$ be the estimation of $s(n)$.

Based on the linear prediction principle, the white noise sequence can be modeled by a linear difference equation. Because of the similar random character of the chaos, $\hat{y}(n)$ and $y(n)$ has the following relationship [2]

$$\hat{y}(n)=-\sum_{k=1}^{P}a_{pk}y(n-k) \tag{2}$$

$$\hat{y}(n-p)=-\sum_{k=1}^{P}a_{pk}y(n-p+k) \tag{3}$$

Because $\hat{y}(n)$ is the weighted linear combination of the data before $y(n)$, it is a forward prediction model. While $\hat{y}(n-p)$ is the weighted linear combination of the data after $y(n)$, it is a backward prediction model. Here, the forward error is

$$\begin{aligned}e_p(k)&=y(n)-\hat{y}(n)\\&=y(n)+\sum_{k=1}^{p}a_{pk}y(n-k)\end{aligned} \tag{4}$$

Based on the solution of the Yule-walker Equation proposed by Levinson [3], we have

$$a_{pk}=a_{p-1,k}+k_pa_{p-1,p-k} \tag{5}$$

Then

$$e_p(n)=e_{p-1}(n)-k_pB_{p-1}(n-1) \tag{6}$$

Where

$$e_{p-1}(n)=y(n)+\sum_{k=1}^{p-1}a_{p-1,k}y(n-k)$$

$$B_{p-1}(n-1)=y(n-p)+\sum_{k=1}^{p-1}a_{p-1,k}y(n-p+k)$$

In a similar way, the backward error can also be obtained.

$$\begin{aligned} B_p &= y(n-p) - \hat{y}(n-p) \\ &= y(n-p) + \sum_{k=1}^{p} a_{pk}\, y(n-p+k) \end{aligned} \tag{7}$$

Then

$$B_p = B_{p-1}(n-1) - k_p e_{p-1}(n) \tag{8}$$

Combing formulas (4) and (7), we can find: when $p=0, e_0(n) = B_0 = y(n)$.

To minimize the forward error and the backward error simultaneously, we have

$$\partial E[e_p^2(n) + B_p^2(n)] \big/ \partial k_p = 0$$

$$E[-2B_{p-1}(n-1)e_{p-1}(n) + k_p e_{p-1}^2(n) + k_p B_{p-1}^2(n-1)] = 0$$

$$\therefore \qquad k_p = \frac{2\mathrm{E}\left[B_{p\text{-}1}(n-1)e_{p-1}(n)\right]}{\mathrm{E}[e_{p-1}^2(n) + B_{p-1}^2(n-1)]} \tag{9}$$

Based on the statistical character of the chaos, the chaotic signals have the ergodic character. So the statistical average can be substituted by the time average.

$$k_p = \frac{2\sum_{\mathrm{n}}\left[B_{p\text{-}1}(n-1)e_{p-1}(n)\right]}{\sum_{n}[e_{p-1}^2(n) + B_{p-1}^2(n-1)]} \tag{10}$$

Then after iterative computations, the coefficients at different orders of the deconvolution filter aiming to the chaos can be obtained.

The setting of order P of the chaotic filter is very important. It can be obtained by the minimal descriptive length criterion

$$\min(f(P)) = N \ln \sigma_p{}^2 + \frac{P}{2}(\ln N) \tag{11}$$

where σ_p is the prediction error, and with the increase of the data length N, it should tend to zero. When $f(P)$ puts up the obvious minimal value, P can be determined.

Because chaotic signal is similar with noise but not the real noise and there are multiform interferences, the recovered signal $\hat{y}(n)$ after chaotic filter is not as the same as the source signal $s(n)$. The estimated error Δ will bring consequentially, that is $\hat{y}(n) = s(n) + \Delta$. How to recover $s(n)$ from $\hat{y}(n)$?

Based on Vautard s singular spectrum analysis (SSA) theory [4], the chaotic attractor orbit matrix X can be reconstituted by using $\hat{y}(n)$. If there is no estimated error Δ, X takes on strangeness. While if there is estimated error Δ, X does not take on strangeness. To perform singular value decomposition (SVD) to X, the singular values $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k \geq 0$ can be got. Here, k is the number of the singular values, and it is related to the system. Let

$$s_i = \log\left(\lambda_i \Big/ \sum_{j=1}^{k} \lambda_j \right) \tag{12}$$

Here, $s_1, s_2, \cdots, s_k$ are called as singular spectrums of the system, which show the relative neighbor of the energy-state variables in the whole system. Generally speaking, only the foreside has large value which corresponds to the characteristic component of the chaotic signal, but other part has lesser value in singular spectrum of the system which corresponds to the characteristic component of the estimate error Δ.

The lagged covariance matrix of $\hat{y}(n)$ is a Toeplitz matrix, and it can be expressed as

$$T_x = \begin{bmatrix} c(0) & c(1) & \cdots & c(D-1) \\ c(1) & c(0) & \cdots & \vdots \\ \vdots & \vdots & \cdots & c(1) \\ c(D-1) & \cdots & c(1) & c(0) \end{bmatrix} \tag{13}$$

where *D* is the dimension of embedding, *c* (*j*) is the covariance of $\hat{y}(n)$ when delay time is *j*, and

$$c(j) = \frac{1}{N-j} \sum_{i=1}^{N-j} \hat{y}_i \hat{y}_{i+j} \qquad 0 \le j \le D-1 \tag{14}$$

Because T_x is a non-negative symmetric matrix, its eigenvalues are also non-negative. Arranging these eigenvalues according to decline sequence, we have $e_1 \ge e_2 \ge \cdots \ge e_m \ge 0$, where e_k corresponds to the eigenvector E_k and is called as experience orthodoxy function (EOF). Define the k th principal component (PC) as the orthogonal projection coefficient of $\hat{y}(n)$ mapped on E_k:

$$a_i^k = \sum_{j=1}^{D} \hat{y}_{i+j} E_j^k \qquad 0 \le i \le N-D \tag{15}$$

Thus, $\hat{s}(n)$ can be reconstructed by using Q PCs and EOFs.

$$\hat{s}_{i+j} = \sum_{k=1}^{Q} a_i^k E_j^k \qquad 1 \le j \le D \tag{16}$$

The estimated error Δ can also be obtained by using residual PCs and EOFs:

$$\Delta_{i+j} = \sum_{k=Q+1}^{D} a_i^k E_j^k \qquad 1 \le j \le D \tag{17}$$

According to the physical features of chaos, the blind deconvolution method for single-input and single-output chaotic convolution mixed signal can be got based on the prediction error analysis and the singular spectrum analysis. Followings are the steps of this method.

[Step 1] According to the observation data $y(n)$, the estimated data $\hat{y}(n)$ can be given by using the chaotic deconvolution filter. And let initial value of Q is smaller then the number of variables in the chaotic dynamics system. For example Q=1.

[Step 2] Based on $\hat{y}(n)$, to reconstruct $\hat{s}(n)$ by using Q PCs and EOFs.

[Step 3] Comparing the comparability between the results in this and last iteration. If they are similar, Q=Q+1,repeat the step 2, or else do the step 4.

[Step 4] Perform decovolution by using $\hat{s}(n)$ and $\hat{y}(n)$ to solve $\hat{h}(n)$.

There are many methods to solve $\hat{h}(n)$. That is, we can use the deconvolution filter again or select other deconvolution methods. In this paper, the division algorithm in z -domain is adopted. Perform FFT on the sequence directly to realize the z-transform, and then adopt the division in the frequency domain, and to wipe influenced by the complex number, finally the real $\hat{h}(n)$ can be obtained.

3 Simulations

Experiment 1. Blind deconvolution of Chen's chaotic convolution signals.

According to the Chen's equation

$$\begin{aligned} x' &= a(y-x) \\ y' &= (c-a)x-xz+cy \\ z' &= xy-bz \end{aligned} \tag{18}$$

when $a=35, b=3$ and $c=28$, and the initial values $x(0)=$ -10, $y(0)=0$ and $z(0)=27$, the chaos will be produced. And the one-dimension output can be regarded as the chaotic source signal $s(n)$. Let $h(n)=0.1\sin\omega t, y(n)=s(n)*h(n)$, P=3. $y(n)$, $\hat{s}(n)$ and $\hat{h}(n)$ can be obtained. The waveforms of these signals are shown in Figure 1.

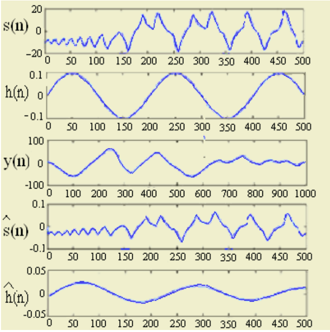


Fig. 1. Results of Experiment 1

There are similitude coefficients [5] of the output $\beta(s(n), \hat{s}(n))=0.9928$ and $\beta(h(n), \hat{h}(n))=0.9265$. The Similitude phase diagram [5] shows that there is only the amplitude difference between $s(n)$ and $\hat{s}(n)$. But there is no phase difference. While there are not only amplitude but also phase differences between $h(n)$ and $\hat{h}(n)$.

Experiment 2. Blind deconvolution of the EEG convolution mixed signals

EEG signal has the chaotic character [1]. Separating EEG signal from the convolution mixed signals has practical significance. Sample 100 points from the EEG signal as the sequence of the original signal $s(n)$ and sample 30 negative points from Rayleign sequence as the system function $h(n)$, $y(n) = s(n) * h(n)$. $y(n)$ and $\hat{s}(n)$ can be obtained. The waveforms of these signals are shown in Figure 2.

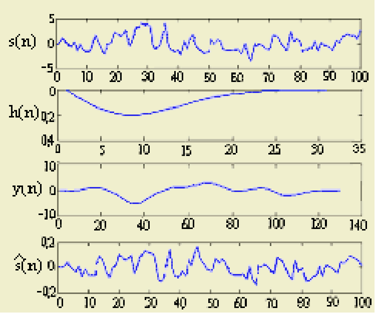


Fig. 2. Results of Experiment 2

4 Conclusion

These results indicate that the resemble coefficient is near 1, which means that the separating result is good. And the length of $h(n)$ influents the separating result greatly. The length of $h(n)$ is nearer that of $s(n)$, the separating result is better. In addition, the order P of the chaotic convolution filter and the iterative number Q of estimated error correction by the singular spectrum analysis have great influence to the separating result.

Because the chaotic signals are ubiquitous in the nature, the blind processing technique of the chaotic convolution mixed signals is practical. This method fully uses the physical features of the chaos. So this blind deconvolution algorithm aiming to special object has the practical meaning to expand the application of the blind signal separation and ulterior research on the character of these special signals.

References

1. Goldberger, A.L., West, B.J.: Chaos and Fractals in Human Physiology. Int. J. Scientific American, **262** (1990) 42-49
2. liu, J., He, Z.Y.: A Survey of Blind Source Separation and Blind Deconvolution. China. J. Acta Electronica Einica, **30** (2002) 591-597
3. Hsue, J.J, Yagle, A.E.: Blind Deconvolution of Symmetric Noncausal Impulse Responses Using Two-Sided Linear Prediction. Int. J. Signal Processing, **42** (1994) 1509-1518
4. Vautard, R., Ghil, M.: Singular Spectrum Analysis in Nonlinear Dynamics with to Paleoclimatic Time Series. Int. J. Phys. D, **35** (1992) 95-126
5. Cheng, X.F., Liu, J.: A Novel Denoising Algorithm Based on Feed Forward Multilayer Blind Separation. International Symposium On Neural Networks, ISNN (2004) 702-707

Stability Analysis of Multichannel Blind Deconvolution

Bin Xia and Liqing Zhang

Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai 200030, China
xbin@sjtu.edu.cn

Abstract. In our previous work [1], we studied the geometrical structures of finite impulse response (FIR) filters and present a permutable cascade decovolution structure based on the filter decomposition [2]. The stability of algorithms was not discussed in [1]. In this paper, we further analyze the cascade structures of filter decomposition and obtain the stability conditions of proposed algorithms. Finally, we give some examples to illustrate the stability conditions are sufficiency.

1 Introduction

Blind deconvolution is to retrieve the independent source signals from sensor outputs by only using the sensor signals and certain knowledge on statistics of the source signals. It is related to independent component analysis (ICA) and blind source separation (BSS) closely. The mixing model of blind deconvolution is a convolutive mixture, while the standard ICA model is time-invariant instantaneous mixture. Blind deconvolution is applicable in many various fields, such as wireless telecommunication, audio and acoustics, image enhancement and biomedical signal processing (EEG/MEG signals) [3–6].

In real world, the mixing model of blind deconvolution is usually described by a non-minimum phase system. So the problem becomes very difficult because the mixing filters are complicated. Zhang *et al* [2] present a new blind deconvolution algorithm which based on demixing filter decomposition. By decomposing the filter into a causal FIR filter and an anti-causal FIR filter, the difficult problem is divided into some easier sub-tasks. We further simplify the filter decomposition structure and decompose the demixing filter into a scalar anti-causal filter and a causal filter. The structure becomes simple and flexible because the two sub-filters are permutable and the anti-causal filter is scalar. In this paper, we study the stability of blind deconvolution algorithm based on filter decomposition approach. We analyze the permutable cascade structure and find that it is helpful to derive the stability conditions.

2 Learning Algorithm Based on Filter Decomposition

In this section, we only briefly outline the blind deconvolution algorithm based on demixing filter decomposition approach. For the details of this algorithm, the reader is referred to [1].

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 514–519, 2005.

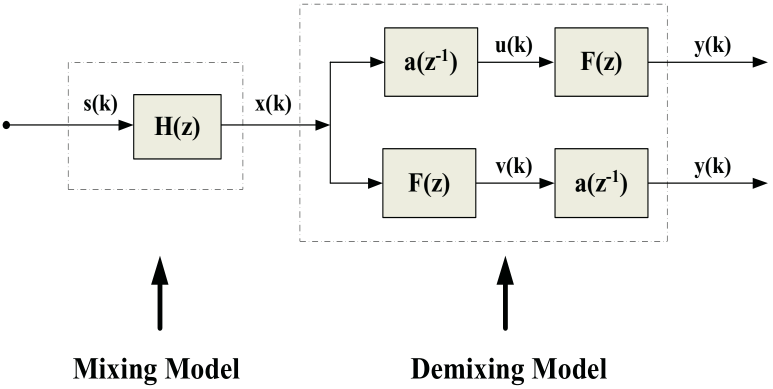


Fig. 1. Illustration of filter decomposition for blind deconvolution.

Consider a convolutive multichannel mixing model, linear time-invariant (LTI) and non-causal systems of form

$$\mathbf{x}(k) = \mathbf{H}(z)\mathbf{s}(k), \tag{1}$$

where $\mathbf{H}(z)$ is mixing filter, $\mathbf{s}(k)$ is an n-dimensional vector of source signals with mutually independent components and $\mathbf{x}(k)$ is the vector of the sensor signals.We introduce a multichannel LTI systems as the demixing model

$$\mathbf{y}(k) = \mathbf{W}(z)\mathbf{x}(k), \tag{2}$$

where $\mathbf{W}(z)$ is demixing filter, $\mathbf{y}(k)$ is an n-dimensional vector of the estimated signals.

As showing in Fig.1, we decomposed $\mathbf{W}(z)$ into permutable cascade structure. The permutable property is helpful to develop the efficient algorithms and to derive the stability conditions. We will discuss it in the following sections. The filter decomposition is described as

$$\mathbf{W}(z) = \mathbf{a}(z^{-1})\mathbf{F}(z) \quad \text{or} \quad \mathbf{W}(z) = \mathbf{F}(z)\mathbf{a}(z^{-1}), \tag{3}$$

where $\mathbf{a}(z^{-1}) = \sum_{p=0}^{N} \mathbf{a}_p z^p$ is an anti-causal scalar FIR filter and $\mathbf{F}(z) = \sum_{p=0}^{N} \mathbf{F}_p z^{-p}$ is a causal FIR filter.

In order to derive the natural algorithms for both sub-filters, we introduce two non-holonomic transforms.

$$d\mathbf{X}(z) = d\mathbf{F}(z) * \mathbf{F}^{\dagger}(z) \tag{4}$$

$$d\mathbf{b}(z) = d\mathbf{a}(z^{-1}) * \mathbf{a}^{-1}(z^{-1}). \tag{5}$$

By minimizing the mutual information of the output signals and using the non-holonomic transforms, we can develop the natural gradient algorithms as follows [1]

$$\triangle\mathbf{F}_p = -\eta \sum_{q=0}^{p} (\delta_{0,q}\mathbf{I} - \boldsymbol{\varphi}(\mathbf{y}(k))\mathbf{y}^T(k-q))\mathbf{F}_{p-q} \tag{6}$$

$$\triangle\mathbf{a}_p = -\eta \sum_{q=0}^{p} (\boldsymbol{\varphi}^T(\mathbf{y}(k))\mathbf{y}(k+q))\mathbf{a}_{p-q} \tag{7}$$

where $\boldsymbol{\varphi}(\mathbf{y}) = (\varphi_1(y_1), \ldots, \varphi_n(y_n))^T$ is the vector of non-linear activation functions, which are closely related to the probability density functions of source signals.

3 Stability of Learning Algorithm

In this section, we discuss stability conditions for blind deconvolution algorithms for non-minimum phase systems. Amari *et al* [7] derived the stability conditions for ICA/BSS. Zhang *et al* [8] analyzed the stability of blind deconvolution and present the stability conditions. The proposed algorithms, developed by using filter decomposition, are different with the algorithms in [8]. So the stability conditions in [8] cannot directly be applied to the algorithm developed for non-causal demixing filters.

From Eqs. (6) and (7) we can rewrite two non-holonomic transforms as follows

$$\frac{d\mathbf{X}_p}{dt} = \eta(\delta_{0,p}\mathbf{I} - \boldsymbol{\varphi}(\mathbf{y}(k))\mathbf{y}^T(k-p)), \tag{8}$$

$$\frac{d\mathbf{b}_p}{dt} = -\eta(\boldsymbol{\varphi}^T(\mathbf{y}(k))\mathbf{y}(k+p)), \tag{9}$$

where $p = 0, 1, \ldots, N$. To analyze the asymptotic properties of the learning algorithms, we take expectation on the above equation

$$\frac{d\mathbf{X}_p}{dt} = \eta\left(\delta_{0,p}\mathbf{I} - E[\boldsymbol{\varphi}(\mathbf{y}(k))\mathbf{y}^T(k-p)]\right), \tag{10}$$

$$\frac{d\mathbf{b}_p}{dt} = -\eta(E[\boldsymbol{\varphi}^T(\mathbf{y}(k))\mathbf{y}(k+p))], \tag{11}$$

If the variational matrix at equilibrium point is negative definite, then the system is stable in the vicinity of the equilibrium point. Taking a variation $\delta\mathbf{X}_p$ on $\mathbf{X}_p$ and a variation $\delta\mathbf{b}_p$ on $\mathbf{b}_p$ respectively, we have

$$\frac{d\delta\mathbf{X}_p}{dt} = -\eta E\left[\boldsymbol{\varphi}^{'}(\mathbf{y}(k))\delta\mathbf{y}\mathbf{y}^T(k-p) + \boldsymbol{\varphi}(\mathbf{y}(k))\delta\mathbf{y}^T(k-p)\right], \tag{12}$$

$$\frac{d\delta\mathbf{b}_p}{dt} = -\eta E\left[[\boldsymbol{\varphi}^{'}(\mathbf{y}(k))]^T\delta\mathbf{y}(k)\mathbf{y}(k+p) + \boldsymbol{\varphi}^T(\mathbf{y}(k))\delta\mathbf{y}(k+p)\right]. \tag{13}$$

Furthermore, we write the differential expression of $\delta\mathbf{y}(k)$

$$\begin{aligned}\delta\mathbf{y}(k) &= [\mathbf{a}(z)\delta\mathbf{F}(z) + \delta\mathbf{a}(z)\mathbf{F}(z)]\,\mathbf{x}(k)\\ &= [\delta\mathbf{X}(z)] + \mathbf{I}\delta\mathbf{b}(z)]\,\mathbf{y}(k).\end{aligned} \tag{14}$$

In [1], we know the matrix $\mathbf{F}_0$ is nonsingular. This means that the learning algorithms keeps the filters $\mathbf{F}(z)$ and $\mathbf{a}(z)$ on the same manifold with the initial filter. This property implies that the equilibrium point of the learning algorithms satisfy the following equation

$$E\left\{\mathbf{I} - \boldsymbol{\varphi}(\mathbf{y}(k))\mathbf{y}^T(k)\right\} = 0. \tag{15}$$

Using the mutual independence and i.i.d. properties of the output signals $y_i, i = 1, \ldots, n$ and the normalized condition (15), we deduce (12) and (13) into

$$\begin{aligned} \frac{d\delta\mathbf{X}_p}{dt} &= -\eta E\{\boldsymbol{\varphi}^{'}(\mathbf{y}(k))[([\delta\mathbf{X}(z)] + \mathbf{I}\delta\mathbf{b}(z)]\mathbf{y}(k))]\mathbf{y}^T(k-p) \\ &\quad + \boldsymbol{\varphi}(\mathbf{y}(k))\mathbf{y}^T(k-p)[(\delta\mathbf{X}(z) + \delta\mathbf{b}(z))^T]\}, \qquad (16) \\ \frac{d\delta\mathbf{b}_p}{dt} &= -\eta E\{[\boldsymbol{\varphi}^{'}(\mathbf{y}(k))]^T(\delta\mathbf{X}(z) + \delta\mathbf{b}(z) * \mathbf{I})\mathbf{y}(k)\mathbf{y}(k+p) \\ &\quad + \boldsymbol{\varphi}^T(\mathbf{y}(k))[\delta\mathbf{X}(z) + \delta\mathbf{b}(z)] * \mathbf{I}]\mathbf{y}(k+p)\}. \qquad (17) \end{aligned}$$

When $p = 0$,

$$\frac{d\delta\mathbf{X}_0}{dt} = -\eta E[\boldsymbol{\varphi}^{'}(\mathbf{y}(k))\delta\mathbf{X}_0\mathbf{y}(k)\mathbf{y}^T(k) + \boldsymbol{\varphi}(\mathbf{y}(k))\mathbf{y}^T(k)\delta\mathbf{X}_0^T], \qquad (18)$$

$$\frac{d\delta\mathbf{b}_0}{dt} = 0. \qquad (19)$$

Rewriting the (18) into component form

$$\frac{d\delta\mathbf{X}_{0,ij}}{dt} = -\eta[k_i\sigma_j^2\delta\mathbf{X}_{0,ij} + \delta\mathbf{X}_{0,ji}], \qquad (20)$$

$$\frac{d\delta\mathbf{X}_{0,ji}}{dt} = -\eta[k_j\sigma_i^2\delta\mathbf{X}_{0,ji} + \delta\mathbf{X}_{0,ij}], \qquad (21)$$

for $i \neq j$, and

$$\frac{d\delta\mathbf{X}_{0,ii}}{dt} = -\eta(m_i + 1)\delta\mathbf{X}_{0,ii}, \qquad (22)$$

for $p = 1, \ldots, N$, and $i, j = 1, \ldots, n$, where $m_i = E[\varphi(y_i)y_i^2]$, $k_i = E[\varphi_i(y_i)]$, $\sigma_i^2 = E[|y_i|^2]$, $i = 1, \ldots, n$. The stability conditions of Eqs. (20) , (21) and (22) are given by

$$\begin{aligned} k_i &> 0, &\quad for &\quad i = 1, \ldots, n, &\qquad (23) \\ k_i k_j \sigma_i^2 \sigma_j^2 &> 1, &\quad for &\quad i, j = 1, \ldots, n, &\qquad (24) \\ m_i + 1 &> 0, &\quad for &\quad i = 1, \ldots, n. &\qquad (25) \end{aligned}$$

Similarly, for $p \neq 0$, we can obtain the stability condition as follows

$$\sum_i k_i\sigma_i^2 > \sum_i (k_i\sigma_i^2)^{-1}. \qquad (26)$$

Due to the limited space, the detailed derivations will left in a separated full paper.

In summary, we obtain the total stability conditions for natural gradient algorithm of blind deconvolution by combining above two parts stability conditions.

$$\begin{aligned} k_i &> 0, &\quad for &\quad i = 1, \ldots, n, &\qquad (27) \\ k_i k_j \sigma_i^2 \sigma_j^2 &> 1, &\quad for &\quad i, j = 1, \ldots, n, &\qquad (28) \\ m_i + 1 &> 0, &\quad for &\quad i = 1, \ldots, n, &\qquad (29) \\ \sum_i k_i\sigma_i^2 &> \sum_i (k_i\sigma_i^2)^{-1}. & & &\qquad (30) \end{aligned}$$

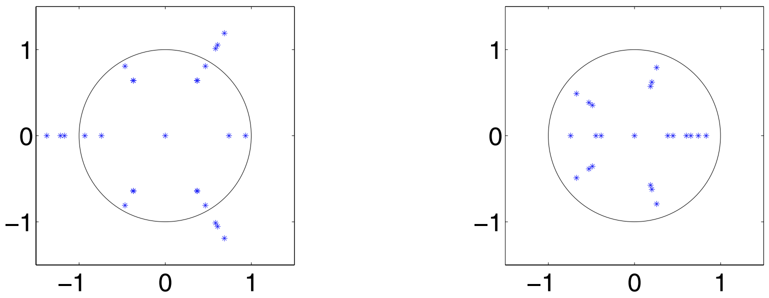


Fig. 2. (a) Zero distributions of mixing model; (b) Pole distributions of mixing model.

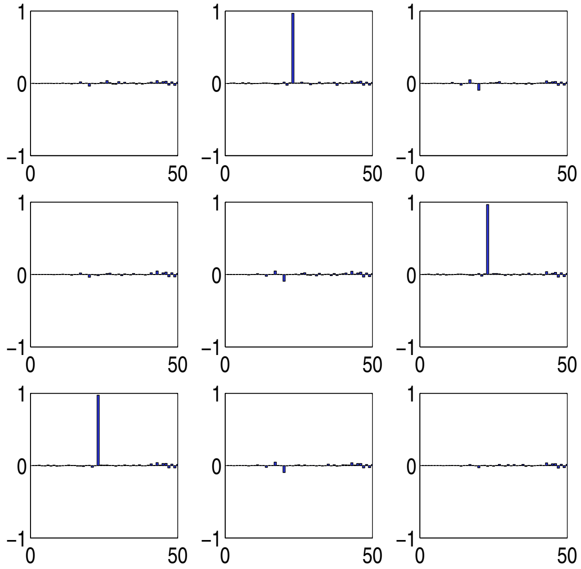


Fig. 3. The coefficients of the global function after convergence.

4 Simulation

In this section, we use computer simulation example to show the efficiency and stability of blind deconvolution algorithms. We use the state space approach to generate nonminimum phase system with three channels. In Fig.2, we plotted the distribution of zeros and poles of mixing model $\mathbf{H}(z)$. It is obvious that the mixing model $\mathbf{H}(z)$ is stable and non-minimum phase. After 30 times iterations, the algorithm converges to stable point and the deconvolution result is good. Fig.3 shows the coefficients of the global function.

5 Conclusions

In this paper, we further analyze the demixing filter structures of blind deconvolution [1]. Using the permutable character of this cascade structure, we derive the stability conditions for non-minimum phase systems. The simulation results show the proposed algorithms are stable and efficient when the stability conditions are satisfied.

Acknowledgments

The project was supported by the National Natural Science Foundation of China under grant 60375015 and the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry.

References

1. Bin Xia, L. Q. Zhang. Multichannel Blind Deconvolution of Non-minimum Phase System Using Cascade Structure. In Neural Information Processing 11th International Conference (ICONIP 2004), (2004) 1186–1191
2. Zhang, L. Q., Amari, S., Cichocki, A.: Multichannel Blind Deconvolution of Non-minimum Phase Systems Using Filter Decomposition. IEEE Trans. Signal Processing, **52** (2004) 1430–1442
3. Amari, S.: Natural Gradient Works Efficiently in Learning. Neural Computation, **10** (1998) 251–276
4. Anthony J. Bell, Terrence J. Sejnowski: An Information-maximization Approach to Blind Separation and Blind Deconvolution. Neural Computation, **7** (1995) 1129–1159
5. Cardoso, J., Laheld, B.: Equivariant Adaptive Source Separation. IEEE Trans. on Signal Processing, **44** (1996) 3017–3030
6. Comon, P.: Independent Component Analysis: A New Concept Signal Processing, **36** (1994) 287–314
7. Amari, S., Chen, Tianping, Cichocki, A.: Stability Analysis of Learning Algorithms for Blind Source Separation. Neural Networks, **10** (1997) 1345–1351
8. Zhang, L. Q., Amari, S., Cichocki, A.: Geometrical Structures of FIR Manifold and Multichannel Blind Deconvolution. Journal of VLIS for Signal Processing Systems, **31** (2002) 31–44

Joint Diagonalization of Power Spectral Density Matrices for Blind Source Separation of Convolutive Mixtures

Tiemin Mei[1,2], Jiangtao Xi[2], Fuliang Yin[1], and Joe F. Chicharo[2]

[1] School of Electronic and Information Engineering, Dalian University of Technology
Dalian 116023, China
meitiemin@163.com, flyin@dlut.edu.au
[2] School of Electrical, Computer and Telecommunications Engineering
The University of Wollongong, NSW, Australia
{jiangtao,joe-chicharo}@uow.edu.au

Abstract. A new approach for the blind separation of convolutive mixtures is proposed based on sources nonstationarity and the joint diagonalization of the output power spectral density matrices. It utilizes a time-domain separation network, but the coefficients are optimized based on a frequency domain objective function. The proposed algorithm has the advantages associated with frequency domain BSS algorithms for long mixing channels, but does not suffer from permutation ambiguity problem.

1 Introduction

Convolutive blind source separation (BSS) is a significant issue due to many potential applications, including audio processing, communication systems and biomedical signals processing problem. People have been trying to solve convolutive BSS directly in the time domain [1]. However, such approaches are not very effective in the cases of long mixing channels, as those in the well-known cocktail party problems, where the mixing channels can have 500-2000 taps if modeled by FIR filters.

Frequency domain based approaches are considered as an effective solution for long mixing channel BSS [2-8]. However, the frequency domain approaches suffer from the problems of permutation ambiguity, which makes signal source restoration very difficult, although some measures have been proposed to overcome this problem [2-6][8].

A possible solution for the permutation ambiguity problem is to use objective functions in frequency domain whilst the optimizing parameters are captured in the time domain. Such approaches are proposed in [7] and [11]. However, the work in [11] is for deconvolution of MIMO-FIR systems where many constrains are imposed to the system, whilst the approach in [7] is also based on identification of the mixing system although the goal is to achieve BSS. We have done extensive simulations on the approach in [7] and the performance is not satisfactory.

In this paper we propose a new approach for this challenging issue based on source nonstationarity and the joint diagonalization of output power spectral density matri-

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 520–525, 2005.

ces. The proposed approach is also hybrid solution where the objective function is derived in the frequency domain while optimization parameters are simply the separating filter coefficients. Compared to the work in [7], the objective function is directly for BSS rather than identification. Also we put fewer constrains on the mixing system as compared to [11]. The proposed approach has the advantages associated with the frequency domain approaches but does not have the inherent known permutation disadvantage.

The paper is organized as follows. After briefly describing the problem in Section 2, the new approach is presented in Section 3. Simulations of the proposed algorithm are given in Section 4 and Section 5 concludes the paper.

2 Problem Statement

Throughout the paper we use $(\cdot)^{-1}, (\cdot)^{H}, (\cdot)^{-H}$ to denote inversion, Hermitian transpose, Hermitian transpose and inversion of matrix, respectively.

For analyzing convenience, we consider only the noise-free *N*-by-*N* case, where there are *N* signal sources, *N* observation signals and *N* separated signals. The mixing channels are assumed to be FIR of length *L*, and the separation channels are also FIR with length $M \geq (N-1)(L-1)+1$ [7]. The mixing and separating models are as follows:

$$\boldsymbol{X}(n) = \boldsymbol{A}(n) * \boldsymbol{S}(n) = \sum_{l=0}^{L-1} \boldsymbol{A}(l)\boldsymbol{S}(n-l) \tag{1}$$

$$\boldsymbol{Y}(n) = \boldsymbol{H}(n) * \boldsymbol{X}(n) = \sum_{l=0}^{M-1} \boldsymbol{H}(l)\boldsymbol{X}(n-l) \tag{2}$$

where column vectors $\boldsymbol{S}(n)$, $\boldsymbol{X}(n)$ and $\boldsymbol{Y}(n)$ are the sources, the observations and the output of separating system, respectively, $\boldsymbol{A}(n)$ and $\boldsymbol{H}(n)$ are *N* by *N* FIR filter matrices, which their entries are the impulse responses of the mixing and separating systems, respectively.

The global relationship between the sources and the output of separating system is as follow:

$$\boldsymbol{Y}(n) = \boldsymbol{H}(n) * \boldsymbol{A}(n) * \boldsymbol{S}(n) = \boldsymbol{G}(n) * \boldsymbol{S}(n) \tag{3}$$

where $\boldsymbol{G}(n) = \boldsymbol{H}(n) * \boldsymbol{A}(n)$, is the global impulse response matrix. Its equivalent description of (3) in z-domain is as follows,

$$\mathbf{Y}(z) = \mathbf{G}(z)\mathbf{S}(z) \tag{4}$$

BSS is considered to be successful if the output $\boldsymbol{Y}(n)$ is a permutated and/or filtered version of the signal sources $\boldsymbol{S}(n)$, which implies that the global transfer function $\mathbf{G}(z)$ satisfies the following property,

$$\mathbf{G}(z) = \mathbf{PD}(z) \tag{5}$$

where $\mathbf{P}$ is a permutation matrix, and $\mathbf{D}(z)$ is a diagonal transfer function matrix.

3 The BSS Algorithm

For the separation of convolutive mixtures on the basis of joint diagonalization of power spectral density matrices, we need the following assumptions concerning the sources $\boldsymbol{S}(n)$ and the mixing system $\boldsymbol{A}(n)$:

- Sources are real, zero mean, nonstationary processes and they are uncorrelated to each other.
- The auto-power spectral densities of sources change with time differently.
- The mixing system $\boldsymbol{A}(n)$ is linearly time invariant, and the transfer function of mixing system $\mathbf{A}(z)=\sum_{n=0}^{L-1}\boldsymbol{A}(n)z^{-n}$ is nonsingular on the unit circle of complex plane.

These three assumptions guarantee that the observations are separable at every frequency bin when it is separated in frequency domain [8].

Considering the nonstationarity of sources, the mixing and separating processes can be expressed in frequency domain as follows,

$$\mathbf{X}(n,e^{j\omega})=\mathbf{A}(e^{j\omega})\mathbf{S}(n,e^{j\omega}) \tag{6}$$

$$\mathbf{Y}(n,e^{j\omega})=\mathbf{H}(e^{j\omega})\mathbf{X}(n,e^{j\omega}) \tag{7}$$

where the index n indicates the nonstationarity of sources.

As the sources are uncorrelated to each other, the sources' power spectral density matrix should be diagonal, that is:

$$\mathbf{P}_{SS}(l,\omega)=\left[p_{s_i s_j}(l,\omega)\right]=\mathrm{diag}\left[\mathbf{P}_{SS}(l,\omega)\right] \tag{8}$$

where the operator diag[·] diagonalizes the matrix by setting all the off-diagonal elements to zeros. Note that the index l is used for denoting the nonstationarity of sources.

For a given frequency ω , (6) and (7) are equivalent to the mixing and separating models in instantaneous BSS respectively. According to the three assumptions, the separation of sub-sources $\mathbf{S}(n,e^{j\omega})$ can be achieved by the joint diagonalization of K crosscorrelation matrices of the output of sub-separating system $\mathbf{H}(e^{j\omega})$ at K (big enough) different times. The crosscorrelation matrix of sub-sources $\mathbf{S}(n,e^{j\omega})$ is nothing but the power spectral density matrix of sources at frequency ω . Hence the joint diagonalization condition is:

$$\mathbf{P}_{YY}(l,\omega)=\mathrm{diag}\left[\mathbf{P}_{YY}(l,\omega)\right]\equiv\mathbf{D}_{YY}(l,\omega) \tag{9}$$

for all $l=l_1,l_2,\ldots,l_K$.

Because $\mathbf{P}_{YY}(l,\omega)$s are positive definite, the joint diagonalization of (9) can be converted into the joint optimization of the following objective functions according to Hadamard's Inequality:

$$f(l,\mathbf{H}(e^{j\omega}))=\frac{1}{2}\log\left(\frac{\det\left[\mathbf{D}_{YY}(l,\omega)\right]}{\det\left[\mathbf{P}_{YY}(l,\omega)\right]}\right) \tag{10}$$

for all $l = l_1, l_2, \dots, l_K$, where $\det[\cdot]$: determinant operator.

For all ω , because $f(l, \mathbf{H}(e^{j\omega}))$ is nonnegative, so the joint optimization of (10) is equivalent to the joint optimization of the following integral functions with respect to the time domain parameters of separating system:

$$f(l, \boldsymbol{H}(n)|_{n=0,1,\dots,M-1}) = \frac{1}{2}\int_{-\pi}^{\pi} \log\left(\frac{\det[\mathbf{D}_{\boldsymbol{YY}}(l,\omega)]}{\det[\mathbf{P}_{\boldsymbol{YY}}(l,\omega)]}\right) d\omega \qquad (11)$$

for $l = l_1, l_2, \dots, l_K$.

It is clear from the above that we have effectively avoided the well known permutation problem, because $\boldsymbol{H}(n)$ is obtained directly from the joint optimization of (11).

The gradient is as follows:

$$\frac{\partial f(l, \boldsymbol{H}(n)|_{n=0,1,\dots,M-1})}{\partial \boldsymbol{H}(n)} = \int_{-\pi}^{\pi} \left[\mathbf{D}_{\boldsymbol{YY}}^{-1}(l,\omega)\mathbf{P}_{\boldsymbol{YY}}(l,\omega) - \mathbf{I}\right]\mathbf{H}^{-H}(e^{j\omega})e^{j\omega n} d\omega \qquad (12)$$

where **I** is identity matrix.

We will use the natural gradient for updating the coefficients, which are given as follows [9][10]:

$$\left\{\frac{\partial f(l, \boldsymbol{H}(n)|_{n=0,1,\dots,M-1})}{\partial \boldsymbol{H}(n)}\right\}^{\text{Natural}} = \int_{-\pi}^{\pi} \left[\mathbf{D}_{\boldsymbol{YY}}^{-1}(l,\omega)\mathbf{P}_{\boldsymbol{YY}}(l,\omega) - \mathbf{I}\right]\mathbf{H}(e^{j\omega})e^{j\omega n} d\omega \qquad (13)$$

So we obtained the learning algorithm as follows:

$$\boldsymbol{H}^{l+1}(n) = \boldsymbol{H}^{l}(n) - \mu\int_{-\pi}^{\pi} \left[\mathbf{D}_{\boldsymbol{YY}}^{-1}(l,\omega)\mathbf{P}_{\boldsymbol{YY}}(l,\omega) - \mathbf{I}\right]\mathbf{H}^{l}(e^{j\omega})e^{j\omega n} d\omega \qquad (14)$$

for $l = l_1, l_2, \dots, l_K$.

The algorithm is implemented as follows: The input data is divided into segments and the power spectral densities are estimated using DFT, and then the learning rule (14) is recursively applied on segment-by-segment basis until a good convergence is reached. Note that the power spectral density matrix is estimated recursively by,

$$\mathbf{P}_{\boldsymbol{YY}}(l,\omega_k) = \beta\mathbf{P}_{\boldsymbol{YY}}(l-1,\omega_k) + (1-\beta)\mathbf{Y}(l,e^{j\omega_k})\mathbf{Y}^{H}(l,e^{j\omega_k}) \quad (0 < \beta < 1) \qquad (15)$$

4 Simulation Results

Computer simulation was conducted for real-world recorded mixtures. Experiment is based on two practical test recordings of speech in a room, as provided to the delegates of ICA'99 conference [12]. The convolutive mixtures were recorded with an omni-directional microphone, and the sampling frequency is 16000Hz. We used the first 131072 samples for our simulation. The parameters are as follows: The length of separation filters is 512. FFT block size is 8192. 20 iterations are used. Also parameters are selected based on trial and error where $\beta = 0.3$ and $\mu = 0.01$. The mixtures and the separated sources are shown in Fig. 1, where the mixtures and the separated sources are normalized to the range [-0.5,0.5]. Listening test shows that very good separation has been achieved. Hence we consider that output 1 contains one source (denoted as source 1) and output 2 contains the other (denoted as source 2). As the original sources are unknown, we use the following approach to estimate the SIRs for each of the two outputs: (Refer to Fig. 1)

(1) Find a time interval T_1, during which the waveform of output 1 has a peak and output 2 exhibits low (silent) samples. Denote the segment of samples in outputs 1 and 2 as s_{11} and s_{21} respectively. It is reasonable to believe that s_{11} is the contribution of source 1 only, and that s_{21} is the leakage of source 1 to output 2.

(2) Similarly, we could find a time interval T_2, during which output 2 exhibits a peak s_{22} but output 1 is low (silent) s_{12}. In the same way, s_{22} can be considered as the contribution of source 2 only, and s_{12} the leakage of source 2 to output 1.

(3) The SIRs for outputs 1 and 2 are calculated as $10\log_{10}\frac{p_{s_{11}}}{p_{s_{12}}}$ and $10\log_{10}\frac{p_{s_{22}}}{p_{s_{21}}}$, respectively.

Based on the above approach, SIRs for channels 1 and 2 are measured as: $\text{SIR}_1 = 21.81\,\text{dB}$, $\text{SIR}_2 = 19.97\,\text{dB}$.

5 Conclusions

The proposed BSS algorithm was based on the nonstationarity and joint diagonalization of power spectral density matrices. It can be implemented with DFT in frequency domain. It does not suffer from the permutation issue. It has the advantages of low computation demand and high separation performance.

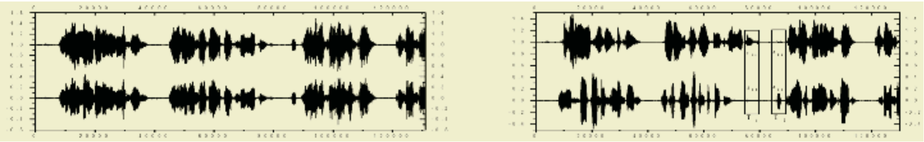


Fig. 1. The real world recorded speech sequences and the corresponding separation results: upper: convolutive mixed speech sequences; lower: the separated speech sequences. Signal segments T_1 and T_2 contain 5000 samples, respectively, which are used for the estimation of signal-to-noise ratios of the separated signals.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant No.60172073 and No.60372082, and the Trans-Century Training Program

Foundation for the Talents by the Ministry of Education of China. This work is also partially supported by Australia Research Council under ARC large Grant No. A00103052.

References

1. Yellin, D., Weinstein, E.: Criteria for Multichannel Signal Separation. IEEE Trans. on SP, **42** (1994) 2156-2168
2. Smaragdis, P.: Blind Separation of Convolved Mixtures in the Frequency Domain. Neurocomputing, **22** (1998) 21-34
3. Parra, L., Spence, C.: Convolutive Blind Separation of Non-stationary Sources. IEEE Trans. on Speech and Audio Processing, **8** (2000) 320-327
4. Araki, S., Makino, S., Nishikawa, T., Saruwatari, H.: Fundamental Limitation of Frequency Domain Blind Source Separation for Convolutive Mixture of Speech. IEEE Trans. on Speech and Audio Process., **11** (2003) 109 – 116
5. Mitianoudis, N. and Davies, M. E.: Audio Source Separation of Convolutive Mixtures. IEEE Trans. on Speech and Audio Processing, **11** (2003) 489-497
6. Ciaramella, A., Tagliaferri, R.: Amplitude and Permutation Indeterminacies in Frequency Domain Convolved ICA. Proc. of the International Joint Conference on Neural Networks, **1** (2003) 708-713
7. Rahbar, K., Reilly, J.: Blind Source Separation of Convolved Sources by Joint Approximate Diagonalization of Cross-spectral Density Matrices. Proc. of ICASSP'01, **5** (2001) 2745-2748
8. Pham, D.-T., Serviére, C., Boumaraf, H.: Blind Separation of Convolutive Audio Mixtures Using Nonstationarity. Proc. of ICA'03. (2003) 981-986
9. Amari, S.: Natural Gradient Works Efficiently in Learning. Neural Computation, **10** (1998) 251-276
10. Sabala, I., Cichocki, A., Amari, S.: Relationships Between Instantaneous Blind Source Separation and Multichannel Blind Deconvolution. Proc. IEEE Conference on Neural Networks, **1** (1998) 39-44
11. Kawamoto, M. and Inouye, Y.: Blind Deconvolution of MIMO-FIR Systems with Colored Inputs Using Second-order Statistics. IEICE Trans. Fundationals, **E86-A** (2003) 597-604
12. See http://www2.ele.tue.nl/ica99/realworld.html

A Block-Adaptive Subspace Method Using Oblique Projections for Blind Separation of Convolutive Mixtures*

Chunyi Peng, Xianda Zhang, and Qutang Cai

National Laboratory of Information Science and Technology
Department of Automation, Tsinghua University
Beijing 100084, China
{pcy98,caiqt98}@mails.tsinghua.edu.cn
zxd-dau@mail.tsinghua.edu.cn

Abstract. This paper presents a block-adaptive subspace algorithm via oblique projection for blind source separation (BSS) problem of convolutive mixtures. In the proposed algorithm, the problem is reformulated into the one of instantaneous mixtures through oblique projections within one block, and then the separation matrix and other model parameters are updated by any static separation approach in a block-adaptive scheme. Compared with other work, the proposed algorithm can obtain lower computational complexity, faster convergence and higher robustness. Simulation results of modulation signals and real speech sources validate the proposed algorithm.

1 Introduction

The BSS problem is to recover or estimate the non-Gaussian independent components from their combined observations, and many efficient algorithms have been proposed to solve this problem of instantaneous mixtures [1][2]. Recently, more attentions have been focused on the case of convolutive mixtures [3]-[6].

The existing BSS approaches for convolutive mixtures can be divided into two groups in frequency domain and time domain. The key of the frequency domain approaches is to reformulate convolutive mixtures in time domain into instantaneous mixtures in frequency domain by Fourier transformation. However, such algorithms would encounter the permutation problem as well as its huge computation for FFT and BSS at each frequency point [3]. The time domain approaches are based on higher-order statistics, e.g., [4], or second-order statistics in subspace methods, e.g., [5], [6]. In those time domain approaches, the optimization of high-dimension cost functions would inevitably cause a high complexity.

This paper proposes a new block-adaptive subspace algorithm without the high-dimension optimization for the BSS of convolutive mixtures. The basic

* This work was supported by the National Natural Science Foundation of China under Grant 60375004.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 526–531, 2005.

idea behind the new method is to reformulate the problem into the instantaneous separation model via oblique projections in one block, and then applies an instantaneous separation algorithm to update the separate matrix and other corresponding parameters for the next block. The new method requires only direct matrix computation, so it can achieve lower complexity and faster convergence.

2 Problem Formulation

The convolutive mixing model with M sources and N mixtures is given by

$$\mathbf{x}(t) = \sum_{\tau=0}^{\text{Ł}} \mathbf{A}(\tau)\mathbf{s}(t-\tau) \tag{1}$$

where $\mathbf{x}(t)$ is an $N \times 1$ observed signal vector, $\mathbf{s}(t)$ is an $M \times 1$ source vector, and $\mathbf{A}(\tau)$ represents the FIR channel matrix with the maximum length of $L+1$.

Let $\mathbf{X}_t = [\mathbf{x}(t), \cdots, \mathbf{x}(t+j-1)]$ and $\mathbf{X}_{t|t+i-1} = [\mathbf{X}_t^T, \cdots, \mathbf{X}_{t+i-1}^T]^T$ denote the observation matrix and block representation at time t, respectively, and then (1) can be rewritten as

$$\mathbf{X}_{t|t+i-1} = \Gamma_i(\mathbf{A})\mathbf{S}_{t-L|t+i-1} \tag{2}$$

where $\Gamma_i(\mathbf{A})$ is a Toeplitz matrix

$$\Gamma_i(\mathbf{A}) = \begin{bmatrix} \mathbf{A}(L) \cdots \; \mathbf{A}(0) & & \\ & \ddots \quad \cdots \quad \ddots & \\ & & \mathbf{A}(L) \cdots \; \mathbf{A}(0) \end{bmatrix}_{Ni \times M(i+L)} .$$

For (2) to be solvable, we assume that

A1. The sources $s_m(t), m = 1, \cdots, M$ are mutually independent.

A2. The source sequence $s_m(t)$ has the linear complexity greater than $2i+2L$.

A3. $\Gamma_i(\mathbf{A})$ has full column rank.

Assumption A1 is fundamental in all BSS problems [2], and *Assumptions A2* and *A3* indicate an isomorphic relation between the output and input subspaces [8], namely, if some $\Gamma_{i_0}(\mathbf{A})$ has full column rank, then we have

$$\mathrm{R}\{\mathbf{X}_{t|t+i-1}\} \simeq \mathrm{R}\{\mathbf{S}_{t-L|t+i-1}\}, \quad \forall i > i_0 \tag{3}$$

where $\mathrm{R}\{\mathbf{X}_{t|t+i-1}\}$ and $\mathrm{R}\{\mathbf{S}_{t-L|t+i-1}\}$ denote the row subspaces spanned by the data block $\mathbf{X}_{t|t+i-1}$ and $\mathbf{S}_{t-L|t+i-1}$, respectively. *Assumption A3* also implies that $N > M$ (more sensors than sources).

3 Block-Adaptive Algorithm Using Oblique Projections

Consider two row subspaces $P = \mathrm{R}\{\mathbf{P}\}$ and $Q = \mathrm{R}\{\mathbf{Q}\}$, then the oblique projector along Q onto P is given by [7]

$$\mathbf{E}_{P|Q} = \begin{bmatrix} \mathbf{P}^H & \mathbf{Q}^H \end{bmatrix} \begin{bmatrix} \mathbf{P}\mathbf{P}^H & \mathbf{P}\mathbf{Q}^H \\ \mathbf{Q}\mathbf{P}^H & \mathbf{Q}\mathbf{Q}^H \end{bmatrix}^{\dagger} \begin{bmatrix} \mathbf{P} \\ \mathbf{0} \end{bmatrix} \tag{4}$$

where $\mathbf{B}^{\dagger}$ is the pseudo-inverse of $\mathbf{B}$. Obviously, the projector $\mathbf{E}_{P|Q}$ satisfies

$$\mathbf{P}\mathbf{E}_{P|Q} = \mathbf{P}, \mathbf{Q}\mathbf{E}_{P|Q} = \mathbf{0}, \tag{5}$$

namely, P and Q are the range and null spaces of the projector $\mathbf{E}_{P|Q}$.

We notice that oblique projectors should keep invariant in two isomorphic subspaces, as the following proposition:

Proposition 1. *If the matrices* $\mathbf{T}_1$ *and* $\mathbf{T}_2$ *have full column rank, then*

$$\mathbf{E}_{P|Q} = \mathbf{E}_{\mathrm{R}\{\mathbf{T}_1\mathbf{P}\}|\mathrm{R}\{\mathbf{T}_2\mathbf{Q}\}} \tag{6}$$

The proof can be easily derived from the oblique projector definition and found in [9]. By *Proposition 1* and (3), oblique projectors in source subspaces can be constructed from the those in observation subspaces.

Consider the BSS problem of convolutive mixtures (2). The algorithm aims to find some operator $\mathbf{E}$ to obtain instantaneous mixtures as follows

$$\mathbf{X}_{t|t+L}\mathbf{E} = \mathbf{F}(\mathbf{A})\mathbf{s}_t \tag{7}$$

where $\mathbf{F}(\mathbf{A})$ are composed of channel response parameters. In our multiple input multiple output (MIMO) system, construct the *present*, *past* and *future* observation subspaces and their corresponding isomorphic source subspaces as

$$\begin{aligned}
X_{\mathrm{pr}} &= \mathrm{R}\{\mathbf{X}_{t|t+L}\} \simeq \mathrm{R}\{\mathbf{S}_{t-L|t+L}\} = S_{\mathrm{pr}}, \\
X_{\mathrm{pa1}} &= \mathrm{R}\{\mathbf{X}_{t-i|t-1}\} \simeq \mathrm{R}\{\mathbf{S}_{t-i-L|t-1}\} = S_{\mathrm{pa1}}, \\
X_{\mathrm{pa2}} &= \mathrm{R}\{\mathbf{X}_{t-i|t}\} \simeq \mathrm{R}\{\mathbf{S}_{t-i-L|t}\} = S_{\mathrm{pa2}}, \\
X_{\mathrm{fu1}} &= \mathrm{R}\{\mathbf{X}_{t+L+1|t+L+i}\} \simeq \mathrm{R}\{\mathbf{S}_{t+1|t+L+i}\} = S_{\mathrm{fu1}}, \\
X_{\mathrm{fu2}} &= \mathrm{R}\{\mathbf{X}_{t+L|t+L+i}\} \simeq \mathrm{R}\{\mathbf{S}_{t|t+L+i}\} = S_{\mathrm{fu2}}.
\end{aligned}$$

Substituting $\mathbf{E}_{X_{\mathrm{fu2}}|X_{\mathrm{pa1}}} = \mathbf{E}_{S_{\mathrm{fu2}}|S_{\mathrm{pa1}}}$ and using (5), the projection of $\mathbf{S}_{t-L|t+L}$ by $\mathbf{E}_{X_{\mathrm{fu2}}|X_{\mathrm{pa1}}}$ can be simplified as

$$\mathbf{S}_{t-L|t+L}\mathbf{E}_{X_{\mathrm{fu2}}|X_{\mathrm{pa1}}} = \mathbf{S}_{t-L|t+L}\mathbf{E}_{\mathrm{R}\{\mathbf{S}_{t|t+L+i}\}|\mathrm{R}\{\mathbf{S}_{t-i-L|t-1}\}} = \left[\mathbf{0}^T, \mathbf{S}_{t|t+L}^T\right]^T, \tag{8}$$

Similarly,

$$\mathbf{S}_{t|t+L}\mathbf{E}_{X_{\mathrm{pa2}}|X_{\mathrm{fu1}}} = \left[\mathbf{s}_t^T, \mathbf{0}^T\right]^T, \tag{9}$$

then the joint projection of $\mathbf{X}_{t|t+L}$ by $\mathbf{E}_1 = \mathbf{E}_{X_{\mathrm{fu2}}|X_{\mathrm{pa1}}}\mathbf{E}_{X_{\mathrm{pa2}}|X_{\mathrm{fu1}}}$ is reduced to

$$\begin{aligned}
\mathbf{X}_{t|t+L}\mathbf{E}_1 &= \mathit{\Gamma}_{L+1}(\mathbf{A})\mathbf{S}_{t-L|t+L}\mathbf{E}_{X_{\mathrm{fu2}}|X_{\mathrm{pa1}}}\mathbf{E}_{X_{\mathrm{pa2}}|X_{\mathrm{fu1}}} \\
&= \mathit{\Gamma}_{L+1}(\mathbf{A})\left[\mathbf{0}^T, \mathbf{s}_t^T, \mathbf{0}^T\right]^T \\
&= \mathit{\Gamma}_{L+1}(\mathbf{A})\mathbf{H}\mathbf{s}_t
\end{aligned} \tag{10}$$

where $\mathbf{H} = [\mathbf{0}_{LM\times(LM+M)}^T, \mathbf{I}_{M\times M}^T, \mathbf{0}_{LM\times(LM+M)}^T]^T$ is used to select the middle M columns of $\mathit{\Gamma}_{L+1}(\mathbf{A})$.

Though the instantaneous mixture model (10) is obtained in a similar way to Yu's work on channel estimation in a single input-multiple output (SIMO) system [8], their differences lie in: $\mathbf{s}_t$ is a row vector and the rank of $\mathbf{X}_{t|t+L}\mathbf{E}_1$ equals one in Yu's work; while $\mathbf{s}_t$ is a $M \times j$ matrix and the rank of $\mathbf{X}_{t|t+L}\mathbf{E}_1$ equals M in this model, so the SVD of (10) yields instantaneous mixtures

$$\mathbf{X}_{t|t+L}\mathbf{E}_1 = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H, \tag{11}$$

$$\mathbf{A}_{ins}\mathbf{s}_t = \mathbf{\Sigma}(1:M,1:M)\mathbf{V}(:,1:M)^H; \tag{12}$$

furthermore, $\mathbf{A}_{ins}$ inevitably varies with different blocks for the uncertainty in a MIMO model. For convenience, the symbols with the upper sign $^{(k)}$, e.g. $\mathbf{U}^{(k)}, \mathbf{\Sigma}^{(k)}, \mathbf{V}^{(k)}$ and $\mathbf{A}_{ins}^{(k)}$, represent the corresponding components for the kth block.

Using (7), (11) and (12), we have

$$\mathbf{U}^{(k)}(:,1:M)\mathbf{A}_{ins}^{(k)} = \mathbf{F}(\mathbf{A}) = \mathbf{const}. \tag{13}$$

Based on (13), a block-adaptive scheme is developed to keep the same instantaneous mixtures model for each block in the below. Define the transform matrix $\mathbf{T}^{(k)}$ and instantaneous mixtures $\mathbf{Z}^{(k)}$ for the kth block as

$$\mathbf{T}^{(k)} = \mathbf{U}^{(1)}(:,1:M)^{\dagger}\mathbf{U}^{(k)}(:,1:M), \tag{14}$$

$$\mathbf{Z}^{(k)} = \mathbf{T}^{(k)}\mathbf{\Sigma}^{(k)}(1:M,1:M)\mathbf{V}^{(k)}(:,1:M)^H, \tag{15}$$

and simplify $\mathbf{Z}^{(k)}$ with (12) - (15)

$$\mathbf{Z}^{(k)} = \mathbf{U}^{(1)}(:,1:M)^{\dagger}\mathbf{U}^{(k)}(:,1:M)\mathbf{A}_{ins}^{(k)}\mathbf{s}_k = \mathbf{A}_{ins}^{(1)}\mathbf{s}_k \tag{16}$$

So the initial value of separation matrix $\mathbf{W}_{1,ini}^{(k)}$ for the kth block is given by the final value of separation matrix $\mathbf{W}_{1,fin}^{(k-1)}$ for the last block, i.e.,

$$\mathbf{W}_{1,ini}^{(k)} = \mathbf{W}_{fin}^{(k-1)} \tag{17}$$

Finally, the separated matrix $\mathbf{W}^{(k)}$ can be computed by any static BSS algorithm in each block.

The proposed algorithm can be seen as a sliding window method with the window size j. To make full use of each block, the well-known JADE in [1] is modified with the following update of higher-order cumulant $\mathbf{c}^{(k)}$ for the kth block

$$\mathbf{c}^{(k)} = \alpha\mathbf{c}^{(k-1)} + (1-\alpha)\mathbf{c}_k \tag{18}$$

where $\mathbf{c}_k$ denotes the higher-order cumulant calculated only from the kth block, and α acts as a forgetting factor. This modified JADE algorithm, called Block-JADE, can deal with a small number of block data.

Compared with other subspace algorithms, e.g. in [5][6], our algorithm has the following advantages: firstly, it reduces a high computational burden by

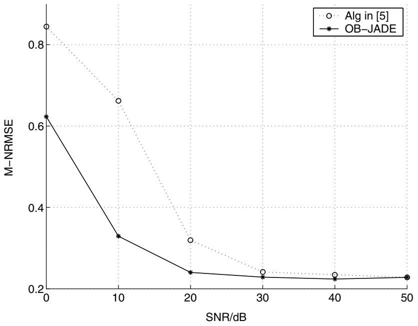


Fig. 1. M-NRMSE comparison versus different SNRs over 500 runs, where two modulation sources are given by $s_1(k) = sin(6\pi k/1000)sin(4\pi k/25)$ and $s_2(k) = sin(\pi k/50 + sin(\pi k/500))$

direct oblique projections avoiding high-dimension optimizations in their work; secondly, the proposed approach is block-adaptive and only requires a smaller number of data each time, while others are all batch algorithms; finally, the SVD in the proposed algorithm benefits the noise elimination and good performance at low SNR.

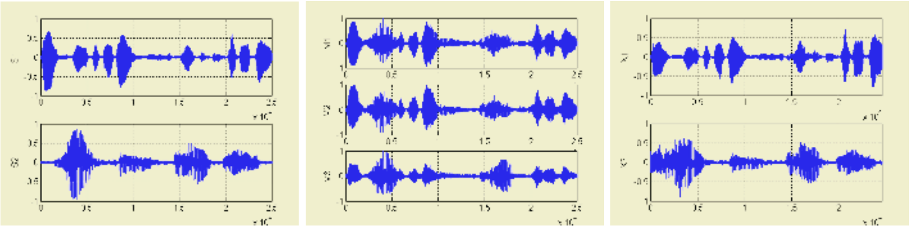
4 Experimental Results

To illustrate the performance of the proposed oblique projection algorithm, some simulations are presented in this section. The channel matrix is randomly generated, with the dimension $N \times M = 3 \times 2$ and the degree is $L = 4$. The data block parameters are chosen to be $i = 4, j = 200$, Block-JADE with $\alpha = (k-1)/k$ is taken as an instantaneous separation approach in our algorithm, called OB-JADE for short. For comparison, the mutually referenced algorithm in [5] is applied at the same time. To access the blind separation capability, multi-source normalized root mean square error (M-NRMSE) is given by:

$$\text{M-NRMSE} = \frac{1}{M}\sum_{i=1}^{M}\sqrt{\frac{1}{N_k \left\|s_i\right\|^2}\sum_{k=1}^{N_k}\left\|\hat{s}_{i(k)} - s_i\right\|^2} \qquad (19)$$

where N_k is the number of independent runs, and $\hat{s}_{i(k)}$ is the ith separated signal sequence in the kth run.

Many simulations of different sources show that the proposed algorithm can efficiently reconstruct sources. Though both can achieve the separation of stationary signals, our algorithm can obtain a lower computation and better performance at low SNR, as shown in Fig. 1. Moreover, the proposed algorithm can obtain satisfactory results for non-stationary signals such as speech signals, while the algorithm in [5] can not. Fig. 2 shows the experiment results of two real speech signals by OB-JADE. It is observed that the speech signals are successfully reconstructed in about 10 blocks (2000 symbols).



(a) (b) (c)

Fig. 2. Blind separation of convolutive speech signals. (a) Speech sources. (b) Convolutive mixtures. (c) Separated signals

5 Conclusion

In this paper, we present a new subspace adaptive algorithm using oblique projections for blind separation of convolutive mixtures. The main idea behind this algorithm is the deconvolution via oblique projections and block-adaptive scheme on the relation between the instantaneous model and real channel parameters. Compared with other existing subspace methods, the proposed approach is a block-adaptive algorithm, with a lower computational cost, faster convergence and noise robustness.

References

1. Cardoso, J., Souloumiac,A.: Blind Beamforming for Non Gaussian Signals.IEE-Proceedings-F, **140** (1993) 362-370
2. Hyvarinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. Wiley, New York (2001)
3. Araki, S., Mukai, R., Makino, S., et al: The Fundamental Limitation of Frequency Domain Blind Source Separation for Convolutive Mixtures of Speech. IEEE Trans. Speech and Audio Processing, **21** (2003) 109-116
4. Amari, S., Douglas, S.C., Cichock, A., et al: Multichannel Blind Deconvolution and Equalization Using the Natural Gradient. Proc. IEEE Workshop on Signal Processing Advance in Wireless Communications, Paris, France (1997) 101-104
5. Mansour A.: A Mutually Referenced Blind Multiuser Separation of Convolutive Mixture Algorithm. Signal Processing, **81** (2001) 2253-2266
6. Hua,Y.B., An, S.J., Xiang, Y.: Blind Identification of FIR MIMO Channels by Decorrelating Subchannels. IEEE Trans. Signal Processing, **51** (2003) 1143-1155
7. Behrens, R.T., Scharf, L.L.: Signal Processing Applications of Oblique Projection Operators. IEEE Trans. Signal Processing, **42** (1994) 1413-1424
8. Yu, X., Tong, L.: Joint Channel and Symbol Estimation by Oblique Projections. IEEE Trans. Signal Processing, **49** (2001) 3074-308
9. Peng, C.Y.: Research on Oblique Projectors and Their Applications in Multi-user Detection and Blind Source Separation. Master Thesis, Tsinghua University (2005)

FIR Convolutive BSS Based on Sparse Representation

Zhaoshui He, Shengli Xie, and Yuli Fu

School of Electronics & Information Engineering, South China University of Technology
Guangzhou 510640, China
{he_shui,y.l.fu@}@163.com, adshlxie@scut.edu.cn

Abstract. Based on sparse representation, this paper discusses convolutive BSS of sparse sources and presents a FIR convolutive BSS algorithm that works in the frequency domain. This algorithm does not require that source signals be i.i.d or stationary, but require that source signals be sufficiently sparse in frequency domain. Furthermore, our algorithm can overcome *permutation problem* of frequency convolutive BSS method. For short-order FIR convolution, simulation shows good performance of our algorithm.

1 Introduction

It is well known that blind signal separation (BSS) technique allows us to separate the source signals only from the observed mixtures. In many cases, such as in wireless telecommunication, image processing, biomedical signal processing, the mixtures are the convolutions of the source signals. Deconvolution is popularly exploited in these practical fields to retrieve the original signals. Thus, blind deconvolution has gained much attractive attention [3], [6], [8], [11], [12].

Recently, many kind of blind deconvolution algorithms have been proposed [4], [5], [6], [7]. These algorithms can be classified as time domain algorithms and frequency domain algorithms. The time domain algorithms generally are based on higher cumulant or information theory, and usually, the iid of the source signals is assumed. However, the iid assumption may be not easy satisfied in practice. Lately, some frequency domain algorithms were developed to deal with deconvolution problems [1], [11], [12]. By the frequency domain algorithms, iid assumption is not necessary. However, as commented in the paper [11], these algorithms do their work at every independent frequency bin. This causes some ambiguity of permutation, frequency coupling, and limited effect. In this paper, we present a frequency deconvolution algorithm, which overcomes permutation problem.

The convolutive mixing model with *n*-source and *n*-sensor can be described as :

$$\boldsymbol{x}(\mathrm{t})=\sum_{\tau=0}^{\mathrm{q}}\boldsymbol{A}(\tau)\boldsymbol{s}(\mathrm{t}-\tau)+\boldsymbol{n}(\mathrm{t}),\quad \mathrm{t}=1,2,\cdots,\mathrm{T} \tag{1}$$

where $\boldsymbol{x}(t)=(x_1(t),\cdots,x_n(t))^T\in R^{n\times1}$ is the vector of observed signals, $\boldsymbol{s}(\mathrm{t})=(\mathrm{s}_1(\mathrm{t}),\cdots,\mathrm{s}_\mathrm{n}(\mathrm{t}))^\mathrm{T}\in\mathrm{R}^{\mathrm{n}\times1}$ is the vector of sources, $A(\tau)\in R^{n\times n},\tau=0,\cdots,q$ is the channel matrix with maximum element order q, and $\boldsymbol{n}(\mathrm{t})=(\mathrm{n}_1(\mathrm{t}),\cdots,\mathrm{n}_\mathrm{n}(\mathrm{t}))^\mathrm{T}\in\mathrm{R}^{\mathrm{n}\times1}$ is the additive

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 532–537, 2005.

noise vector. Our aim is to estimate the $A(\tau)\in R^{n\times n},\tau=0,\cdots,q$ up to a scaling and permutation factor, we are interested in finding $\hat{A}(\tau)\in R^{m\times n},\tau=0,\cdots,q$ satisfying

$$\hat{A}(\tau)=A(\tau)\boldsymbol{PD},\ \tau=0,\cdots,q \tag{2}$$

where $\boldsymbol{D}\in \mathrm{R}^{\mathrm{n}\times \mathrm{n}}$ and $\boldsymbol{P}\in \mathrm{R}^{\mathrm{n}\times \mathrm{n}}$ are, respectively, constant diagonal and permutation matrices. If we succeed in finding $\hat{A}(\tau)\in R^{n\times n},\tau=0,\cdots,q$, we can estimate $\hat{s}(\omega)$ in some conditions.

The source signals are assumed to be very *sparse* in frequency domain. It means that they are zeros or nearly zeros most of the frequency bin. Fortunately, many signals show good *sparse* property in frequency domain. For example, many audio signals, especially, music signals.

2 Instantaneous Undetermined BSS Model

Recently, *sparse representation* had succeeded in instantaneous BSS. The ability to directly estimate the mixing matrix by clustering in some domain in which the sources become extremely sparse was exploited by Zibulevsky et al [10], which exhibited the first practical blind algorithm for the case that sensors are less than sources. Sequentially, some undetermined BSS algorithms were proposed. For example, Zibulevsky and Pearlmutter separated the source signals by sparse decomposition [10]; Lee, Lewicki and Li performed BSS by overcomplete representation [2], [13], [8]; A two-step approach was proposed for undetermined BSS by Bofill, in which the mixing matrix and the sources are estimated separately [9]. However, there are no successful convolutive BSS algorithm using sparse representation, and this paper just does this.

Consider the following instantaneous BSS model,

$$\boldsymbol{x}(t)=\boldsymbol{As}(t)\quad t=1,\cdots,T \tag{3}$$

where $\boldsymbol{x}(t)=(x_1(t),\cdots,x_m(t))^T\in R^{m\times 1}$ is the vector of observed signals, $\boldsymbol{s}(\mathrm{t})=(\mathrm{s}_1(\mathrm{t}),\cdots,\mathrm{s}_\mathrm{n}(\mathrm{t}))^\mathrm{T}\in \mathrm{R}^{\mathrm{n}\times 1}$ is the vector of sources, $\boldsymbol{A}=[\boldsymbol{a}_1,\cdots,\boldsymbol{a}_\mathrm{n}]\in \mathrm{R}^{\mathrm{m}\times \mathrm{n}}$ is *m* by *n* mixing matrix. When $\mathrm{m}<\mathrm{n}$, sensors are less than sources and we call it undetermined BSS. The objective of sparse representation is to estimate the mixing matrix $\boldsymbol{A}$ such that sources $s(t)$ corresponding to $\boldsymbol{A}$ are as sparse as possible, and it means many entries of sources $\boldsymbol{s}(t)$ are zeros or nearly zero most of the time. Generally, estimating mixing matrix $\boldsymbol{A}$ in (3) sums up to solve the following optimal problem [8], [9], [10],

$$\begin{cases}\min\limits_{A,s(t)} J=\min\limits_{A,s(t)}\dfrac{1}{T}\sum\limits_{t=1}^{T}J\left(s(t)\right)=\min\limits_{A,s(t)}\dfrac{1}{T}\sum\limits_{t=1}^{T}\sum\limits_{i=1}^{n}\left|s_i(t)\right| \\ \boldsymbol{x}(t)=\boldsymbol{As}(t)\quad t=1,\cdots,T\end{cases} \tag{4}$$

Usually, we can constrain that all column vectors of $\boldsymbol{A}$ are unit vectors with 2-norms being equal to [8], [14][1]. For convenience, we note t to be sample point in work domain without loss of generality.

[1] Note that ω is frequency independent quantity and t is time independent quantity in this paper.

With respect to model (4), we have the following result without proof, and its proof is included in reference [15].

Theorem. For arbitrary fixed sample point t, as long as nonzero entries of source vector $\boldsymbol{s}(t)$ are less than sensors, the natural gradient of $\boldsymbol{A}$ is

$$\Delta \boldsymbol{A} = \boldsymbol{A}\boldsymbol{A}^T \frac{\partial J}{\partial \boldsymbol{A}} = -\boldsymbol{A}\frac{1}{T}\sum_{t=1}^{T}\left(\left(sign\left(\boldsymbol{s}(t)\right)\right)\cdot \boldsymbol{s}^T(t)\right) \tag{5}$$

Therefore, we get updating formula of mixing matrix $\boldsymbol{A}$:

$$\boldsymbol{A}^{(k+1)} = \boldsymbol{A}^{(k)} - \mu\Delta\boldsymbol{A}^{(k)} = \boldsymbol{A}^{(k)} + \mu\boldsymbol{A}^{(k)}\sum_{t=1}^{T}\left(\phi(\boldsymbol{s}(t))\cdot(\boldsymbol{s}(t))^{T}\right) \tag{6}$$

To constrain 2-norms of all column vectors of $\boldsymbol{A}$ to be 1, we do the following *normalized operation* during iterative procedure:

$$a_{ri}^{(k)} = a_{ri}^{(k)} \Big/ \sqrt{\sum_{j=1}^{m}\left(a_{ji}^{(k)}\right)^2}\,,\ i = 1,\cdots,n; r = 1,\cdots,m \tag{7}$$

where $\boldsymbol{A}^{(k)} = \left(a_{ri}^{(k)}\right)_{m\times n}$. Next, we extend above instantaneous BSS to convolutive BSS.

3 Convolutive BSS Algorithm Based on Sparse Representation

In noise-free, being taken Fourier transform, expression (1) is changed into

$$\boldsymbol{x}(\omega) = \tilde{\boldsymbol{A}}(\omega)\boldsymbol{s}(\omega) = \boldsymbol{A}(0)\boldsymbol{s}(\omega) + \cdots + \boldsymbol{A}(q)\boldsymbol{s}(\omega)e^{-qj\frac{2\pi}{T}\omega} = \bar{\boldsymbol{A}}\cdot\boldsymbol{es}(\omega),\quad \omega = 1,\cdots,T \tag{8}$$

where $\bar{\boldsymbol{A}} = \left[\boldsymbol{A}(0),\boldsymbol{A}(1),\cdots,\boldsymbol{A}(q)\right], \tilde{\boldsymbol{A}}(\omega) = \boldsymbol{A}(0) + \boldsymbol{A}(1)e^{-j\frac{2\pi}{T}\omega} + \cdots + \boldsymbol{A}(q)e^{-qj\frac{2\pi}{T}\omega}$

$$\boldsymbol{es}(\omega) = \left(\boldsymbol{s}^T(\omega), e^{-j\frac{2\pi}{T}\omega}\boldsymbol{s}^T(\omega),\cdots,e^{-qj\frac{2\pi}{T}\omega}\boldsymbol{s}^T(\omega)\right)^T \in C^{(q+1)n\times 1} \tag{9}$$

Now we can estimate $\boldsymbol{A}(\tau), \tau = 0,\cdots,q$ by estimating $\bar{\boldsymbol{A}}$. Equation (8) can be written in compact form:

$$\boldsymbol{x}(\omega) = \bar{\boldsymbol{A}}\cdot\boldsymbol{es}(\omega), \omega = 1,\cdots,T \tag{10}$$

We know that $\bar{\boldsymbol{A}}$ is a real matrix. Taking real part of two sides of (10), we obtain

$$\boldsymbol{rx}(\omega) = \mathrm{Re}\left(\boldsymbol{x}(\omega)\right) = \bar{\boldsymbol{A}}\cdot\mathrm{Re}\left(\boldsymbol{es}(\omega)\right) = \bar{\boldsymbol{A}}\cdot\boldsymbol{res}(\omega), \omega = 1,\cdots,T$$

i.e.

$$\boldsymbol{rx}(\omega) = \bar{\boldsymbol{A}}\cdot\boldsymbol{res}(\omega), \omega = 1,\cdots,T \tag{11a}$$

By analogy, Taking image part of two sides of (10), we obtain

$$\boldsymbol{ix}(\omega) = \bar{\boldsymbol{A}}\cdot\boldsymbol{ies}(\omega), \omega = 1,\cdots,T \tag{11b}$$

where $\boldsymbol{ix}(\omega)$, $\boldsymbol{ies}(\omega)$ are, respectively, the image part of $\boldsymbol{x}(\omega)$ and $\boldsymbol{es}(\omega)$.

We assume that source signals $\boldsymbol{s}(t)$ are sparse enough in frequency domain, so many entries of $\boldsymbol{s}(\omega), \omega = 1,\cdots,T$ are zeros or nearly zeros in most of the frequency bin. Therefore, many entries of $\boldsymbol{res}(\omega)$ and $\boldsymbol{ies}(\omega)$ are zeros or nearly zeros. Comparing (11a) and (11b) with (3), we have the natural gradient of $\bar{A}$ as follow

$$\Delta\bar{A} = -\bar{A}\frac{1}{2T}\left[\sum_{t=1}^{T}\left(\left(sign\left(\boldsymbol{res}(t)\right)\right)\cdot\boldsymbol{res}^T(t)\right) + \sum_{t=1}^{T}\left(\left(sign\left(\boldsymbol{ies}(t)\right)\right)\cdot\boldsymbol{ies}^T(t)\right)\right]$$

Now we get the learning formula of $\bar{A}$:

$$\bar{A}^{(k+1)} = \bar{A}^{(k)} - \mu\Delta\bar{A}^{(k)} = \bar{A}^{(k)} + \mu\bar{A}^{(k)}\left[\sum_{t=1}^{T}\left(\left(sign\left(\boldsymbol{res}(t)\right)\right)\cdot\boldsymbol{res}^T(t)\right) + \sum_{t=1}^{T}\left(\left(sign\left(\boldsymbol{ies}(t)\right)\right)\cdot\boldsymbol{ies}^T(t)\right)\right] \tag{12}$$

For convolutive BSS, we can modify *normalized operation* （7） into

$$a_{ri}^{(k)}(\tau) = (q+1)a_{ri}^{(k)}(\tau) \Big/ \sqrt{\sum_{j=1}^{m}\left(a_{ji}^{(k)}(0)\right)^2 + \cdots + \sum_{j=1}^{m}\left(a_{ji}^{(k)}(q)\right)^2} \tag{13}$$

where $i = 1,\cdots,n; r = 1,\cdots,m; \tau = 0,\cdots,q$ .After being normalized, $A^{(k)}(\tau), \tau = 0,\cdots,q$ such that

$$\sum_{j=1}^{m}\left(a_{ji}^{(k)}(0)\right)^2 + \cdots + \sum_{j=1}^{m}\left(a_{ji}^{(k)}(q)\right)^2 = q+1,\ i = 1,\cdots,n \tag{14}$$

Normalized operation （13） is consistent with constraint (2), and make algorithm more robust. Furthermore, (8) can be denoted as

$$\boldsymbol{x}(\omega) = \tilde{A}(\omega)\boldsymbol{s}(\omega),\quad \omega = 1,\cdots,T \tag{15}$$

It is obvious that $\tilde{A}(\omega) = DFT\left(A(\tau)\right) = A(0) + A(1)e^{-j\frac{2\pi}{T}\omega} + \cdots + A(q)e^{-qj\frac{2\pi}{T}\omega} \in C^{n\times n}$. We can estimate $\boldsymbol{s}(\omega)$ as follow:

$$\boldsymbol{s}(\omega) = \left(\tilde{A}(\omega)\right)^{-1}\boldsymbol{x}(\omega),\quad \omega = 1,\cdots,T \tag{16}$$

From above discussion, we summarize *our convolutive BSS algorithm* as follow:

1) Taking discrete Fourier transform of mixtures $\boldsymbol{x}(t)$, we obtain $\boldsymbol{x}(\omega)$;
2) Initialize $\bar{A}$ to be $\bar{A}^{(0)}$ by random, and let $k = 0$, set step-size μ;
3) From(9),compute $A^{(k)}(\tau), \tau = 0,\cdots,q$ by $\bar{A}^{(k)}$, then normalize $A^{(k)}(\tau), \tau = 0,\cdots,q$ by （13）;
4) Compute $\tilde{A}^{(k)}(\omega), \omega = 1,\cdots,T$ corresponding to $\bar{A}^{(k)}$ by (9);
5) Compute $\boldsymbol{s}^{(k)}(\omega)$ by （16）;
6) Compute $\boldsymbol{es}^{(k)}(\omega)$ from $\boldsymbol{s}^{(k)}(\omega)$ by(9), and we obtain $\boldsymbol{res}^{(k)}(\omega)$ and $\boldsymbol{ies}^{(k)}(\omega)$ by taking real-part and image-part of $\boldsymbol{es}^{(k)}(\omega)$;
7) Substitute $\boldsymbol{res}^{(k)}(\omega)$ and $\boldsymbol{ies}^{(k)}(\omega)$ into (12), and we have $\bar{A}^{(k+1)}$, then let $k = k+1$;
8) If iteration converges, goto 9), else goto 3);
9) Output $\boldsymbol{s}^*(\omega) = \boldsymbol{s}^{(k)}(\omega), \omega = 1,\cdots,T$, we get separated signal $\hat{\boldsymbol{s}}(t), t = 1,\cdots,T$ by taking inverse discrete Fourier transform of $\boldsymbol{s}^*(\omega)$.

4 Simulation

To demonstrate the performance of our algorithm, the correlative coefficient and signal noise ratio (SNR) [2] are employed to check the performance of the deconvolution algorithm. On the other hand, we compare the performance of our algorithm with CONVBSS algorithm [1].

Example: three sensors, three sources convolutive BSS
The sources(65536 samples) come from experiment FourVoices of reference [9], and they are No.2, No.3, No.4 of four sources, where two women voice and a man voice. The mixing channel A is

$$A(z)=\begin{pmatrix} 0.9000 & 0.1000 & 0.7000 \\ 0.2000 & 0.9000 & -0.5000 \\ 0.4000 & 0.3000 & 0.8000 \end{pmatrix}+\begin{pmatrix} 0.6000 & 0.2000 & 0.4000 \\ 0.7000 & 0.5000 & 0.5000 \\ -0.4000 & 0.3000 & 0.1000 \end{pmatrix}z^{-1}$$

We initiate mixing channel A by *MATLAB command* randn('state',5); $A^{(0)}$=randn(3,3,2). Set $\mu = 0.01$. After 22 iterations, algorithm is converge.

Table 1. Simulation results

	SNR, dB			Correlative coefficient matrix
Sources	1	2	3	
CONVBSS	18.7533	19.4306	12.0908	$\begin{pmatrix} 0.9410 & 0.0366 & -0.0019 \\ -0.0288 & 0.9441 & 0.1272 \\ 0.0092 & -0.1073 & 0.9381 \end{pmatrix}$
Our algorithm	26.7115	29.0835	25.7853	$\begin{pmatrix} 0.9663 & -0.1001 & 0.1236 \\ -0.1022 & -0.9744 & -0.0788 \\ -0.1050 & -0.1225 & 0.9533 \end{pmatrix}$

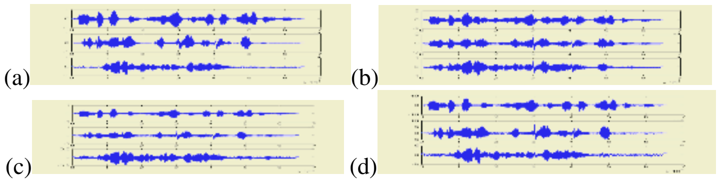


Fig. 1. (a) Sources; (b) Mixtures; (c) Separated signals by CONVBSS; (d) Separated signals by our algorithm

By comparison, it is obvious to see that our algorithm is more accurate than CONVBSS, and CONVBSS is a little faster than it.

In conclusion, this paper discusses convolutive BSS of sparse sources and presents a new deconvolution algorithm. Our algorithm does not require that source signals be stationary or i.i.d, but require that source signals be sufficiently sparse in frequency domain. However, the convergence of our algorithm is not very fast. In above simulation, it took more than 7 minutes to complete convolutive BSS on Intel Celerom 1.7G PC, so it is very difficult for it to work in real-time environment. Additionally, when the order q of FIR mixing channel $A(\tau), \tau = 0,\cdots,q$ is very high, too many parameters are required to be optimized, and the performance of BSS becomes poor. In the future, we're going to improve our algorithm.

Acknowledgements

The work is supported by the Guang Dong Province Science Foundation for Program of Research Team (grant 04205783), the National Natural Science Foundation of China (Grant 60274006), the Natural Science Key Fund of Guang Dong Province, China (Grant 020826), the National Natural Science Foundation of China for Excellent Youth (Grant 60325310) and the Trans—Century Training Program, the Foundation for the Talents by the State Education Commission of China, and the SRF for ROCS, SEM.

References

1. Parra, L., Spence, C.: Convolutive Blind Separation of Nonstationary Sources. IEEE Trans. Speech Audio Processing, **8** (2000) 320-327
2. Lee, T.W., Lewicki, M.S., et al.: Blind Source Separation of More Sources than Mixtures Using Overcomplete Representations. IEEE Signal Processing Letters, **6** (1999) 87-90
3. Tan, L., Wei, G.: Multi-input Multi-output(MIMO) Blind Deconvolution via Maximum Entropy(ME) Method. Acta Electronica Sinica, **28** (2000) 114-116
4. Yellin, D., Weinstein, E.: Criteria for Multichannel Signal Separation. IEEE Trans. Signal Processing, **42** (1994) 2158-2168
5. Thi, H.N., Jutten, C.: Blind Source Separation for Convolutive Mixtures. Signal Processing, **45** (1995) 209-229
6. Liu, J., He, Z.: A Survey of Blind Source Separation and Blind Deconvolution. Acta Electronica Sinica, **30** (2002) 570-576
7. Tokkola, K.: Blind Separation of Delayed Sources Based on Information Maximization. Proc of ICASSP, Atlanta: ICASSP (1996) 3509-3512
8. Li, Y., Cichocki, A., Amari S.: Analysis of Sparse Representation and Blind Source Separation. Neural Computation, **16** (2004) 1193-1234
9. Bofill, P., Zibulevsky, M.: Underdetermined Source Separation Using Sparse Representations. Signal Processing, **81** (2001) 2353-2362
10. Zibulevsky, M., Pearlmutter, B.A.: Blind Source Separation by Sparse Decomposition in a Signal Dictionary. Neural Computation, **13** (2001) 863-882
11. Donoho, D.L., Elad, M.: Maximal Sparsity Representation via l^1 Minimization. Proc. Nat. Aca, Sci., **100** (2003) 2197-2202
12. Adriana D., Luis C.: A Novel Frequency Domain Approach for Separating Convolutive Mixtures of Temporally-White Signals. Digital Signal Processing, **13** (2003) 301-316
13. 13.Lewicki, M.S., Sejnowski, T.J.: Learning Overcomplete Representations. Neural Computation, **12** (2000) 337-365
14. Takigawa, I., Kudo, M., Toyama J.: Performance Analysis of Minimum l_1-norm Solutions for Underdetermined Source Separation. IEEE Trans. Signal Processing, **52** (2004) 582-591
15. He, Z., Xie, S., Deng, A., Tan, B.: An Adaptive BSS Algorithm for Undetermined Blind Source Separation. Submitted to IEEE Trans. Signal Processing

Blind Separation Combined Frequency Invariant Beamforming and ICA for Far-field Broadband Acoustic Signals

Qi Lv[1], Xianda Zhang[1,⋆], and Ying Jia[2]

[1] Dept. of Automation, Tsinghua University, Beijing 100084, China
[2] Intel China Research Center, Beijing, China

Abstract. Many famous blind source separation (BSS) in frequency domain have been developed while they can still not avoid the permutation problem. We propose a new BSS approach for far-field broadband acoustic signals via combining the frequency invariant bemforming (FIB) technique and complex-valued independent component analysis (ICA). Compared with other frequency methods, our method can avoid the permutation problem and has much faster convergency rate. We also present a new performance measure to evaluate the separation. Finally, the simulation is given to verify the efficiency of the proposed method.

1 Introduction

Blind source separation (BSS) is a method for recovering independent source signals from their mixtures without any prior knowledge of signals and mixing process besides some statistical features [1]. Since the pioneering work by Jutten and Herault [2], BSS has drawn lots of attention in signal processing community and neural networks community [1], [3] ,[4] ,[5] ,[6].

Early BSS studies dealt with an instantaneously mixing process [5] ,[6], while recent reports are mainly concerned with convolutive mixtures [4] ,[7] ,[8] ,[9] which is much more difficult from theoretical and computational points of view. Roughly speaking, BSS methods for convolutive mixtures can be classified into the two types: the time domain approach and frequency domain one.

In time domain, BSS problem can be solved by applying independent component analysis (ICA) directly to the convolutive mixtures model [7]. This type of BSS methods can avoid the permutation indeterminacy which can hardly be avoided in frequency domain and can achieve good separation once the used algorithm converges. Its disadvantage is that ICA for convolutive mixtures is not as simple as ICA for instantaneous mixtures and computationally expensive for long FIR filters because it includes convolution operations [4].

In frequency domain, the convolutive mixtures problem of time domain is converted into instantaneous mixtures problem at each frequency. Hence, the

[⋆] This work was supported by the major programm of the National Natural Science Foundation of China under Grant 60496311 and the Intel Sponsored Research Foundation.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 538–543, 2005.

complex-valued ICA can be applied at each frequency bin [4]. The merit of these approaches is that the ICA algorithm becomes simple and can be performed separately at each frequency [4]. However, the indeterminacy of permutation and gain of the ICA solution becomes a serious problem. Luckily, many permutation correction approaches have been proposed [8] ,[9]. Recently, Sawada et. al. have proposed a robust and precise method for solving the permutation problem [4]. But it is a pity that their approach computationally expensive for permutation correction.

In this paper, we propose a frequency domain approach for blind separation of mixtures of acoustic signals in far field. Our approach consists of two steps: 1) filter using the frequency invariant beamforming (FIB); 2) separation using the complex-valued ICA algorithm in frequency domain. The merit of our approach is that it avoid the permutation problem and has good separated results. We also proposed a new performance measure to evaluate the quality of separation.

The rest of this paper is organized as follows. In section 2 we propose our separation method. After that, a new performance measure is discussed in section 3. The simulation result is given in section 4. Finally, this paper is concluded in Section 5.

2 New BSS Method for Acoustic Signals

In this section, we present our new BSS architecture which has two parts: FIB design and separation matrix design.

We first assume that there are q sensors in a linear array and p unknown sources in far field emitting acoustic signals from direction $\mathbf{\Theta} = [\theta_1, \cdots, \theta_p]$, where θ_i is the direction to the ith source measured relative to the array axis. And we also assume $p < q$. The discrete time signal received at jth sensor is given by

$$x_j[k] = \sum_{i=1}^{p} s_i[k - \tau_j(\theta_i)] + v_j[k] \tag{1}$$

where $v_j[k]$ is the addition white noise, $s_i[k]$ is the ith source signal, and $\tau_j(\theta_i) = d_j \sin\theta_i / c$ is the propagation delay of ith source to the jth sensor, d_j is the position of the jth sensor, c is the propagation velocity of the signals. Define the q-dimensional vector of stacked array data as

$$\mathbf{x}[k] = [x_1[k], \cdots, x_q[k]] \tag{2}$$

with a frequency response given by

$$\mathbf{X}(f) = \mathbf{A}(\mathbf{\Theta}, f)\mathbf{S}(f) + \mathbf{V}(f) \tag{3}$$

where $\mathbf{S}(f) = [s_1(f), \cdots, s_p(f)]$ is the source signal vector in frequency, $\mathbf{V}(f) = [v_1(f), \cdots, v_q(f)]$ is the additive noise vector, and $\mathbf{A}(\mathbf{\Theta}, f)$ is the $q \times p$ source direction matrix with its element $a_{ij} = e^{-j2\pi f \tau_i(\theta_j)}$. The BSS problem is to recover $\mathbf{s}[k] = [s_1[k], \cdots, s_p[k]]$ using only $\mathbf{x}[k]$. Equation (3) indicates that the mixing process at each frequency is instantaneous. Then, the complex-valued ICA approach can be applied at each frequency to solve the BSS problem [4]

$$\mathbf{Y}(f) = \mathbf{W}(f)\mathbf{X}(f) \tag{4}$$

where $\mathbf{Y}(f) = [y_1(f), \cdots, y_p(f)]$ is the separated result in frequency domain, $\mathbf{W}(f)$ is a $p \times q$ separation matrix. The learning rule of the separation matrix is given by [4]

$$\Delta\mathbf{W} = \mu \left[\mathbf{I} - \left\langle \mathbf{\Phi}(\mathbf{Y})\mathbf{Y}^H \right\rangle_t\right] \mathbf{W} \tag{5}$$

where μ is a step-size parameter, $\langle \cdot \rangle_t$ denotes the averaging operator overtime, and $\mathbf{\Phi}(\cdot)$ is some nonlinear function.

For the mixing matrix is different at different frequency, the ICA approach can not avoid the permutation problem (see [4]). If the the mixing matrix is essentially identical for all frequencies, we can serially update the separation matrix from one frequency to another. And the permutation problem need not be considered. Luckily, we have the FIB technique to realize that. The main idea of FIB is to design a filter $\mathbf{b}(f)$ such that the response of this beamformer may be made approximately constant with respect to frequency over the design bandwidth $f_L \sim f_U$, i.e.,

$$r(\theta, f) = \mathbf{b}^H(f)\mathbf{a}(\theta, f) \approx r_{FI}(\theta), \quad \forall\theta, \forall f \in [f_L, f_U] \tag{6}$$

where $\mathbf{a}(\theta, f) = [e^{-j2\pi f\tau_1(\theta)}, \cdots, e^{-j2\pi f\tau_q(\theta)}]^T$, and $r(\theta, f)$ is the response of beamformer. Several methods of designing a FIB have been proposed [10] ,[11]

After we apply an FIB to the received array data, the beamformer output in frequency is

$$\mathbf{Z}(f) = \mathbf{B}^H(f)\mathbf{X}(f) \tag{7}$$

where $\mathbf{Z}(f)$ is referred as the frequency invariant beamspace (FIBS) data observation vector, $\mathbf{B}(f) = [\mathbf{b}_1(f), \cdots, \mathbf{b}_p(f)]$ is $q \times p$ filter response matrix, and $\mathbf{b}_i(f)$ is the ith set of beam shaping filter response vector. By using (3), the FIBS data vector can be rewritten as

$$\begin{aligned}\mathbf{Z}(f) &= \mathbf{B}^H(f)\mathbf{X}(f) \\ &= \mathbf{B}^H(f)\mathbf{A}(\mathbf{\Theta}, f)\mathbf{S}(f) + \mathbf{B}^H(f)\mathbf{V}(f) \\ &= \mathbf{A}_B(\mathbf{\Theta}, f)\mathbf{S}(f) + \mathbf{V}_B(f)\end{aligned} \tag{8}$$

where $\mathbf{A}_B(\mathbf{\Theta}, f) = \mathbf{B}^H(f)\mathbf{A}(\mathbf{\Theta}, f)$ is the $p \times p$ FIBS source direction matrix, and $\mathbf{V}_B(f) = \mathbf{B}^H(f)\mathbf{V}(f)$ is the $p \times 1$ FIBS noise vector.

Because the beamformers are designed to satisfy the frequency invariant property (6), the FIBS source direction matrix is approximately constant for all frequencies within the designed band, i.e., $\mathbf{A}_B(\theta, f) \approx \mathbf{A}_B(\theta)$, $\forall f \in [f_L, f_U]$. Hence, the mixing process of acoustic signals is completely characterized by a single beamspace source direction matrix $\mathbf{A}_B(\theta)$ which is independent of frequency and only decided by the direction-of-arrival (DOA) of the source signals.

After FIB preprocessing, we separate source signals from FIBS data observation vector $\mathbf{Z}(f)$ which can be rewritten as the following mixing model:

$$\mathbf{Z}(f) \approx \mathbf{A}_B(\theta)\mathbf{S}(f) + \mathbf{V}_B(f) \tag{9}$$

Then, we use method (5) to update the separation matrix $\mathbf{W}$ serially from frequency f_L to f_U.

Our proposed method, compared with general frequency domain BSS methods [4] ,[8] ,[9], can avoid the indeterminacy of permutation and scaling because the mixing process with FIB applied is invariant through a broad frequency band. On the other hand, our approach has much faster convergence rate since it uses the information of all the frequency data to update the separation matrix serially.

3 Performance Measure

In this section, we discuss how to measure the quality of separation. We first define

$$k_i(f) \triangleq \arg\max_k |\mathbf{C}_{ik}(f)| \tag{10}$$

$$m_i(f) \triangleq \arg\max_m |\mathbf{C}_{mi}(f)| \tag{11}$$

where $\mathbf{C}(f) = \mathbf{W}(f)\mathbf{A}_B(\theta, f)$. Then we define k_i and m_i as the most frequently occurring numbers of group $k_i(f)$ and $m_i(f)$ respectively. A new performance measure ($\mathbf{P_m}$) of separation is given by

$$\mathbf{P_m} = \frac{\sum_{f=1}^{f_s}\sum_{i=1}^{p}\left(\left|\mathbf{C}_{ik_i}(f)\right|^2 \left|s_{k_i}(f)\right|^2 + \left|\mathbf{C}_{m_i i}(f)\right|^2 \left|s_i(f)\right|^2\right)}{\sum_{f=1}^{fs}\sum_{i=1}^{p}\left(\sum_{j=1,j\neq k_i}^{p}\left|\mathbf{C}_{ij}(f)\right|^2 \left|s_j(f)\right|^2 + \sum_{j=1,j\neq m_i}^{p}\left|\mathbf{C}_{ji}(f)\right|^2 \left|s_i(f)\right|^2\right)} \tag{12}$$

where f_s is the sampling frequency. The proposed performance measure of separation, in a sense, describes the average ratio of the total separated signal power to the total interference power.

4 Simulation Result

To demonstrate the efficiency of the proposed approach, we consider two speech signals (see Fig. 1) of 30s with the sampling frequency $f_s = 48$k Hz emitting from $-5°$ and $20°$ respectively. The source signals can be downloaded from the internet address http://medi.uni-oldenburg.de/demo/demo_separation.html.

The frame size of short time Fourier transform (STFT) is 1600, and the STFT overlap is 1200. Two FIB's were designed according to [11] to be frequency invariant over the frequency band $[300, 3400]$ Hz. The aperture size is 5 half-wavelengths. Then, 17 sensors are needed at least and the array is approximately 2.8m long. We compared our method with the one in [4]. The step size in equation (5) is 0.0005 for our approach while it is 0.01 for the method in [4]. The nonlinear function is selected as $\mathbf{\Phi}(\cdot) = e^{j\cdot\arg(\cdot)}$. The separation performance defined by equation (12) is plotted in Fig. 2.

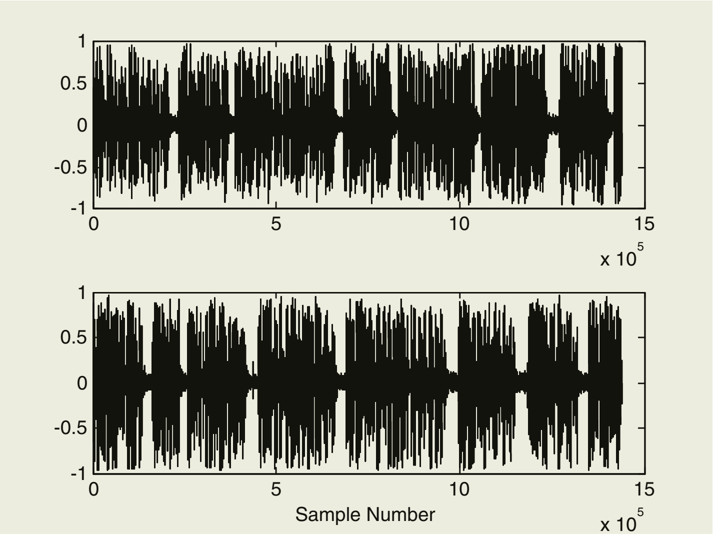


Fig. 1. Two source signals.

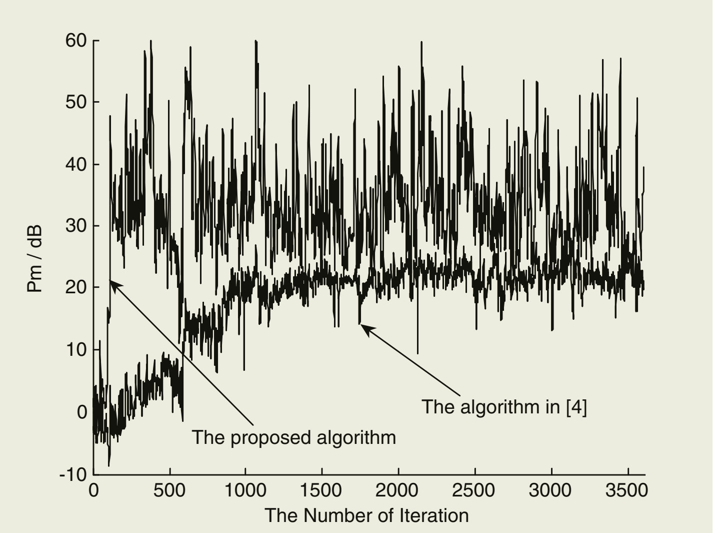


Fig. 2. The performance measure of separation.

5 Conclusion

We have proposed a new BSS approach by combining the FIB technique and complex-valued ICA for far-field broadband acoustic signals. The application of FIB makes our approach avoid the permutation problem, and the updating of separation matrix can be realized in the frequency serially which brings much faster convergence rate than other frequency-domain methods. We also proposed a new performance measure for the ability of separation. At last, the simulation result is given to verify the efficiency of the proposed approach.

References

1. Comon, P.: Independent Component Analysis, a New Concept Signal Processing **36** (1994) 287–314
2. Jutten, C., Herault, J.: Blind Separation of Sources Part I: An Adaptive Algorithm Based on Neuromimetic Architecture. Signal Processing **24** (1991) 1–10
3. Amari, S.: Natural Gradient Works Efficiently in Learning. Neural Computation **10** (1998) 251–276
4. Sawada, H., Mukai, R., Araki, S., Makino, S.: A Robust and Precise Method for Solving the Permutation Problem of Frequency-domain Blind Source Separation. IEEE Trans. on Speech and Audio Processing, **12** (2004) 530–538
5. Cardoso, J.F., Laheld, B.: Equivariant Adaptive Source Separation. IEEE Trans. Signal Processing, **44** (1996) 3017–3029
6. Zhu, X.L., Zhang, X.D.: Adaptive RLS Algorithm for Blind Source Separation Using a Natural Gradient. IEEE Signal Processing Letters, **9** (2002) 432–435
7. Douglas, S.C., Sun, L., Ser, X.: Convolutive Blind Separation of Speech Mixtures Using the Natural Gradient. Speech Commun, **39** (2003) 65–78
8. Kurita, S., Saruwatari, H., Kajita, S., Takeda, K., Itakura, F.: Evalutation of Blind Signal Separation Method Using Directivity Pattern Under Reverbrant Conditions. In Proc. ICASSP (2000) 3140–3143
9. Ikram, M.Z., Morgan, D.R.: A Beamforming Approach to Permutation Alignment for Multichannel Frequency-domain Blind Speech Separation. In Proc. ICASSP (2002) 881–884
10. Ward, D.B., Kennedy, R.A., Williamson, R.C.: Fir Filter Design for Frequencyinvariant Beamformers. IEEE Signal Processing Lett. **3** (1996) 69–71
11. Ward, D.B., Kennedy, R.A., Williamson, R.C.: Theory and Design of Broadband Sensor Arrays with Frequency Invariant Far-field Beam Patterns. J. Acoust. Soc. Amer. **97** (1995) 1023–1034

Blind Source Separation-Based Encryption of Images and Speeches

Qiuhua Lin[1], Fuliang Yin[1], and Hualou 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. Blind source separation (BSS) has been successfully applied in many fields such as communications and biomedical engineering. Its application for image and speech encryption, however, has been scarce. Motivated by the fact that the security of many public-key cryptosystems relies on the apparent intractability of the computational problems such as the integer factorization problem, we present a BSS-based method for encrypting images and speeches by utilizing the BSS underdetermined problem. We discuss how to construct the mixing matrix for encryption and how to generate the key signals. Computer simulation results show that the BSS-based method has high level of security.

1 Introduction

With the widespread use of the computer networks in our daily life, more and more image and speech information has been transmitted over the Internet. Their security has become an important topic in the field of information security. As such, a variety of the encryption techniques have been introduced. The image encryption methods mainly include encryption without compression, encryption combining compression, and partial encryption of compressed images [1], [2]. As for speech encryption, there are traditional analogue encryption techniques such as scrambling in frequency and time domain [3], and some new encryption methods such as chaotic cryptosystem [4].

Blind source separation (BSS) aims to recover a set of source signals from their observed mixtures without knowing the mixing coefficients. So far, BSS has been successfully applied in many fields such as communications and biomedical engineering [5], [6]. Its application for image and speech encryption, however, has been scarce [7], [8], [9]. In this paper, we present a BSS-based method for encrypting images and speeches by utilizing the BSS underdetermined problem since the security of many cryptographic techniques depends upon the intractability of computational problem such as the integer factorization problem [10]. By using the key signals, we first construct the BSS underdetermined problem in encryption process, and then circumvent this problem by means of the key signals to achieve the decryption. We show that the BSS-based method can achieve high level of security by properly con-

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 544–549, 2005.

structing the mixing matrix and by generating the key signals that retains almost all the characteristics of the one-time pad cipher [10].

2 BSS Model and Underdetermined Problem

Suppose that there exist *M* independent source signals $s_1(t),\ldots,s_M(t)$ and *N* observed mixtures $x_1(t),\ldots,x_N(t)$ of the source signals (usually $N \geq M$). The simple BSS mixing model is:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) \tag{1}$$

where $\mathbf{s}(t) = [s_1(t),\ldots,s_M(t)]^T$, $\mathbf{x}(t) = [x_1(t),\ldots,x_N(t)]^T$, and $\mathbf{A}$ is an $N \times M$ mixing matrix that contains the mixing coefficients. The goal of BSS is to find an $M \times N$ demixing matrix $\mathbf{W}$ such that $M \times 1$ output vector

$$\mathbf{u}(t) = \mathbf{W}\mathbf{x}(t) = \mathbf{W}\mathbf{A}\mathbf{s}(t) = \mathbf{P}\mathbf{D}\mathbf{s}(t) \tag{2}$$

where $\mathbf{P} \in \mathbf{R}^{M \times M}$ is a permutation matrix and $\mathbf{D} \in \mathbf{R}^{M \times M}$ is a diagonal scaling matrix.

When $N < M$, i.e., the number of the mixed signals is less than that of the source signals, BSS becomes a difficult case of the underdetermined problem, in which the complete separation is usually out of the question [11].

3 BSS-Based Encryption Method

The block diagram of the BSS-based encryption is shown in Fig. 1. $s_1(t), ..., s_P(t)$, *t* = 1, … , *T*, are *P* ($P > 1$) original signals to be encrypted simultaneously where *T* is the size (data length) of each original signal. $s_{n1}(t), ..., s_{nP}(t)$ are *P* key signals. I_0 is the secret seed for generating the key signals. $x_1(t), ..., x_P(t)$ are *P* encrypted signals for transmission. $\hat{s}_1(t), ..., \hat{s}_P(t)$ are *P* decrypted signals by BSS.

3.1 Encryption

The BSS-based encryption first generates the *P* key signals $s_{n1}(t), ..., s_{nP}(t)$ and a $P \times 2P$ mixing matrix $\mathbf{A}$ for encryption, and then mixes the *P* original signals with the *P* key signals by using the mixing matrix $\mathbf{A}$ according to the BSS mixing model (1), i.e., $\mathbf{s}(t) = [s_1(t), ..., s_P(t), s_{n1}(t), ..., s_{nP}(t)]^T$, and the *P* encrypted images are $\mathbf{x}(t)$ $= [x_1(t),..., x_P(t)]^T$ $= \mathbf{A}\ \mathbf{s}(t)$. Obviously, the encryption process turns BSS into the difficult case of the underdetermined problem since there are 2*P* source signals but *P* mixed signals. In such a case, the *P* original signals cannot be recovered from the *P* encrypted signals through BSS without the *P* key signals.

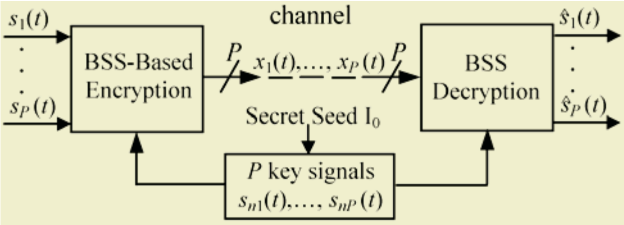


Fig. 1. Block diagram of BSS-based encryption method

3.2 Decryption

On the receiving side, when the P encrypted signals $x_1(t),\ldots,x_P(t)$ and the P key signals $s_{n1}(t),\ldots,s_{nP}(t)$ are available, they are first combined to form $2P$ mixed signals $\mathbf{x}(t)=[x_1(t),\ldots,x_P(t),s_{n1}(t),\ldots,s_{nP}(t)]^T$, on which the BSS algorithm are then performed. It is known that $\mathbf{s}(t)=[s_1(t),\ldots,s_P(t),s_{n1}(t),\ldots,s_{nP}(t)]^T$, thus there are $2P$ mixed signals and $2P$ source signals now, the underdetermined problem resulting from the BSS-based encryption becomes the simplest BSS case. As a result, the P original signals and the P key signals can be well recovered together through BSS. After discarding the P key signals, we can obtain the P decrypted signals $\hat{s}_1(t),\ldots,\hat{s}_P(t)$.

From (2), we can find that $\hat{s}_1(t),\ldots,\hat{s}_P(t)$ and $s_1(t),\ldots,s_P(t)$ may have order and amplitude difference, i.e., the decrypted signals by BSS may have small difference with the original signals. Therefore, the BSS-based method is suitable for encrypting images and speeches permitting small distortion but not for text data encryption.

3.3 Image Encryption

Given P original images to be encrypted, the P original signals $s_1(t),\ldots,s_P(t)$ in Fig. 1, i.e., the P input signals of the BSS-based method, can be easily obtained by converting two-dimensional (2-D) images into one-dimensional (1-D) data. After decrypted, the 1-D signals are then returned to 2-D images for presentation [8].

3.4 Speech Encryption

To encrypt speech signal, we first divide the original speech signal into equal time periods called frames, and then sub-divide each frame into P smaller equal time periods called segments, which are the P original signals $s_1(t),\ldots,s_P(t)$ in Fig. 1. The BSS-based method will encrypt the speech signal frame by frame [9].

However, as mentioned above, the P decrypted signals $\hat{s}_1(t),\ldots,\hat{s}_P(t)$ may be different in orders and amplitudes from the P original signals $s_1(t),\ldots,s_P(t)$. To correctly recover the original speech from the decrypted signal by BSS, we make use of some prior knowledge about the waveform of the P original signals $s_1(t),...,s_P(t)$ including the zero cross count *zcc*, the maximum s_{max} and the minimum s_{min}. The *zcc* is used for the order rearrangement, whereas s_{max} and s_{min} are used for the amplitude reconstruction.

4 Key Signals Generating

From Fig. 1, we can see that the key signals have the same size and the same number with those of the original signals to be encrypted. Therefore, the BSS-based method has the merit of simulating the key characteristics of the one-time pad cipher that is unconditionally secure [10] since the key signals are pseudo-randomly generated and have the same length as the original signals. Considering that BSS usually cannot separate more than two Gaussian signals, the key signals need to be generated according to the following two conditions:

Condition 1: The key signals are statistically independent.

Condition 2: The key signals are non-Gaussian.

In practice, the key signals are formed with pseudorandom values with uniform distribution, which are generated by a pseudorandom number generator (PRNG) initialized with the secret seed I_0 (refer to Fig. 1).

5 Mixing Matrix Constructing

Unlike the normal BSS that aims to completely separate all source signals, the mixing matrix $\mathbf{A}$ for encryption must ensure the illegal users cannot recover the P original signals $s_1(t),...,s_P(t)$ from the P encrypted signals $x_1(t),...,x_P(t)$. Therefore, we construct the mixing matrix $\mathbf{A}$ based on the *L*-row decomposable theorem (Theorem 1 in [11]) that has been proved for the source separability of BSS as follows:

$$\mathbf{A} = [\mathbf{B} \quad \beta\mathbf{B}] \tag{3}$$

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. We have shown that the P original signals are inseparable from the P encrypted signals without the P key signals under the mixing matrix $\mathbf{A}$ in (3) [8]. In addition, with proper selection of β, e.g., $\beta \ge 10$ for image encryption and $\beta \ge 1$ for speech encryption, the energy level of the key signals is much high than that of the original signals. Therefore, the original signals can be well masked by the key signals to achieve the security goal.

6 Experimental Results

To illustrate the efficiency of the proposed method, we carried out extensive computer simulations with nature images and speeches. Fig. 2 shows one example of encrypting two original images “Lena” and “Girl” with 256 grey levels, the size of which is 256×256. Fig. 3 shows one example of encrypting a speech signal of 6.875 seconds long (110000 samples), sampled at 16 KHz. In the experiments, the Comon’s algorithm [12] is used for the BSS decryption. Other popular BSS algorithms can also be used [8], [9]. To save the space, we omit the values of two mixing matrixes $\mathbf{A}$ for encryption with $\beta=10$ for image encryption and $\beta=1$ for speech encryption. From Fig. 2 and Fig. 3, it is easy to see that the encrypted signals have reached the security goal, and the decrypted signals are all of good quality.

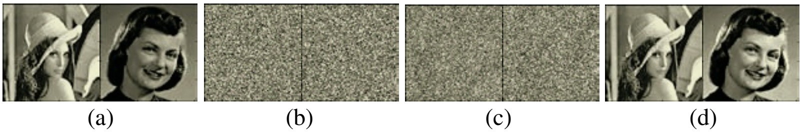


Fig. 2. Example of BSS-based encryption of two original images (P=2, T=65536). (a) Two original images $s_1(t)$ and $s_2(t)$. (b) Two key images $s_{n1}(t)$ and $s_{n2}(t)$. (c) Two encrypted images $x_1(t)$ and $x_2(t)$. (d) Two decrypted images $\hat{s}_1(t)$ and $\hat{s}_2(t)$ with Comon’s algorithm

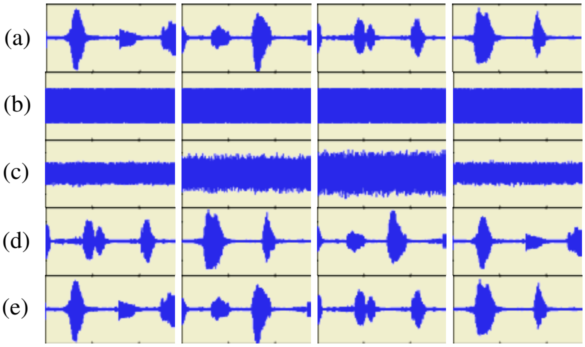


Fig. 3. Example of BSS-based encryption of a speech signal (spoken digits 1-10 with a little music interference) with 110000 samples (P=4, T=27500). (a) Four original speech segments $s_1(t)$, … , $s_4(t)$. (b) Four key signals $s_{n1}(t)$, … , $s_{n4}(t)$. (c) Four encrypted speech segments $x_1(t)$, … , $x_4(t)$. (d) Four decrypted speech segments $\hat{s}_1(t)$, ... , $\hat{s}_4(t)$. (e) Four recovered speech segments after waveform reconstruction by using zcc, s_{max} and s_{min} in Table 1

Table 1. Waveform information about 4 original speech segments in speech encryption

	$s_1(t)$	$s_2(t)$	$s_3(t)$	$s_4(t)$
zcc	4315	5905	7456	5239
s_{max}	0.8594	0.7306	0.5633	0.8795
s_{min}	-0.9597	-1.0002	-0.5514	-0.7750

7 Conclusion

Motivated by the fact that the security of many public-key cryptosystems relies on the apparent intractability of the computational problems such as the integer factorization problem, we take advantage of the BSS underdetermined problem to propose a BSS-based encryption method for encrypting images and speeches. Since the key signals approximately have the desired characteristics of the one-time pad cipher, and the mixing matrix for encryption can ensure the inseparability of the original signals without the key signals, the BSS-based method has high level of security.

When encrypting one image, we can split it into P sub-images before encryption, and then recover the original image from the P BSS decrypted signals by the similar post-process used in waveform reconstruction of the BSS decrypted speech signals.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. 60402013, No. 60172073, 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. Beker, H.J., Piper, F.C.: Secure Speech Communications. London Academic Press Inc (1985)
4. Li, K., Soh, Y.C., Li, Z.G.: Chaotic Cryptosystem with High Sensitivity to Parameter Mismatch. IEEE Trans Circuits and Systems I: Fundamental Theory and Applications, **50** (2003) 579–583
5. Hyvärinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. John Wiley (2001)
6. Cichocki, A., Amari, S.: Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications. John Wiley (2003)
7. 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
8. Lin, Q.H., Yin, F.L., Zheng, Y.R.: Secure Image Communication Using Blind Source Separation. Proc. IEEE 6th CAS Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communication, **I** (2004) 261–264
9. Lin, Q.H., Yin, F.L., Mei, T.M., Liang, H.L.: A Speech Encryption Algorithm Based on Blind Source Separation. Proc. Int. Conf. Communications, Circuits and Systems, **II** (2004) 1013–1017
10. Menezes, A., Van Oorschot, P., Vanstone, S.: Handbook of Applied Cryptography. CRC Press Inc (1996)
11. Cao, X.R., Liu, R.W.: General Approach to Blind Source Separation. IEEE Trans Signal Processing, **44** (1996) 562–571
12. Comon, P.: Independent Component Analysis, A New Concept?. Signal Processing, **36** (1994) 287–314

A Digital Audio Watermarking Scheme Based on Blind Source Separation

Xiaohong Ma, Chong Wang, Xiangping Cong, and Fuliang Yin

School of Electronic and Information Engineering
Dalian University of Technology, Dalian 116023, China
maxh@dlut.edu.cn

Abstract. An effective digital audio watermarking scheme based on blind source separation (BSS) for blind watermark detection is proposed in this paper. In the embedding procedure, the original watermark is modulated by chaotic sequence to generate an embedding watermark. Then watermarked audio signal is combined by two signals which are mixed by original audio signal and embedding watermark according to BSS mixing model. In the detecting procedure, watermarked audio signal is split into two signals first, and then BSS technique is applied to them to obtain the separated embedding watermark. At last, we can determine the watermark presence by correlation values between the separated embedding watermark and embedding watermark. Experimental results show the effectiveness of the proposed method.

1 Introduction

The widespread use of the Internet, proliferation of the low-cost and reliable storage devices, and deployment of long-distance Gbps networks has made the copyright protection of digital audio works more and more important and urgent. Therefore, a robust audio watermarking technology to protect copyrighted digital audio datum from illegal sharing and tampering is needed. The audio watermarking techniques proposed so far can be divided into two main groups: those embed the watermark directly in the time domain and those operate in the transformed domain. The representative time domain schemes are the Least Significant Bits (LSB) [1] and echo hiding [2], [3]. The schemes of transformed domain usually use DFT [4], DCT [5] or DWT [6]. Recently, a new transformed domain audio watermarking scheme is proposed in the cepstrum domain [7].

In this paper, we explore a blind source separation based audio watermark scheme which is different from the method in [8]. In [8], the demixing matrix was known, but it is unknown in our method. In the embedding procedure, we regard the original audio signal and the embedding watermark which is generated by spread spectrum of the *m* sequence as two independent source signals. The two observed data could be obtained by mixing two source signals according to BSS mixing model. Then the watermarked audio signal is formed by the combination of them. In the detecting procedure, watermarked audio signal is split into two signals first, and then BSS technique is applied to them to obtain the separated embedding watermark. At last,

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 550–555, 2005.

we can determine the watermark presence by correlation values between the separated embedding watermark and embedding watermark.

2 Blind Source Separation

Suppose that there exist N independent source signals $s_1, s_2, \cdots, s_N$ and M observed mixtures $x_1, x_2, \cdots, x_M$ of the source signals (usually $M \geq N$). Assuming that these mixtures are linear, instantaneous and noiseless, in this case the BSS mixing model can be represented as

$$\mathbf{x} = \mathbf{A} \cdot \mathbf{s} \tag{1}$$

where $\mathbf{s} = [s_1, s_2, \cdots, s_N]^{\mathrm{T}}$ and $\mathbf{x} = [x_1, x_2, \cdots, x_M]^{\mathrm{T}}$ represent source signals vector and observed vector respectively, and $\mathbf{A}$ is the mixing matrix. The aim of BSS is to find a demixing matrix $\mathbf{V}$ to recover the source signals

$$\hat{\mathbf{s}} = \mathbf{V} \cdot \mathbf{x} = \mathbf{V} \cdot \mathbf{A} \cdot \mathbf{s} = \mathbf{C} \cdot \mathbf{s} \tag{2}$$

where $\hat{\mathbf{s}}$ is the estimation of source signals vector.

3 Watermark Embedding

The block diagram of watermark embedding is shown in Fig. 1. There are three main stages, the embedding watermark generation, the sources mixing and the combination.

3.1 Embedding Watermark Generation

To ensure security of the proposed watermarking scheme, we use a chaotic sequence to modulate the original watermark w_0.

First, the original watermark w_0 is a m sequence which is generated by a linear feedback shift register. The length of m sequence is expressed as

$$l = 2^n - 1 \tag{3}$$

where n is the number of the registers.

Second, the Logistic map is defined as

$$x_{k+1} = \mu x_k (1 - x_k), \qquad x_k \in (0,1) \tag{4}$$

where iterative coefficient μ controls the characteristics of the sequence. The Logistic map is a chaotic sequence as $3.5699456 < \mu \leq 4$. Since chaotic sequences are sensitive to the initial condition, therefore, the Logistic map is used in our scheme. We use a chaotic sequence of length k as the spreading codes.

Last, the original watermark w_0 is modulated by chaotic sequence to generate an embedding watermark w with length $L = k \cdot l \cdot L_c$, where L_c is the length of carrier signal.

To improve the security of the proposed method, we take the initial value x_0 and the iterative coefficient μ as the secret key.

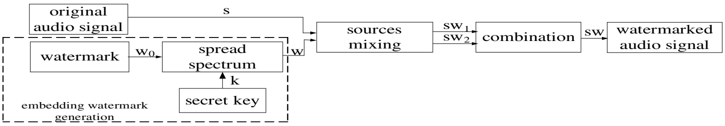


Fig. 1. Block diagram of watermark embedding part

3.2 Source Mixing

We regard original audio signal s and the embedding watermark w as two source signals, and both of them must have the same length L. We use (1) as the mixture model, and a 2×2 matrix $\mathbf{A}=\begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix}$ as the mixing matrix. $\mathbf{A}$ should be a full rank matrix, and the elements of it which correspond to the embedding watermark are smaller than other elements relating to the original audio signal to achieve the robustness and inaudibility. The equation of this mixing procedure is as follows

$$\mathbf{x}=\begin{bmatrix} sw_1 \\ sw_2 \end{bmatrix}=\mathbf{A}\cdot\mathbf{s}=\begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix}\begin{bmatrix} s \\ w \end{bmatrix}=\begin{bmatrix} a_1 s+b_1 w \\ a_2 s+b_2 w \end{bmatrix} \tag{5}$$

where sw_1 and sw_2 are two observed signals with length L.

3.3 Combination

To convert two observed signals sw_1 and sw_2 into one watermarked audio signal, we take following two steps. First, generate a new signal sw_m using following method

$$\begin{cases} sw_m(2i-1)=sw_1(i), \\ sw_m(2i)=sw_2(i), \end{cases} \quad i=1,2,\cdots,L \tag{6}$$

where the length of signal sw_m is $2L$.

Then, the sample frequency of the signal sw_m is doubled to generate the watermarked audio signal sw to meet the expectation of no distortion in listening test.

4 Watermark Detecting

The block diagram of watermark detecting is shown in Fig. 2. There are mainly four stages in this part, the embedding watermark generation which is the same as the watermark embedding part, the splitter, the BSS demixing and correlation detection.

4.1 Splitter

As the watermarked audio signal may be attacked, we use symbol sw' to represent the watermarked audio signal in the watermark detecting part. First, the watermarked

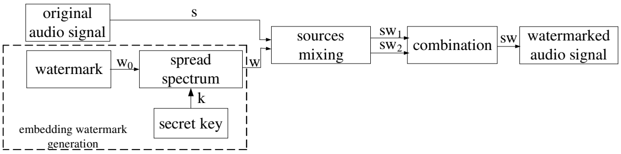


Fig. 2. Block diagram of watermark detecting part

audio signal sw' is down sampled to form the signal sw_m' whose length is $2L$. Then, it is split into two observation signals sw_1' and sw_2' according to following method

$$\begin{cases} sw_1'(i) = sw_m'(2i-1), \\ sw_2'(i) = sw_m'(2i), \end{cases} \quad i = 1,2,\cdots,L \tag{7}$$

where the length of signal sw_1' and sw_2' are both L.

4.2 BSS Demixing

Applying [9] to the two observation signals sw_1' and sw_2', we can get the separated original audio signal $\hat{s}$ and the separated embedding watermark signal $\hat{w}$.

4.3 Correlation Detection

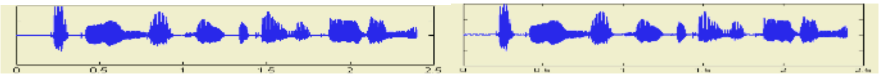
If we knew the secret key x_0 and μ exactly, the embedding watermark w would be generated correctly. Then the correlation values between the separated embedding watermark $\hat{w}$ and generated embedding watermark w will have a distinct peak in the correct position. Otherwise, there will be no distinct peak or exist a peak but in wrong position. Using this method, we can judge whether there exist watermark or not.

5 Experimental Results

The sample frequency of the original audio signal was 8kHz. The number of the registers was $n = 5$ to generate the original watermark $\mathbf{w}_0$, which length was $l = 31$. Chaotic sequence of length $k = 8$ was used as the spreading codes. The carrier frequency was 100Hz and length of carrier signal was $L_c = 100$. The elements of $\mathbf{A}$ was set to be $a_1 = 0.9$, $b_1 = 0.1$, $a_2 = 0.95$, and $b_2 = 0.05$ respectively.

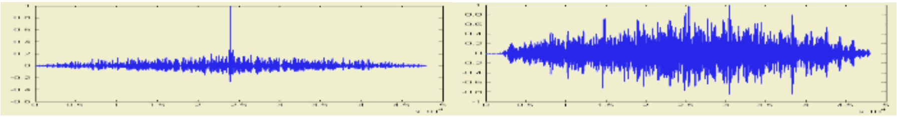
The waveform of original audio signal and watermarked audio signal are shown in Fig. 3(a) and 3(b) respectively. From Fig. 3 we can see that there is no distortion between them. And the same is in the hearing test.

Without any attack, the correlation detection results with correct and incorrect secret key are shown in Fig. 4(a) and 4(b) respectively. It shows that we can't detect the presence of watermark by using incorrect key.



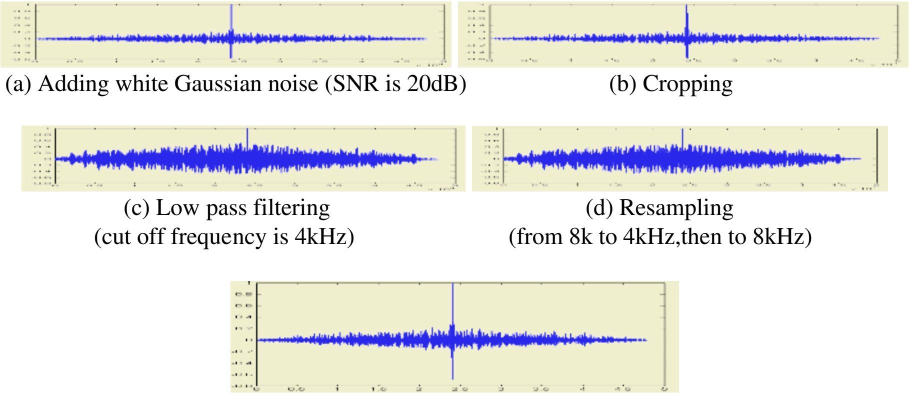
(a) Original audio signal (b) Watermarked audio signal

Fig. 3. Waveforms of the audio signal



(a) With correct secret key (b) With incorrect secret key(1% difference)

Fig. 4. Correlation detection results with correct and incorrect secret key



(a) Adding white Gaussian noise (SNR is 20dB) (b) Cropping

(c) Low pass filtering
(cut off frequency is 4kHz)

(d) Resampling
(from 8k to 4kHz,then to 8kHz)

(e) MP3 compression

Fig. 5. Correlation detection results with various attacks

A variety of experiments are performed to demonstrate the robustness and effectiveness of the proposed method under different attacks, such as adding noise, cropping, low pass filtering, resampling and MP3 compression, are shown in Fig. 5. The results show that our scheme survives every attack.

6 Conclusion

This paper presents a robust audio watermarking scheme based on BSS for blind watermark detection. Our embedding scheme is quite simple, and very large watermark information can be embedded into the original audio signal. Moreover, the security of the secret key is assured by the sensitivity of the initial value of the chaotic sequence. In particular, we can implement the exactly synchronization without embedding additional synchronous information into original audio signal. Experimental results show that there is no audible distortion between the original audio signal and the watermarked audio signal. And the proposed method is robust enough to against the general signal processing operation.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. 60172073, and No. 60372082.

References

1. Gerzon, M. A., Graven, P. G.: A High-rate Buried-data Channel for Audio CD. Journal of the Audio Engineering Society, **43** (1995) 3–22
2. Kim, H. J., Choi, Y. H.: A Novel Echo-hiding Scheme With Backward and Forward Kernels. IEEE Trans. on Circuits and Systems for Video Technology, **13** (2003) 885–889
3. Ko, B. S., Nishimura, R., Suzuki, Y.: Log-scaling Watermark Detection in Digital Audio Watermarking. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04), **3** (2004) III-81–84
4. Arnold, M.: Audio Watermarking: Features, Applications and Algorithms. IEEE International Conference on Multimedia and Expo(ICME 2000), **2** (2000) 1013–1016
5. Yeo, I. K., Kim, H. J.: Modified Patchwork Algorithm: A Novel Audio Watermarking Scheme. IEEE Transactions on Speech and Audio Processing, **11** (2003) 381–386
6. Li, X. Y., Zhang, M., Sun, S. L.: Adaptive Audio Watermarking Algorithm Based on SNR in Wavelet Domain. 2003 International Conference on Natural Language Processing and Knowledge Engineering, (2003) 287–292
7. Lee, S. K., Ho, Y. S.: Digital Audio Watermarking in the Cepstrum Domain. IEEE Trans. On Consumer Electronics, **46** (2000) 744–750
8. Toch, B., Lowe, D., Saad, D.: Watermarking of Audio Signals Using Independent Component Analysis. Third International Conference on Web Delivering of Music (WEDELMUSIC 2003) 71–74
9. Hyvärinen, A., Oja, E.: A Fast Fixed-point Algorithm for Independent Component Analysis. Neural Computation, **9** (1997) 1483–1492

Lidar Signal Processing for Under-water Object Detection

Vikramjit Mitra[1], Chiajiu Wang[2], and Satarupa Banerjee[3]

[1] ECE, Worcester Polytechnic Institute, MA, USA
vmitra@ece.wpi.edu
[2] ECE, University of Colorado, Colorado Springs, CO, USA
cwang@eas.uccs.edu
[3] CIS, Villanova University, PA, USA
satarupa.banerjee@villanova.edu

Abstract. This paper presents Artificial Neural Network (ANN) based architecture for underwater object detection from Light Detection And Ranging (Lidar) data. Lidar gives a sequence of laser backscatter intensity obtained from laser shots at various heights above the earth surface. Lidar backscatter can be broadly classified into three different classes: water-layer, bottom and fish. Multilayered Perceptron (MLP) based ANN architecture is presented, which employ different signal processing techniques at the data preprocessing stage. The Lidar data is pre-filtered to remove noise and a data window of interest is selected to generate a set of coefficient that acts as input to the ANNs. The prediction values obtained from ANNs are fed to a Support Vector Machine (SVM) based Inference Engine (IE) that presents the final decision.

1 Introduction

Aerial survey of Biomass using Lidar technique is a valuable tool for fishery research [1], which outperforms conventional sonar, and echo sounder based techniques, by its high efficiency in remote object classification [2], high area coverage with high data density and rapid response [3]. Underwater fish detection from Lidar backscatter signals requires knowledge of the fish-reflectivity, which is in the range of 5% to 50% [4, 5]. Lidar systems transmit and receive Laser beams at high frequencies and operate in ultraviolet, visible and infrared region. The transmitted Laser beam interacts with and is changed by the target, part of which is reflected back to the instrument receiver where it is analyzed. This paper presents the result obtained from the implementation of MLP based ANN architecture using different signal preprocessing techniques. The signal processing techniques extract features from Lidar data that are used to train the ANNs.

2 The Lidar Data

The data used in this research were obtained from National Oceanic and Atmospheric Administration (*NOAA*), who have employed a nonscanning, radiometric Lidar (Fig. 2.1), where the laser beam is linearly polarized and frequency doubled, that

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 556–561, 2005.

produces 100 mJ of green light (532 nm) in a 12 ns pulse at the rate of 30 pulse/s. The receiver device uses a 17 cm diameter refracting telescope, where a front polarizer selects either the cross or co-polarized component. It is observed [6] that crosspolarized component produces the best contrast between fish and other particles from water and the backscatter attenuation coefficient is lower at night than daylight. This paper considers copolarized component used at daylight. Fig. 2.2 shows a typical Lidar backscatter plot, where the signal intensity decays exponentially. The backscatter for underwater objects has a small visible bump on the decay portion due to return from a school of fish. Lidar return for bottom has two visible peaks (Fig. 2.2), one due to the air-water interface and the other from the bottom return. To recognize a visible bump as a school of fish, it is necessary to monitor 5 consecutive Lidar pulses. Churnside [6] proposed two techniques, (1) 2-Point fit method and (2) 2-Point fit for clear-water return and Gaussian fit for fish return, to detect underwater fish schools. Lidar return decays exponentially as a function of scattering depth in water due to absorption and scattering:

$$S_w(z)=a(e^{-2\alpha z}\Big/(n\theta h+z)^2)+b\ . \tag{1}$$

where, $S_w(z)$ = Lidar backscatter signal, a = amplitude of laser pulse energy, α = Lidar attentuation coefficient, z = depth below water surface, h = height above the surface, n = refractive index of water, θ = ratio of cosine of transmitted beam angle and cosine of incidence angle and b = background signal level. There is an additional contribution due to the presence of a fish school and is given as:

$$S_W(z)=a[1+\frac{\beta_f(z)}{\beta_w}][e^{-2\alpha z}\Big/(n\theta h+z)^2]+b\ . \tag{2}$$

where, $\beta_f(z)$ and β_w are backscatter coefficients of the fish and water respectively. The existing techniques are largely dependent upon visual judgment of Lidar return patterns and require human intervension for classifyng Lidar returns. ANN based systems have been proposed [7] to do the classification on-the-fly.

3 Lidar Signal Processing

The raw Lidar data was contaminated with high frequency noise, which is removed by a Low-pass Butterworth Filter. It is observed that the Lidar return before water surface and after a certain depth below water surface do not contain any significant information regarding underwater objects; hence those portions are eliminated by taking rectangular window of the filtered signal. It is also seen [7] that the window starting from the first peak to the point where the backscatter intensity decays to 0.2 times its peak contain most of the information regarding underwater objects. The selected window is processed by 3 different techniques: (1) Linear Prediction Coding (LPC), (2) Fast Fourier Transform (FFT) and LPC coding, (3) FFT, LPC and Polynomial Approximation Coding (PAC).

3.1 Linear Predictive Coding (LPC)

LPC is widely used in speech signal processing [8]. In LPC an estimate $\hat{s}(n)$ for a sample value $s(n)$ is given as a linear combination of previous sample values, which is done by taking a weighted sum of a finite number, N, of previous signal values. LPC filter is assumed to be a conventional all-pole filter, and the optimal coefficients, are obtained from the Yule-Walker equations. This system is characterized by the following equations:

$$\hat{s}(n) = \sum_{i=1}^{N} a_i s(n-i) \ . \tag{3}$$

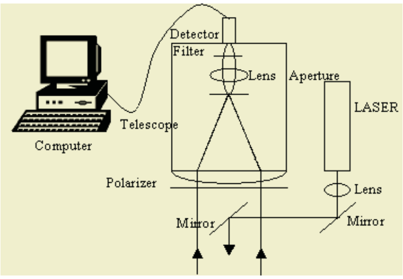


Fig. 2.1. Diagram of Lidar configuration

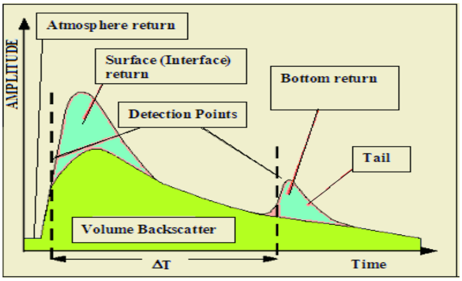


Fig. 2.2. Lidar Backscatter

3.2 Polynomial Approximation Coding (PAC)

The polynomial approximation requires pre-defined linear vector for modeling a signal. It finds the coefficient of a polynomial $p(x)$ of degree 'n', that fits the data, $p(x(i))$ to $y(i)$ in a least square sense. The result p is a row vector of length 'n+1' containing the coefficients in descending power. This research uses an arithmetic progression series as reference vector that generates a coefficient vector, p, of length 5, from the data vector, y.

$$p(x) = p_1 x^n + p_2 x^{n-1} + p_3 x^{n-2} + ... + p_n x + p_{n+1} \ . \tag{4}$$

3.3 Lidar Frequency Spectrum

Change in the exponential decay of the Lidar return is reflected in the frequency spectrum of the different classes. In case of water layer due to lack of any bumps or second peaks in the exponential decay the spectrum has a single spike (Fig. 3.1). For underwater object return there are damped side lobes (Fig. 3.2) in the spectrum due to visible bumps on the decay portion. The bottom return has a distinct side lobe in the spectrum (Fig. 3.3) due to the second peak. Fig. 3.4 gives the overlaying frequency plots for the 3 cases.

3.4 LPC-MLP Approach

In this approach, the data window is coded by LPC algorithm generating 5 coefficients, which are used as the input to a 3 hidden layered MLP. All the results presented in this paper are based on a training set of 900 and testing set of 300 Lidar samples (distributed equally amongst the 3 classes). Fig. 3.5 presents the block diagram of the system.

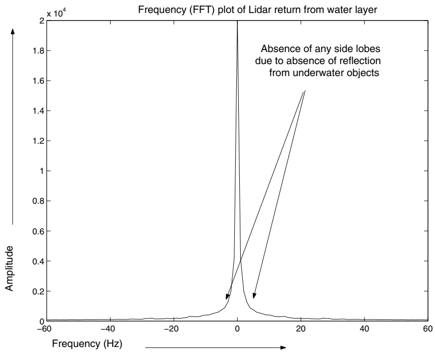


Fig. 3.1. FFT for water-layer

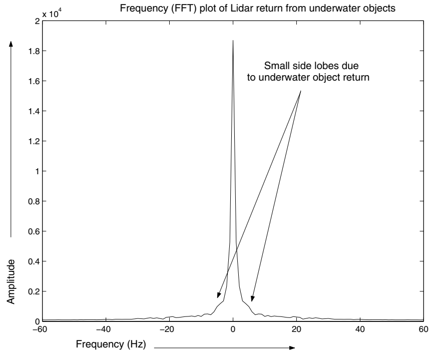


Fig. 3.2. FFT for underwater objects

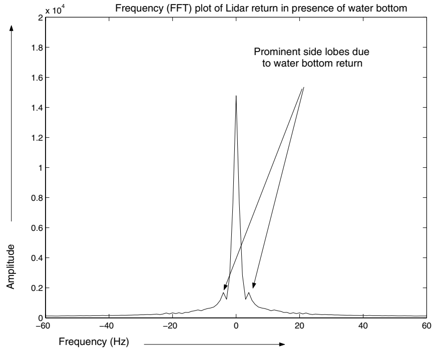


Fig. 3.3. FFT for bottom

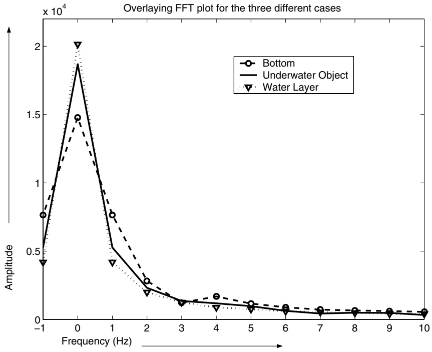


Fig. 3.4. Overlaying FFT of Lidar return

3.5 FFT-LPC-MLP Technique

In this approach the signal processing module consists of an FFT algorithm followed by an LPC coder. The FFT algorithm is used to obtain the spectral information of the signal and as the spectrum is symmetric, hence only half of the frequency points are selected. The data points close to the central peak are crucial as they have the presence or absence of the side lobes that distinguishes between the different classes. 15 points after the central frequency are selected which is coded by LPC and used as MLP inputs.

3.6 FFT-LPC-PAC Parallel MLP Technique

Previous research [7] claims that for certain data sets encoded by LPC the ANN worked with higher accuracy but for other data sets the accuracy was lower. This is also true for data sets encoded by PAC, but in this case the prediction accuracy was found to be higher for those data sets, for which the prediction accuracy was comparatively lower due to LPC coding. The technique discussed in this section incorporates a parallel ANN based architecture with an aim to average the prediction accuracy obtained from LPC and PAC. The LPC and PAC coefficients obtained from the Lidar data processing are fed in parallel to two ANNs and the decisions obtained from each of them are vector multiplied with one another to yield the final prediction, as shown in Fig. 3.6.

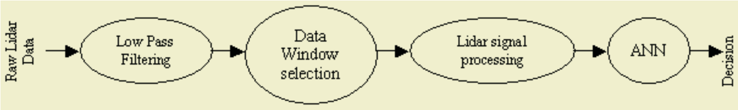


Fig. 3.5. System Block diagram

4 Inference Engine (IE)

Due to computation approximation, the ANN decisions are close but not exactly equal to the desired values. To circumvent this, an SVM based IE is implemented, which has a cascaded structure (Fig. 4.1) that provides the final decision. The first SVM (SVM-1) classifies between Bottom and the remaining, whereas the second SVM (SVM-2) classifies between Fish-school and Water Layer. The Output of this IE is the final decision generated by the proposed architecture.

5 Results

The ANNs presented in this paper are 3-hidden layered MLP, with sigmoid axon, momentum learning rule of 0.7 having a step size of 1.0. The number of processing elements are 50, 100 and 100. To design the SVM based IE, the parameters required are - γ, the regularization parameter and σ^2, the Gaussian bandwidth. These are obtained by minimizing the cost at the initial levels and are optimized by using a Bayesian framework. The SVM parameter values are given in Table 5.1. The prediction accuracies obtained from the three different techniques are presented in Table 5.2.

6 Conclusions

Efficient on-the-fly classification of Lidar return is presented in this paper. The results show that the parallel ANN architecture works with a higher efficiency than the other two. Though the LPC and LPC-FFT based ANNs worked much faster with lower processing time but the FFT-LPC-PAC based system provides higher prediction accuracy. This research considers copolarized Lidar return used at day light, which not only adds substantial noise but also has greater signal attenuation. Due to this the Lidar returns for fish and water-layer were very close to each other and extracting features to distinguish them was a critical problem. Crosspolarized component used at night will result in much better detection accuracy and the system will be almost noise free, hence the low pass filter will not be needed at the signal preprocessing stage.

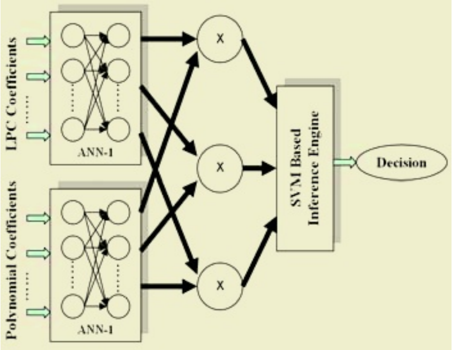
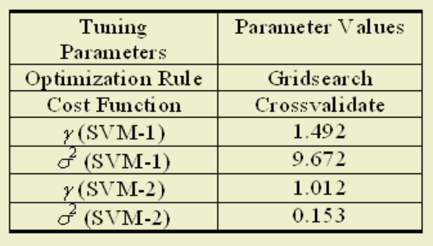


Fig. 3.6. FFT-LPC-PAC parallel MLP diagram

Table 5.1. SVM parameters



Tuning Parameters	Parameter Values
Optimization Rule	Gridsearch
Cost Function	Crossvalidate
γ (SVM-1)	1.492
σ^2 (SVM-1)	9.672
γ (SVM-2)	1.012
σ^2 (SVM-2)	0.153

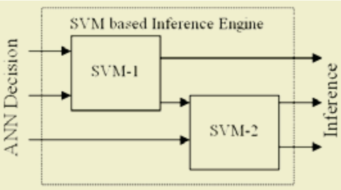
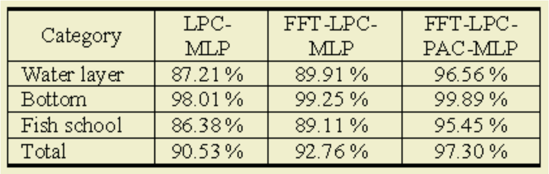


Fig. 4.1. SVM based IE

Table 5.2. Performance comparison



Category	LPC-MLP	FFT-LPC-MLP	FFT-LPC-PAC-MLP
Water layer	87.21 %	89.91 %	96.56 %
Bottom	98.01 %	99.25 %	99.89 %
Fish school	86.38 %	89.11 %	95.45 %
Total	90.53 %	92.76 %	97.30 %

References

1. Churnside, G., Wilson, J., Tatarskii, V.: Lidar Profiles of Fish Schools. Journal of Applied Optics, Optical Society of America, **36** (1997) 6011-6020
2. Veerabuthiran, S.: Exploring the Atmosphere With Lidars. Journal of Resonance, **8** (2003) 33-43
3. Guenther, G., Eisler, T., Riley, J., Perez, S.: Obstruction Detection and Data Decimation for Airborne Laser Hydrography. Proceedings of Canadian Hydrographic Conference, Halifax, NS, Canada (1996) 51-63
4. Squire, J., Krumboltz, H.: Profiling Pelagic Fish Schools Using Airborne Optical Lasers and other Remote Sensing Techniques. Marine Technology Society Journal, **15** (1981) 443-448
5. Churnside, J., and McGillivary, P.: Optical Properties of Several Pacific Fishes. Journal of Applied Optics, Optical Society of America, **30** (1991) 2925-2927
6. Churnside, J., Wilson, J., Tatarskii, V.: Airborne Lidar for Fisheries Application. Journal of Optical Engineering, **40** (2001) 406-414
7. Mitra, V., Wang, C., Banerjee, S.: Lidar Detection of Underwater Objects Using Neural Networks with Linear Prediction and Fourier Transform for Feature Extraction. Proceedings of Application of Neural Network in Engineering, **13** (2003) 695-700
8. Atal, B., Schroeder, M.: Predictive Coding of Speech Signals and Subjective Error Criteria. IEEE Trans. on Acoustics, Speech, and Signal Processing, **27** (1979) 247–254

Ultra-wideband Nearfield Adaptive Beamforming Based on a RBF Neural Network

Min Wang[1], Shuyuan Yang[2], and Shunjun Wu[1]

[1] National Lab. of Radar Signal Processing, Xidian University
Xi'an, Shaanxi 710071 China
wmgo@263.net

[2] Institute of Intelligence Information Processing, Department of Electrical Engineering
Xidian University, Xi'an, Shaanxi 710071 China
syyang@xidian.edu.cn

Abstract. An adaptive beamforming method based on radial-basis function (RBF) neural network is examined for ultra-wideband (UWB) array illuminated by nearfield source in this paper. An analysis of the principle of space-time processing employing Gaussian monocycle model as UWB signal is conducted. The nearfield regionally constrain of UWB beamformer is reflected by a set of samples exerted on neural network training sample space. The recursive least square algorithm has been used for network weights updating. It improves the robustness against large errors in distance and directions of arrival. The efficiency and feasibility of presented approach is proved through the experimental results.

1 Introduction

In the past years, the array antenna technology had mainly used in the narrowband and wideband system. Now with the development of emerging ultrawide-band (UWB) pulse technology in impulse radar and impulse-radio communication, the principle of space-time processing can be also applied in the design and research of the UWB system [1],[2],[3],[4].

The common UWB beamforming method is to compensate time delay using a time delay-sum beamformer in UWB system. But time-delay in channels cannot be estimated accurately and compensated in easy. Another scheme is of optical realization by true time delay beamforming in UWB pulse array. But it is a so complicated system. A UWB array has so wide range of frequency band, and its response varies with frequency obviously. A large number of sensors may be needed under general broadband solution. And with the increase of the frequency range, the number of sensors also increases. Moreover, many application occasions of UWB pulse array cannot satisfy an ordinary farfield condition, such as in some scenarios of medical diagnosis appliances [3]. It is well known that dynamically adaptive beamformers can achieve better performance than fixed-weight beamformers when noise and interference are time-varying or location are inaccurate. The goal of this paper is to consider the adaptive realization of nearfield beamforming in UWB pulse array.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 562–567, 2005.

In this paper, a radial basis function neural network (RBFNN) method for the beamforming of nearfield UWB pulse array is presented. The construction of the network and its learning algorithm are depicted in detail. For the fine property of neural network (NN) such as its large capacity, parallelism, nonlinear mapping and self-learning, RBFNN can fulfill a rapid implementation of beamforming and is more robust to the environment disturbance. Experimental results obtained with uniform linear array receiving UWB pulse train signal help to assess the usefulness of the proposed method.

2 Principle of Spatio-time Processing of Nearfield UWB Pulse Array

The commonly used UWB signal model is Gaussian impulse waveform, which is of high resolution and penetration for its very short duration. We give a representation of generalized Gaussian pulse (GGP) as follow:

$$\Omega(t)=\frac{E_0}{1-\alpha}\left\{e^{-4\pi\left(\frac{t-t_0}{\Delta T}\right)^2}-\alpha e^{-4\pi\alpha^2\left(\frac{t-t_0}{\Delta T}\right)^2}\right\}. \tag{1}$$

Here E_0 is the peak amplitude at the time $t=t_0$ (usually $E_0=1$), $\triangle T$ is a nominal duration, and α is a scaling parameter.

Paper [2] gives a structure of UWB (carrier free) array beamformer. The beamforming system consists of a linear array with omnidirectional sensors uniformly spaced, which are grouped into two subarrays with same size. Each channel of beamformer firstly carries out temporal correlation processing equivalent to matched filter for received samples. An adjustable digital delay line or a transverse digital filter (FIR) is employed to get an accurate compensation of time delay and contributes to the beamformer in the desired direction. In the following, we consider the nearfield case of UWB signal impinging on the array and the robustness of the beamformer against location errors.

Given the radical distance r_m , azimuth θ_m in planar coordinate system, consider a broadband array beamformer with M elements and K taps attached at each element. The elements of the array are located at $\{\mathbf{x}_m=(r_m,\theta_m)\}$, $m=1,2,\dots,M$. Without loss of generality, the coordinate system is defined such that its origin is at the phase center of the array. If the signal target is located at $\mathbf{x}_s=(r_s,\theta_s)$ with $r_s<R_a^2/\lambda$, where R_a^2 is the largest array dimension and λ is the operating wavelength, the near-field propagation model is required and the near-field steering vector of the array beamformer is defined as [5]:

$$\mathbf{a}(\mathbf{x}_s,f)=\frac{r_s}{e^{j2\pi f r_s/c}}\left[\frac{e^{j2\pi f r_{1s}/c}}{r_{1s}},\dots,\frac{e^{j2\pi f(r_{ms}/c-k)}}{r_{ms}},\dots,\frac{e^{j2\pi f(r_{Ms}/c-K+1)}}{r_{Ms}}\right]^T. \tag{2}$$

where the superscript $(\cdot)^T$ represents transpose, and f is the frequency, c is the propagation speed, $r_s=|\mathbf{x}_s|$ and $r_{ms}=|\mathbf{x}_m-\mathbf{x}_s|$ are the distances from the signal source to the phase center of the array and *m*th the element, respectively. We study its principle of the UWB pulse array in paper [2] in frequency domain. Let the expression of input vector of the beamformer after Fourier transform be $\mathbf{u}(f)$ formed by J frequency bins, the corresponding array output be $\mathbf{y}(f)$. Then we get the following equation:

$$\mathbf{u}(f)=\mathbf{a}(\mathbf{x}_s,f)\mathbf{s}(f)\,. \tag{3}$$

$$\mathbf{y}(f)=\mathbf{G}(f)\mathbf{u}(f)\,. \tag{4}$$

$$\mathbf{G}(f)=[g_1^*(f)\;g_2^*(f)...g_M^*(f)]^H\,. \tag{5}$$

Here $g_m(f)$ is the response of each element channel. Accordingly, we get the frequency response function of UWB impulse array:

$$\mathbf{b}(f,\mathbf{x}_s)=\mathbf{G}^H(f)\mathbf{a}(f,\mathbf{x}_s)\,. \tag{6}$$

$$\mathbf{y}(t,\mathbf{x}_s)=\mathbf{s}(t)\otimes\mathbf{b}(t,\mathbf{x}_s)\,. \tag{7}$$

$$\mathbf{b}(t,\mathbf{x}_s)=\int_{-\infty}^{\infty}\mathbf{b}(f,\mathbf{x}_s)e^{j2\pi ft}df\,. \tag{8}$$

Taking each channel frequency response as the row vector of $\mathbf{G}(f)$ with size $M\times J$, each column of matrix $\mathbf{G}(f)$ is corresponding to the complex weight at specified frequency of UWB pulse array. The desired response $\mathbf{b}(f,\mathbf{x})$ we want to design is given as the response corresponding to the specified UWB sources $\{\mathbf{x}\}$ over a certain region in nearfield. Using the LCMV method, the nearfield adaptive beamformer tries to minimize the output power subject to some constraints. If the point number L of $\mathbf{b}(f,\mathbf{x})$ is less than free degrees $N=MJ$ of matrix $\mathbf{G}(\omega)$, equation (6) can be treated as the L linear constraints. Assume $\mathbf{R}_{\mathbf{uu}}$ denotes the covariance matrix of input vector and $\mathbf{A}$ is steering matrix, the optimal solution to the constrained minimization problem is obtained by:

$$\mathbf{G}_{opt}=\mathbf{R}_{\mathbf{uu}}^{-1}\mathbf{A}(\mathbf{A}^H\mathbf{R}_{\mathbf{uu}}^{-1}\mathbf{A})^{-1}\mathbf{b}\,. \tag{9}$$

In following, we will develop a neural networks method for UWB signal beamforming over a near-field spatial region.

3 Adaptive Beamforming by RBF Network

Paper [6] shows analogy of spatio-temporal processing between the biological neuron and digital spatio-temporal neural network system. The complex-valued spatio-temporal multiplayer neural network is utilized for ultra-wideband beamforming application. Elemental unit of the spatio-temporal digital neuron is a TDL/FIR (time-delay-line/Finite Impulse response).

Radical Basis Function (RBF) network is derived form regular theory and has the optimal approximation ability for complicated functions [7],[8]. It has a faster learning speed compared to global methods, such as the MLP with BP rule, and only part of the input space needs to be trained. When a new sample is fed to the network, the network modifies weights only in the vicinity of the sample point, and maintains constant weights in the other regions. The RBFNN is insensitive to the order of the appearance of the adjusted signals and hence more suitable for online for sequent adaptive adjustment.

The nonlinear relation at the nodes of the second layer is a radial-basis function mapping the input to a high dimension with nonlinear active functions. The first layer takes an input, and the output layer is linear. Assume the number of nodes in the input layer, hidden layer and output layer are *M*, *J* and *P* respectively.

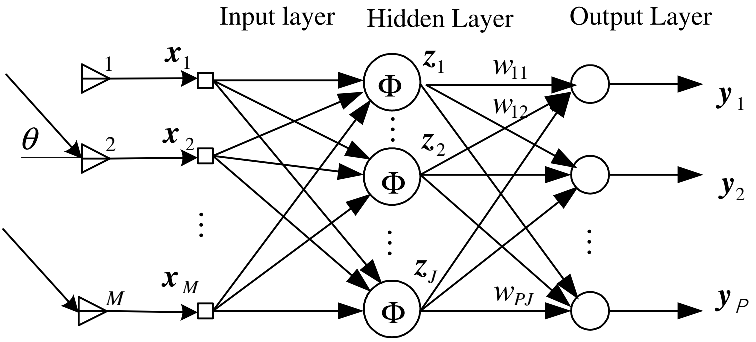


Fig. 1. The architecture of RBFNN as a beamformer realization for UWB array.

The output of the *j*-th node for the *n*-th sample of the hidden layer is:

$$z_{jp} = \Phi(\|X_n - C_j\|) = \exp[-\sum_{i=1}^{M}(x_i - c_{ji})^2 / \sigma_{ji}{}^2]. \tag{10}$$

C_j and σ_{ji} are the center and variance of the radical function respectively. Then a linear network follows:

$$y_{ip} = W_{i0} + \sum_j W_{ij}\Phi(\|X_n - C_j\|). \tag{11}$$

where y_{ip} is the output of the *i*-th node for the *n*-th sample in the output layer, W_{ij} is the weight of the *i*-th output node from the *j*-th hidden node.

The antenna array serves as nonlinear mapping from signal sources to antenna measurement $G: R^P \to C^M$ while beamforming is the inversion problem performing the inverse mapping $F: C^M \to R^P$ .Before the network begins to work, a training process is performed. Assume the incidence angle varies from -90^0 to $+90^0$, and the training data are obtained by uniformly sampling in angle field at spaced s^0 . Input the training samples into the network, and we adopt such a learning algorithm: firstly a self-organized feather maps (SOFM) clustering method is used to select the centers of the basis function, and then the radius of neurons are determined by $\sigma = D/\sqrt{2J}$, where *D* is the maximum distance of these selected centers. For the weight of output layer, a recurrent least square (RLS) algorithm is used.

The training process is as follow:

1. Evaluate the nearfield steering vector over the interested nearfield region by (2).
2. Specify the desired response over the interested nearfield region.

3. Generate the training set { $\mathbf{a}(\mathbf{x}_s, f)$, $\mathbf{b}(f, \mathbf{x}_s)$ } using given desired response.
4. Apply SOFM clustering to the input set to obtain the centers of the hidden neurons; use a RLS learning algorithm to learn the optimal weights for the output layer.

4 Experiments and Preliminary Results

We demonstrate the feasibility of our constructed RBF neural network in UWB array beamforming by applying it to a uniform linear array with element number M=11. The UWB signal being used in the experiments is a monocycle of GGP with nominal duration time $\Delta T = 2ns$, sample period T_s = $100ps$. The spatio-time output of the beamformer is depicted in Fig.2. The source is located at $r = 10\lambda_U$, $\theta = 0$.

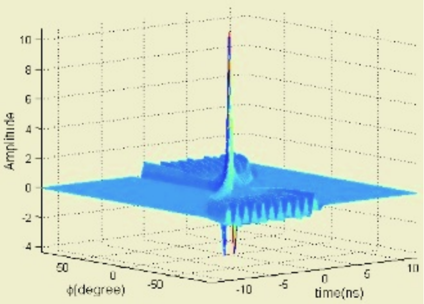


Fig. 2. The spatio-time output of UWB pulse array

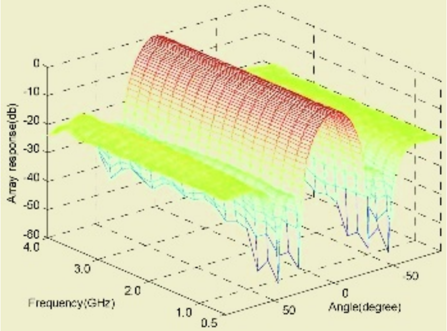


Fig. 3. The Distortionless Nearfield UWB beampattern in certain bandwidth using RBFNN

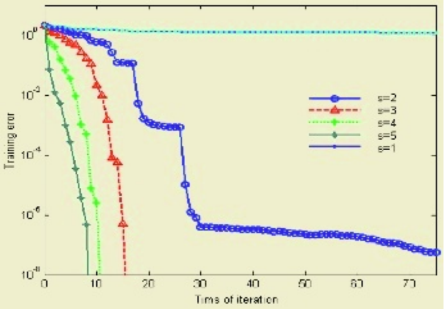


Fig. 4. Convergence curves for different training samples

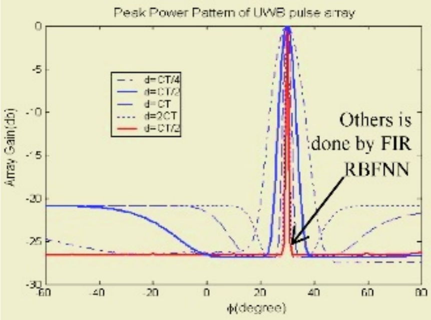


Fig. 5. The peak power directivity pattern of using FIR and RBF NN

The system is trained to steer its beams toward desired UWB signals coming from the angle 30^0. The snap number is 200. We compare the performance of RBFNN-based beamformer with that of FIR filters. In the RBF, the training samples of the network are obtained by a uniform sampling of θ from -90°to + 90°spaced s^o . From Fig.3, the frequency-angle characteristic of the Nearfield UWB array output using RBFNN approximates the given response. Fig.4 gives the variation of error with the iteration times in different *s*, from which we can see that more training samples we use, more fast of the convergence of the network. That is, a smaller *s* will lead to a rapider convergence. In the test a maximum iteration times 100 and *s*=10 is adopted. It means only 19 samples are used for learning.

Fig.5 shows the energy directivity pattern employing FIR filters with different interelement spacing $d=C\triangle T/4$, $d=C\triangle T/2$, $d=C\triangle T$, $d=2C\triangle T$ and the trained RBF network with $d=C\triangle T/2$, which are plotted in blue and red respectively.

5 Conclusions

The paper presents a RBF neural network based method to design adaptive UWB pulse array beamformer in the case of nearfield. After a discussion on UWB pulse array processing principle in time domain and frequency domain, we consider the adaptive beamforming by employing RBF neural network. The method presented in the paper is characterized of low complexity, rapid convergence and robust output. The simulation results show the feasibility of this method.

References

1. Fontana, R.J.: Recent System Applications of Short-Pulse Ultra-Wideband (UWB) Technology. IEEE Transactions on Microwave Theory and Techniques, **52** (2004) 2087-2104
2. Malek, G.M., Hussain: Principles of Space-Time Array Processing for Ultrawide-Band Impulse Radar and Radio Communications. IEEE Trans. on vehicular technology, **51** (2002) 393-403
3. Converse, M., Bond, E.J., Hagness, S.C.: Ultrawide-band Microwave Space-time Beamforming for Hyperthermia Treatment of Breast Cancer: A Computational Feasibility Study. IEEE Trans. on Microwave Theory and Techniques, **52** (2004) 1876 -1889
4. Anderson, F., Christensen, W., Fullerton, L.: Ultra-wideband Beamforming in Sparse Arrays. Microwaves, Antennas and Propagation, IEE Proceedings, **138** (1991) 342-346
5. Zheng, R.Y., Gourbran, R.A.: Robust Near-field Adaptive Beamforming with Distance Discrimination. IEEE Trans. on ASSP, **12** (2004) 478-488
6. Suksmono, A.B., Hirose, A.: Ultra-wideband Beamforming by Using a Complex-Valued Spatio-temporal Neural Network. ICONIP 2004, LNCS **3316** (2004)104-109
7. Park, J., Sandberg, I. W.: Universal Approximation Using Radial-basis-function Networks. Neural Computation, **3** (1991) 246-257
8. Jiao, J.: Theory of Neural Networks System. Xi'an, China, Xidian University Press. (1991)

Automatic Digital Modulation Recognition Using Support Vector Machines and Genetic Algorithm

Jie Li[1], Jing Peng[2], Heng Chu[1], and Weile Zhu[1]

[1] School of Electronic Engineering, University of Electronic Science and Technology of China
Cheng du, Sichuan 610054, China
xnkjlj@163.com, xiaoyong76@163.net, wlzhu@uestc.edu.cn
[2] School of Network Technology, Hebei University of Science and Technology
Shijiazhuang, Hebei 050061, China
pengjingpost@163.com

Abstract. A new method, based on support vector machines (SVMs) and genetic algorithm (GA), is proposed for automatic digital modulation recognition (ADMR). In particular, the best feature subset from the combined statistical feature set and spectral feature set is optimized using genetic algorithm. Compared to the conventional artificial neural network (ANN) method, the method proposed avoids overfitting and local optimal problems. Simulation results show that this method is more robust and effective than other existing approaches, particularly at a low signal noise ratio (SNR).

1 Introduction

ADMR is an important issue in both multimode smart radio systems and military communication systems. It is one kind of pattern recognition problems and there are different methods proposed for this subject. Decision theoretic approaches use probabilistic and hypothesis testing arguments to formulate the recognition problem but they have difficulties in forming the right hypothesis and setting the right threshold [1]. When training data are available, the artificial neural network approaches [1],[2] and other statistical recognition methods have been proposed. The ANN methods have gained popularity in the past decade but they couldn't avoid the problems of overfitting, local minimization, etc [4].

This paper introduces a new ADMR method using SVMs and GA. 11 digital modulation types are considered in the paper: ASK2, ASK4, FSK2, FSK4, BPSK, QPSK, PSK8, MSK, QAM4, QAM8 and QAM16. Simulation results show that this method is more flexible, robust and effective than other existing approaches.

2 Problem Statement and Feature Extraction

To reduce the size of raw data set, most ADMR systems often extract some distinct attributes called features before recognizing. In this paper, we use a new combined feature set composed of a statistical feature subset and a spectral feature subset.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 568–573, 2005.

2.1 Statistical Feature Set

Supposing that we operate in a synchronous, coherent environment with single-tone signaling and that timing, carrier, and waveform recovery have been accomplished. After preprocessing, we obtain the baseband complex sequence that can be written as:

$$y(n) = x(n) + w(n) \tag{1}$$

where x(n) is supposed to be a digitally modulated signal and w(n) is an additive Gaussian noise sequence. x(n) and w(n) are independently and identically distributed (i.i.d.) and jointly uncorrelated stochastic sequences.

High order statistics (HOS) are used as a feature subset in our scheme because they characterize the shape of the distribution of the noisy baseband samples effectively [3]. Let y(n) denote received complex baseband signal samples and its 2nd cumulants, 4th cumulants can be estimated from the samples by the process given below.

① Remove the mean of y(n).

② Compute sample estimates of C_{20}, C_{21} by the following formulae:

$$\begin{aligned} \hat{C}_{20} &= \frac{1}{N}\sum_{n=1}^{N} y^2(n) \\ \hat{C}_{21} &= \frac{1}{N}\sum_{n=1}^{N} | y(n) |^2 \end{aligned} \tag{2}$$

where N is the length of samples and ^ denotes a sample average.

③ Sample estimates of the 4th cumulants are given by the following formulae.

$$\begin{aligned} \hat{C}_{40} &= \frac{1}{N}\sum_{n=1}^{N} y^4(n) - 3\hat{C}_{20}^{\,2} \\ \hat{C}_{41} &= \frac{1}{N}\sum_{n=1}^{N} y^3(n) y^*(n) - 3\hat{C}_{20}\hat{C}_{21} \\ \hat{C}_{42} &= \frac{1}{N}\sum_{n=1}^{N} | y(n) |^4 - | \hat{C}_{20} |^2 - 2\hat{C}_{21}^{\,2} \end{aligned} \tag{3}$$

④ Compute the normalized cumulants.

$$\overset{\neg}{C}_{4k} = \hat{C}_{4k} / \hat{C}_{21}^{\,2}, \text{ k=0, 1, 2} \tag{4}$$

2.2 Spectral Feature Set

Azzouz and Nandi [1] proposed a suitable spectral-based feature set for ADMR, which contains hidden information in instantaneous amplitude, instantaneous frequency, or instantaneous phase. In this study, it is used as a feature subset for modulation classification. The five features proposed are described as below:

- Maximum value of the power spectral density of the normalized centred instantaneous amplitude: γ_{max}.
- Standard deviation of the absolute value of the normalized centred instantaneous amplitude: $\sigma_{a\sigma}$.
- Standard deviation of the absolute value of the centered nonlinear components of the instantaneous phase: σ_{ap}.
- Standard deviation of the direct value of the centered nonlinear components of the instantaneous phase: σ_{dp}.
- Standard deviation of the absolute value of the normalized centred instantaneous frequency: σ_{af}.

3 Digital Modulation Recognition Using SVMs and GA

The proposed ADMR scheme is composed of four parts: signal data preprocessing, feature extraction, GA feature selection and SVMs classification. Firstly, we discuss the construction of multiclass SVMs classifier in detail. Then, the digital modulation recognition algorithm is introduced.

3.1 Multiclass Support Vector Machines Based on Error-Correcting Output Codes

Support vector machines are originally designed for binary classification [4]. How to effectively extend it for multiclass classification is still an ongoing research issue [5]. The way we construct the multiclass classifier is by combining several binary SVMs, according to an error-correcting output codes (ECOC) scheme [5].

ECOC have been proposed to enhance generalization ability in pattern classification. Coding the classes using codewords suggests the idea of adding error recovering capabilities to decomposition methods, which makes classifiers are less sensitive to noise. This goal is achieved by means of the redundancy of the coding scheme, as shown by coding theory. By class coding scheme, we can implicitly decompose the k-polychotomy into a set of dichotomies f_1, ..., f_B, where B is the length of the codeword coding a class.

First, a collection of binary classifiers $\{f_1, ..., f_B\}$are constructed and each classifier f_b is trained according to column b of decomposition matrix G[5], and $g_{ij} \in G$ is the target value of the jth decision function $D_j(x)$ for class i: where

$$g_{ij}(x) = \{ \begin{matrix} 1 & if\ D_j(x) > 0\ for\,class\,i, \\ -1 & otherwise. \end{matrix} \tag{5}$$

The jth column vector $g_j=(g_{1j},...,g_{nj})^T$ is the target vector for the jth decision function, where n is the appropriate class label. The ith row vector $(g_{i1},...,g_{iB})$ corresponds to a codeword for class i.

Then, we can find the class label of x by calculating its distance from each codeword. Introducing ”don’t care output” and denote its value by 0, we can define an error $\varepsilon_{ij}(x)$ by

$$\varepsilon_{ij}(x) = \begin{cases} 0 & for \quad g_{ij} = 0, \\ \max(1 - g_{ij} D_j(x), 0) & otherwise \end{cases} \tag{6}$$

where $g_{ij}D_j(x) \geq 1$ denotes that x is on the correct side of the jth decision function with more than or equal to the maximum margin and $g_{ij}D_j(x)<1$ denotes that x is on the wrong side or even if it is on the correct side but the margin is smaller than the maximum margin.

Then the distance of x from class i is given by

$$d_i(x) = \sum_{j=1}^{B} \varepsilon_{ij}(x). \tag{7}$$

Thus x is classified into class:

$$\arg \min_{i=1,\ldots,n} d_i(x) \tag{8}$$

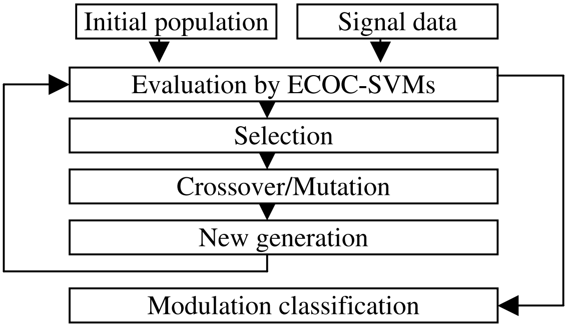


Fig. 1. ECOC-SVMs classification with GA subset selection

3.2 GA ECOC-SVMs Modulation Recognition Algorithm

Typical pattern recognition applications use feature selection to reduce input dimension and remove redundant input information, which implies a decrease in classification accuracy. In our ADMR application, GA is used to select the most suitable features for the particular recognizers in different SNR environment. Individuals represent subset of features by means of binary strings. Each string has a length L, which is the total number of input features we extract in preprocessing step for ADMR. In the genome string, a binary ’1’ denotes the presence of the appropriate feature at the corresponding index number and a binary ’0’ denotes an absence. The proposed ADMR algorithm can be shown in figure 1.

4 Simulation Results

In this section, a variety of experiments are presented to illustrate the performance of the proposed recognition scheme. All the simulation signals are digitally generated in MATLAB environment. Random integers up to M-level (M=2,4,8,16) are generated by a uniform random number generator. The simulation signals are also band-limited and additive white Gaussian noise is added according to different SNR.

Example 1: We use the BCH codes as ECOC codes and four BCH codes with 15,31,63, and 127 word lengths are used to properly set the minimum Hamming distances. First, we study the influence of BCH codes length on the performance of ECOC-SVMs classifier. Table 1 shows that the recognition rates of test data improved as the word length is increased and they reach the maximum at the word length of 63. This may because Hamming distance unclassifiable regions decrease as the codes length is increased.

Table 1. Recognition rates (%) of ECOC-SVMs with different code length, RBF kernels(γ= 1)

SNR (dB)	(15,7)	(31,15)	(63,31)	(127,55)
-5	86.25	88.33	90.51	89.50
0	94.16	97.40	98.48	97.42
5	96.59	97.84	98.91	98.05
10	97.37	98.05	99.13	98.27
15	98.23	98.48	99.57	99.13
20	99.32	99.78	99.99	99.57

Example 2: Secondly, we study the performance of ECOC-SVMs classifier with GA based feature selection in different SNR environment. In this trial, mutation rate and crossover rate are respectively set to 0.1 and 0.95. Overall classification rates are used as the fitness function and the roulette wheel method is used as the selection function. Performance of binary string representation GA ECOC-SVMs with dynamic feature selection is shown in table 2. Generally, simulation results from reduced feature set selected at different SNR show significant improvements over the performance the ECOC-SVMs classifier. Moreover, feature subset contains typically four or five features less than the original feature set.

Table 2. Performance of ECOC-SVMs with GA dynamic feature selection, RBF kernels (γ= 0.65)

SNR (dB)	Binary string	Feature selection	Recognition rates (%)
-5	11011011011111	11	91.99
0	11110100100111	9	98.48
5	11010101000111	8	98.91
10	11011010010101	8	99.13
15	10101011010111	9	99.78
20	10011011011101	9	99.99

Example 3: At last, we give an overall performance comparison on the decision-theoretic algorithm proposed in [3], the ANN method proposed in [1] and the pro-

posed ECOC-SVMs method. Simulation results are shown in table 3. As shown in the table, both the ANN method and the decision theoretic algorithm don't perform well at a low SNR, for instance, -5dB. The ANN methods' shortcoming is that it doesn't work efficiently in a low SNR environment and it also needs a long training time. On the other hand, both the methods proposed in [1] and [3] can only recognize fewer modulation modes compared to the proposed ECOC-SVMs. The reason that the proposed GA ECOC-SVMs method is both robust and effective is not only the using of a better classifier structure but also the adoption of the optimized feature set.

Table 3. Overall performance comparison on the decision theoretic algorithm, the ANN method and the proposed ECOC-SVMs method

SNR (dB)	Decision theoretic	ANN	GA ECOC-SVMs
	Overall perf. (%)	Overall perf. (%)	Overall perf. (%)
-5	48.85	63.62	91.99
0	56.34	83.02	98.48
5	81.12	92.26	98.91
10	96.75	97.40	99.13
15	98.56	98.32	99.78
20	99.32	99.50	99.99

5 Conclusion

In this paper, a new automatic digital modulation recognition method using ECOC-SVMs and GA is introduced. A new feature set combined statistical and spectral feature subset is used for modulation classification to make the SVMs classifier more robust in Gaussian noise environment. Moreover, GA is used to perform feature selection to reduce the input dimension and increase performance of the ECOC-SVMs classifier. Compared to the conventional ANN method and the decision theoretic algorithm, the proposed method can recognize more digital modulation types. Furthermore, significant improvements can be seen particularly at a low SNR.

References

1. Nandi, A.K., Azzouz, E.E.: Modulation Recognition Using Artificial Networks. IEEE Trans. Signal processing, **56** (1997) 165-175
2. Nandi, A.K., Azzouz, E.E.: Algorithms for Automatic Modulation Recognition of Communication Signals. IEEE Trans. Communications, **46** (1998) 431-435
3. Swami, A., Sadler, B.M.: Hierarchical Digital Modulation Classification Using Cumulants. IEEE Trans. Communications, **48** (2000) 416-429
4. Vapnik, V.: An Overview of Statistical Learning Theory. IEEE Trans. Neural Networks, **10** (1999) 988-999
5. Klautau, A., Jevtic, N., Orlitsky, A.: On Nearest-Neighbor Error-Correcting Output Codes with Application to All-Pairs Multiclass Support Vector Machines, Journal of Machine Learning Research, **4** (2003) 1-15

A Unified Framework for Synthesis of Cosine-Modulated Filter Banks and Corresponding Wavelets*

Ying Tan

School of Information Science and Technology
University of Science and Technology of China, Hefei 230027, China
ytan@ustc.edu.cn

Abstract. By directly optimizing the impulse response coefficient of filter banks, the synthesis of cosine-modulated FIR filter banks and cosine-modulated wavelets can be carried out in a unified framework. Through proper transformation, the problem is formulated as a quadratic-constrained least-squares (QCLS) minimization problem where all constraint matrices are symmetric and positive definite. Furthermore, we create an analog neural network by the augmented cost of the QCLS problem for designing filter banks and wavelets in real time. It turns out that the analysis and synthesis filters with high stopband attenuation and compactly supported wavelets are easily obtained by this method. Computer simulations illustrates the efficiency and effectiveness of our method.

1 Introduction

Recently, the cosine-modulated filter bank (CMFB) with perfect reconstruction (PR) character in the field of multirate signal processing has emerged as an attractive choice of filter bank with respect to implementation cost and design saving. It is shown [1],[2] that the 2M polyphase components of the prototype filter can be grouped into M power-complementary pairs, where each pair is implemented as a two-channel lossless lattice filter bank. The lattice coefficients are optimized to minimize the stopband attenuation of the prototype filter, but this is a highly nonlinear optimization problem with respect to lattice coefficients. Consequently, it is difficult to obtain the PR CMFB with high stopband attenuation. Recently, several authors [3], [4-9] formulate the design problem as a quadratic-constrained least squares (QCLS) problem and obtain the CMFB with high stopband attenuation. In this paper, we convert the QCLS problem in [3], [4] into a very simple format whose constraint matrices are of symmetric and positive definite properties, and then recast them into an analog neural network framework. Such that, we can not only obtain cosine-modulated filter banks with high stopband attenuation but also realize real-time filter bank design.

* This work was supported by the Natural Science Foundation of China with grant no. 60273100, and partially supported by the Distinguished Youth Scientific and Technological Foundation of An Hui Province (2002-2003).

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 574–579, 2005.

It is shown in [5] that filter banks satisfying regularity conditions can be used to generate orthonormal bases of compactly supported wavelets. These regularity conditions can also be translated into some additional constraints imposed on the frequency response of the lowpass filter of the CMFB so as to design cosine-modulated compactly supported wavelets.

2 Quadratic-Constraint Formulations

2.1 Quadratic-Constraint Formulation for Paraunitary Cosine-Modulated FB

The impulse responses of the analysis and synthesis filters $h_k(n)$ and $f_k(n)$ of the cosine-modulated filter bank are cosine-modulated versions of a prototype filter $p(n)$, which are given as

$$h_k(n)=2p(n)\cos\left((2k+1)\tfrac{\pi}{2M}\left(n-\tfrac{N-1}{2}\right)+(-1)^k\tfrac{\pi}{4}\right) \tag{1a}$$

$$f_k(n)=h_k(N-1-n) \tag{1b}$$

where $0\le n\le N-1$ and $0\le k\le M-1$.

Here, the lengths of filters $H_k(z)$ and $F_k(z)$ are the same and are assumed to be multiples of 2M, i.e., $N=2mM$.

A simple filter design method is directly optimising the impulse response $p_0(n)$ of FB. For this purpose, the PR conditions can be written as a set of quadratic-constraint

$$\begin{cases} p_{\frac{1}{2}}^T p_{\frac{1}{2}} = 1/2 \\ p_{\frac{1}{2}}^T C_{k,r}\, p_{\frac{1}{2}} = 1 \end{cases} \tag{2}$$

where $p_{\frac{1}{2}}=[p_0(0),\ p_0(1),\ \ldots,\ p_0(mM-1)]^T$, and $C_{k,r}$ is a group of symmetrical and positive definite matrices . Note that *k* goes from 0 to $M/2-1$ and *r* is in the range of 0 to *m-1*, so there are $mM/2+1$ constraints in (2).

Because of limited space, we omit the detailed derivations of Eq.(2). Interesting readers can refer to our related materials [4], [8], [9].

Since the system is almost approximate perfect reconstruction in this case, according to related PR theory of multirate filter bank it is sufficient to only minimize the stopband energy of the prototype filter under the quadratic-constraints of Eq.(2) in even M case. The cost of stopband energy of the prototype filter $\phi=\int_{(\pi/2M)+\varepsilon}^{\pi}\left(P_0(e^{j\omega})\right)^2 d\omega$ can be further simplified as

$$\phi = p_{\frac{1}{2}}^T Q\, p_{\frac{1}{2}} \tag{3}$$

where **Q** is a real, symmetric, and positive-definite matrix with entries as $[Q]_{i,j}=4\int_{(\pi/2M)+\varepsilon}^{\pi}\cos(i\omega-N/2)\cos(j\omega-N/2)d\omega$ in which ε denotes the bandwidth of transition band of prototype filter.

2.2 Quadratic-Constraint Formulation for Compactly Supported Orthonormal Cosine-Modulated Wavelet

It is well-known that filter banks which satisfy regularity conditions can be used to generate orthonormal bases of compactly supported wavelets. Here, we want to obtain a cosine modulated PR filter bank whose lowpass filter $H_0(z)$ is maximally regular. It has been shown [5] that a Mth-band orthonormal wavelet can be obtained from a M-channel PR filter bank by using an infinite tree structure. Moreover, the Mth-band wavelet would have L vanishing moments if and only if the function $H_0\left(e^{j\omega}\right)$ has zeros of order L at frequencies $\omega_l = 2l\pi / M, \quad 1 \le l \le M-1$ and $H_0(1) = \sqrt{M}$.

Two above conditions can be expressed as a quadratic formation of coefficients vector $p_{\frac{1}{2}}$ and finally give a group of equivalent regularity conditions as follows

$$\begin{cases} p_{\frac{1}{2}}^T R_0 p_{\frac{1}{2}} = 1 \\ p_{\frac{1}{2}}^T R_{kl} p_{\frac{1}{2}} = 1 \end{cases} \tag{4}$$

where $R_0 = W_0$ and $R_{kl} = \gamma \mathrm{Re}\left(W_{kl}\right) + W_0$. Here, the choice of the scalar γ must guarantee R_{kl} being a set of absolute diagonally dominant matrices.

The above $L(M-1)+1$ additional conditions in (4) ensure that designed cosine modulated wavelet has L vanishing moments. Together with the PR conditions in (2), we can design cosine modulated PR wavelet with specified vanishing moments.

2.3 United Quadratic-Constraint Formulation for PR CMFB and Wavelets

From above discussion, we have known that both the PR conditions of cosine-modulated filter banks and regularity conditions of its lowpass analysis filter can be converted into a set of same formation of quadratic-constraints whose constraint matrices are symmetric and positive definite as in (2) and (4). So the design problem of both CMFB and its corresponding wavelets can be unified as a unification method which can be expressed as a kind of QCLS minimization problem subject to a set of quadratic-constraints, i.e.,

$$h_{opt} = \min_{p_{\frac{1}{2}}} \quad \Phi = p_{\frac{1}{2}}^T Q p_{\frac{1}{2}} \quad s.t. \quad \begin{cases} Eq.(2) \\ Eq.(4) \end{cases} \tag{5}$$

3 Neural Network for Designing FB and Wavelets

According to (5), we can build the cost of the constrained optimization problem and map it as the energy function of an analog neural network

$$J(X, \beta) = X^t \mathbf{Q} X + \sum_{i=1}^{l} \beta_i p_i(X) \tag{6}$$

where β_i is a positive constant, $p_i(X)$is a penalty function satisfy these conditions: (a) it is a continuous and differentiable function of unknown column vector X which corresponding the vector $\mathbf{P}_{\frac{1}{2}}$ in (4), (b) $p_i(X)\geq 0,\ \forall X$. (c) $p_i(X)=0$ *iff* $X^t\mathbf{A}_i X=1$ such as $p_i(X)=(X^tA_iX-1)^2$ where $\mathbf{A}_i$ is called as a constraint matrix.

Obviously, the energy function defined in (6) is just the cost function of QCLS minimization problem using penalty function method. According to penalty function theory, the minimizer of (6) is the exact solution of original QCLS optimization problem in (5) when penalty factor β_i approaches infinity.

3.1 Our Proposed Neural Network

According to the energy function in (6), the neural network dynamics should be such that the time derivative of J is negative. So the motion equation of the *k*th neuron is

$$\frac{du_k(t)}{dt} = -\frac{\partial J}{\partial v_k(t)} \quad v_k(t) = f(u_k(t)) \qquad k = 1, \quad ..., \quad I. \tag{7}$$

where $u_k(t)$ and $v_k(t)$ are input and output of the *k*th neuron at time t, individually, $f(u)$ is a monotonically increasing activation function, and I is number of neurons.

Theorem 1. X is a equilibrium point of J in (6) if and only if $QX = \sum_{i=1}^{l} 2\beta_i(1 - X^tA_iX)A_iX$.

Theorem 2. Suppose $V = V(t,\ V_0)$ is the solution of (6) satisfying $V(0) = V_0 \neq 0$, then $J(V(t,\ V_0))$ is a monotonically decreasing function of time t.

Lemma 1. For $\forall X_1,\quad X_2 \in \Theta$ and $X_1 \neq X_2$, then $X_1^tA_iX_2 = 0,\quad i = 1,2,...,l$.

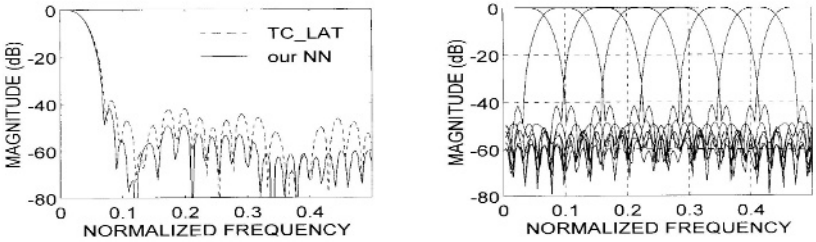
Clearly, the statement of lemma 1 means every two different vectors in feasible region are mutually orthogonal with respect to each constraint matrix A_i.

Theorem 3. $\overline{X}$ is a stable global minimizer of J in (6) if and only if $\overline{X}$ satisfies $(\overline{X})_{A_i} = \max_{X\in\Theta}(X)_{A_i}$, $i=1,\ ...,\ l$, where $(X)_M$ stands for weighted norm of X with respect to matrix M.

In summary, according to theorems 2 and 3, we can find that the neural network of (7) has only unique global stable. Thus, the designs of paraunitary CMFB and wavelets can be in deed carried out by the neural network method without any initialization. The output of the neural network is just the impulse response coefficients of the prototype filter of the filter bank and wavelet to be designed when it is settled down.

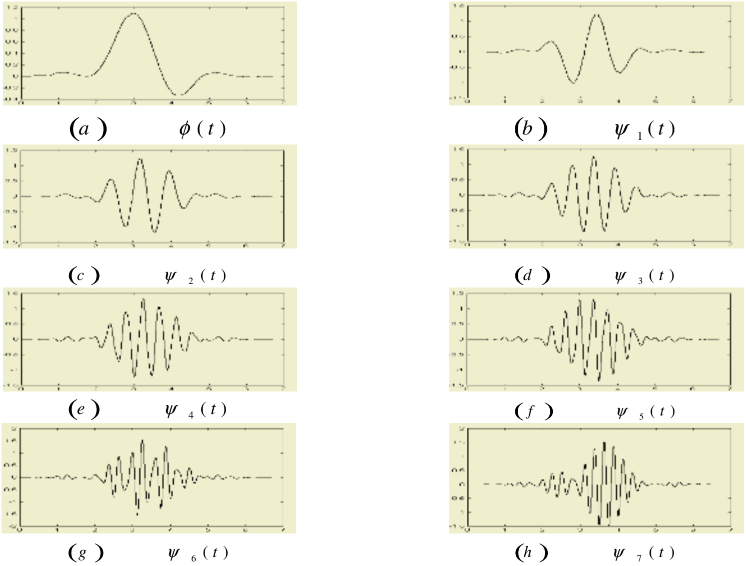
4 Simulations

Example 1. To verify our analyses and derivations, we design an 8-channel PR CMFB by using our proposed neural network when M=8 and m=3 as an example. When we choose the penalty factor be 1000 and initialize network randomly, the designed network can evolve and settle down in magnitude order of circuit constant. The magnitude responses of the prototype filter $P_0(z)$ and analysis FB $H_k(z)$ are shown in Fig.1. For comparison, we also plot an 8-channel PR CMFB designed by conventional two-channel lossless lattice (TC_LAT) parameterized method in Fig.1 (a) in dashed line. It can be seen that the stopband attenuation by our NN design method is about 3dB lower than those by the conventional TC_LAT method.



(a) (b)

Fig. 1. Magnitude responses of the prototype filter (a) and analysis filters (b) of our designed 8-channel CMFB



(*a*) $\phi(t)$

(*b*) $\psi_1(t)$

(*c*) $\psi_2(t)$

(*d*) $\psi_3(t)$

(*e*) $\psi_4(t)$

(*f*) $\psi_5(t)$

(*g*) $\psi_6(t)$

(*h*) $\psi_7(t)$

Fig. 2. Designed scaling function $\phi(t)$ and seven wavelets $\{\psi_k(t), 1 \le k \le 7\}$ by our neural network

Example 2. For M=8 and m=3, we design the compactly-supported orthornormal cosine-modulated wavelets with two vanishing moments. The optimized scaling function and seven wavelet functions are shown in Fig.2.

5 Conclusions

This paper proposed a new unified design method for both paraunitary cosine-modulated FIR filter banks and cosine-modulated compactly supported wavelets. By using neural network to solve the corresponding minimization problem, we obtained cosine modulated filter banks with higher stopband attenuation, and cosine modulated compactly supported wavelets. Two simulations were given to support our theory.

References

1. Koilpillai, R., et. al.: Cosine-modulated FIR Filter Banks Satisfying Perfect Reconstruction. IEEE Trans. Signal Processing, **40** (1992) 770-783
2. Tan, Y., He, Z.: Linear Phase Paraunitary Cosine Modulated Filter Bank Design Formula. 3rd IEEE Int. Conf. on Electronics, Circuit & Systems (ICECS'96), Greece (1996) 633-636
3. Nguyen, T., Koilpillai, R.: The Theory and Design of Arbitrary-length Cosine-modulated Filter Banks and Wavelets, Satisfying Perfect Reconstruction. IEEE Trans. Signal Processing, **44** (1996) 473-483
4. Tan, Y., Gao, X., He, Z.: Design of Perfect Reconstruction Cosine-modulated QMF Banks. Proc. of IEEE ISCAS'97, Hong Kong, (1997) 2441-2445
5. Zou, H., Tewfik, A.: Discrete Orthogonal M-band Wavelets Decompositions. Proc. ICASSP'92
6. Maa, C., Shanblatt, M.: Linear and Quadratic Programming Neural Network Analysis. IEEE Trans. Neural Network, **3** (1992) 580-594
7. Tan, Y., He, Z.: Neural Network Approaches for the Extraction of the Eigenstructure. Neural Networks for Signal Processing. Proc. of the 1996 IEEE Workshop, Japan, (1996) 23-32
8. Tan, Y., Gao, X., He, Z.: Neural Network Design Approach of Cosine-modulated FIR Filter Bank and Compactly Supported Wavelets with Almost PR Property. Signal Processing, **69** (1998) 29-48
9. Tan, Y.: A Neural Network Approach for Signal Detection in Digital Communication. J. of VLSI Signal Processing, **32** (2002) 42-54

A Systematic Chaotic Noise Reduction Method Combining with Neural Network

Min Han[1], Yuhua Liu[1], Jianhui Xi[1,2], and Zhiwei Shi[1]

[1] School of Electronic and Information Engineering
Dalian University of Technology, Dalian, Liaoning 116024, China
minhan@dlut.edu.cn
[2] Department of Automation, Shenyang Institute of Aeronautical Engineering
Shenyang, Liaoning 110034, China

Abstract. It has been found that noise limits the prediction of deterministic chaotic system. Due to the lack of knowledge on dynamical system and nature of noise, the estimate of noise level is obviously important to the commonly used noise reduction method. On the basis of noise level estimate and optimized method, a systematic chaotic noise reduction method is proposed combining with Finite Impulse Response Neural Network (FIRNN) in this paper. Firstly, the initial noise level is estimated using wavelet analysis. Then, a Local Projection noise reduction method is applied while a FIRNN is used as a main diagnostic tool to determine the optimal noise level. Simulation on real monthly noisy sunspot time series shows that the proposed method works properly for noisy chaotic signals.

1 Introduction

Noise reduction, as an integral part of signal estimation, has been studied for many years in different fields. Recent studies show that noise limits the performance of many techniques used for identification and short-time prediction in chaos. If the noise level in data series is moderate, the separation of deterministic signal from the noise could be undertaken [1]. A number of nonlinear noise reduction algorithms have been developed that take into account the nonlinear nature of the data. In general, most dynamical measures are reasonably robust to small noise levels, but as the noise level approaches a few percent, estimates might become quite unreliable [2]. Therefore, the estimation of the level is important.

The Local Projection method [3] is a popular nonlinear method of noise reduction. During the procedure of noise reduction, the effectiveness is generally influenced by the selection of (1) the size of the neighborhood; (2) the number of iterations of the procedure required to achieving optimal noise reduction. Different selections might produce different noise levels removal and would make the data over-correction. Furthermore, the distortion of the signal caused by unsuitable noise reduction method might possibly have far more effects than the noise itself, especially with regard to the short term prediction. Such potential problems need a more effective systematic noise reduction method to select these parameters coupling with a noise level estimation.

The noise level estimate method in [4] often undervalues seriously and the computation is highly complex. In this study, we apply wavelet analysis to estimate the noise

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 580–585, 2005.

level on the basis of multi-resolution in one hand, on the other hand, use Local Projection (LP) method to remove noise. At the same time, considering that a neural network could approximate the chaotic attractors much better for low noise level series [1],[5],[6],[8], the Finite Impulse Response Neural Network (FIRNN)[6] is used to make short-term prediction based on the noise reduced data. The prediction accuracy could be seen as an indicator on noise reduction.

This paper is organized as follows: Chapter 2 shows the architecture and the basic algorithm of FIRNN. The systematic noise reduction method is proposed in Chapter 3. Chapter 4 gives the simulation results of this method and some conclusions are drawn from the studies in the last chapter.

2 Finite Impulse Response Neural Network (FIRNN)

FIRNN resembles a standard feed-forward network where each synapse is replaced by an adaptive FIR linear filter as illustrated in Fig.1, here, q^{-1} represents a time-domain unit delay operator. The FIR filters form a weighted sum of past values of its inputs. The neurons which are arranged in layers receive the filtered inputs and then pass the sum through a nonlinear sigmoid function. Training the network is accomplished through a modification of the *back propagation algorithm* called *temporal algorithm* in which error terms are symmetrically filtered backward through the network as illustrated below: Given an input sequence $x(k)$, k=1, 2, …, n, the network produced the output sequence $y(k) = N[W, x(k)]$,where W represent the set of all filter coefficients in the network. Assuming the desired output $d(k)$ at each instant is provided to the network, the instantaneous error is $e^2(k) = \|d(k) - N[W, x(k)]\|^2$ which is the squared Euclidean distance between the network output and the desired output. The objective cost function to minimize over W is $C = \sum_{k=1}^{n} e^2(k)$ where the sum is taken over all training sequence.

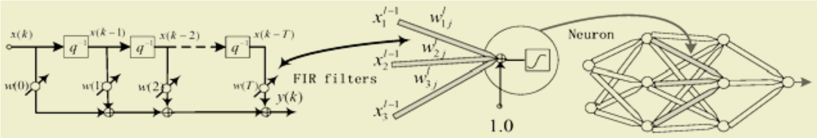


Fig. 1. Structure of FIRNN

Synaptic filters are updated at each increment time according to

$$\boldsymbol{w}_{ij}^{l}(k+1) = \boldsymbol{w}_{ij}^{l}(k) - \eta \delta_j^{l+1}(k) x_i^l(k) \tag{1}$$

Where

$$\delta_j^l(k) = \begin{cases} -2e_j(k) f'(s_j^L(k)) & l = L \\ f'(s_j^l(k)). \sum_{m=1}^{N_{l+1}} \boldsymbol{\delta}_m^{l+1}(k).\boldsymbol{w}_{jm}^{i} & 1 \le l \le L-1 \end{cases} \tag{2}$$

Here, $e_j(k)$ is the error of output node, $f\,'()$ is the derivation of sigmoid function,

$$\delta_m^l(k) \equiv [\delta_m^l(k)\delta_m^l(k+1)\cdots\delta_m^l(k+T^{l-1})] \tag{3}$$

is a vector of propagated gradient terms. To calculate $\delta_j^l(k)$ for a given neuron we filter the δ s from the next layer backwards through FIR synapses to which the given neuron feed. Temporal back propagation preserves the symmetry between the forward propagation of states and the backward propagation of error terms. Parallel distributed processing is maintained. Furthermore, the number of operations per iteration only grows linearly with the number of layers and synapses in the work [6]. Once the network is trained by $y(k-1)$, the single step estimate of true series $y(k)$ is $\hat{y}(k) = N_q[y(k-1)]$ and the long term prediction is achieved by feeding the estimate $\hat{y}(k)$ back as input to the network $\hat{y}(k) = N_q[\hat{y}(k-1)]$. The system could be iterated forward to achieve predictions as far into the future as desired.

3 The Proposed Noise Reduction Method Coupling with Neural Network

In this paper, we propose a possible systematic chaotic noise reduction method including estimating the noise level in real chaos. The method combines the noise level determination using wavelet analysis and the noise reduction using Local Projection method in [7], while the prediction accuracy of FIRNN based on cleaned series is used as main diagnostic tool for defining all parameters. At last, the autocorrelation function is used to show the noise character of the removal components [8]. A step by step procedure of the proposed method is as follows:

1) Using the wavelet analysis to the raw data to estimate an initial value of noise level *v*. *v* could be defined as the ratio of standard noise deviation σ_{noise} to standard data deviation σ_{data}, $v = \sigma_{noise} / \sigma_{data}$. Firstly, apply a *J*-level decomposition to the noisy data to obtain two parts: approximation signal *aJ* and the detailed signals *di*, *i*=1, …, *J*. Secondly we apply a *n*-level decomposition to *d*1. $\sigma_{noise} = m / 0.6745$, where *m* is the median absolute deviation of detail coefficients at the highest resolution level.
2) Apply the Local Projection noise reduction method to the raw data with the initial noise level *v*. Then, the larger and the smaller noise levels than the initial one are respectively used to remove noise from raw data.
3) Determine the prediction accuracy of the data produced by 2) using the FIRNN. Here, we adopt the root mean square error (*RMSE*) to evaluate the performance as prediction accuracy. The *RMSE* is defined as

$$RMSE = \sqrt{\frac{1}{2n}\sum_{i=1}^{n}(y(i)-\hat{y}(i))^2} \tag{4}$$

Where n is the prediction steps, y and $\hat{y}$ represent the actual output data and the predicted data respectively.

4) Calculate the mean *RMSE* of twenty simulations for different v because of the sensitivity of chaotic time series to the initial weights of neural network.
5) Shrink the range of this noise level, and repeat 2)-4).
6) Select the noise level from 5) which produces the smallest prediction error as the terminal noise level.

However, whatever noise reduction methods we use to raw chaotic data, some noise might more or less influence the prediction accuracy. There are two main factors: ① the prediction error couldn't be smaller than the noise level since the future measurement of noise couldn't be predicted; ② the dynamical evolution has to be estimated from the data and this estimate would be affected by noise [1]. Therefore, the prediction error might increase rapidly with noise level. The following simulations also indicate this conclusion.

4 Simulations

In this chapter, we verified our method by the real data: monthly chaotic sunspots time series from January of 1753 to January of 2001 which is shown in Fig. 2.

To demonstrate the wavelet analysis to estimate the initial noise level, a thorough study is first focused on noisy Lorenz and Rossler data with the additive white noise. Table 1 lists the estimated noise level for 20%, 30%, 40% noisy Lorenz and Rossler data using this method. The results indicate that the wavelet undervalues the real noise level more or less, but it provides us with the proximate value.

Using the same method, the initial noise level estimate of sunspots is 0.19. We apply the LP method to reduce the sunspots noise with the neighbor radius about 0.15, 0.25 and 0.35 corresponding to the noise level 15%, 25% and 35%, and the numbers of iterations are exactly the one which removed the used levels. Other parameters such as the embedding dimension and the minimal number of neighbors are chosen as 8 and 35 respectively. The structure of FIRNN is arranged in a way that makes use of the time-delay τ .We use the first normal 2900 points of the raw and noise reduced data to train the network with 1×5×4×1 nodes and $\tau = 8$, 3 and 2 in taps in each layer.

If the noise reduction method really extracts the useful information of chaotic data, the short term prediction accuracy of neural network would be much better. The *RMSE* results of one year prediction (12 months) are shown in Table 2. From this table, when the noise level is 35%, the prediction error is the smallest, and the conclusion could be made that the optimal noise level is near 35%. Therefore, we apply the LP method to the raw data respectively with the noise level for 30% and 40% and the prediction error *RMSE* are given in Table 3. The results show that the smallest prediction error is corresponding to 35% noise level prior to 30%. On the basis of the optimized method, the best noise level is in the range of 30% and 35%. When we select the 34% noise level, the prediction error is 2.9456. Now, we could say that the optimal noise level was about 34% and the final noise reduced data is shown in Fig.3. The autocorrelation function of the removal components is shown in Fig.4 which implies the removal part is noise.

Fig.5 shows the phase-space diagram of monthly sunspots time series and indicates the validity of the proposed systematic method. From this figure, the significant changes in the shape of phase-space diagrams imply the enhancement of determinism. All these results indicate that the 34% noise reduced data closely resemble those of noise-free data and the FIRNN system could learn its approximate attractor.

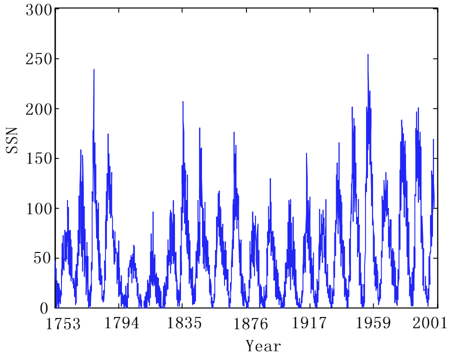


Fig. 2. Time series of raw sunspots data

Table 1. Compared between actual and estimated noise levels of noise added to Rossler and Lorenz

Signal	Actual noise level(%)	Estimated noise level(%)
	20	19.65
Lorenz	30	28.80
	40	35.81
	20	20.50
Rossler	30	26.57
	40	36.63

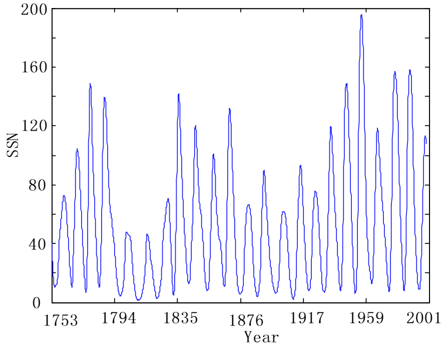


Fig. 3. The 34% noise reduced data

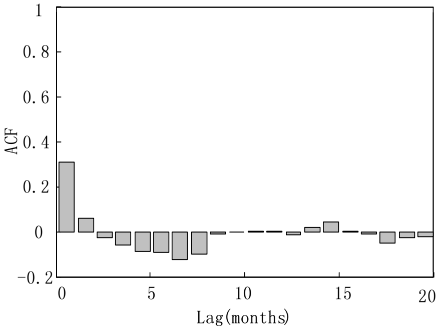


Fig. 4. Autocorrelation function of noise part

Table 2. Prediction accuracy of data

Items	RMSE
Raw data	10.5
15% noise level data	5.7730
25% noise level data	4.4886
35% noise level data	3.0789

Table 3. Prediction accuracy of data

Items	RMSE
40% noise level data	3.8657
35% noise level data	3.0789
30% noise level data	3.1496

5 Conclusions

In this paper, we propose a systematic chaotic noise reduction method coupling with FIRNN. The proposed method is applied to noisy chaotic monthly sunspots time series. The FIRNN is used to predict the cleaned data and the prediction error is considered as the main diagnostic tool to verify the effectiveness of the proposed approach. The simulation of noise-reduction of the sunspot series indicates:

1) The proposed noise-reduction method removes the noise effectively and gets the regular phase space portrait. The significant improvement in the prediction accuracy provides an important support to the above method.
2) The method can keep the chaotic characteristics of sunspots as much as possible. The phase space figures of noise reduced sunspots data are plotted to show the regularities of underlying chaos.

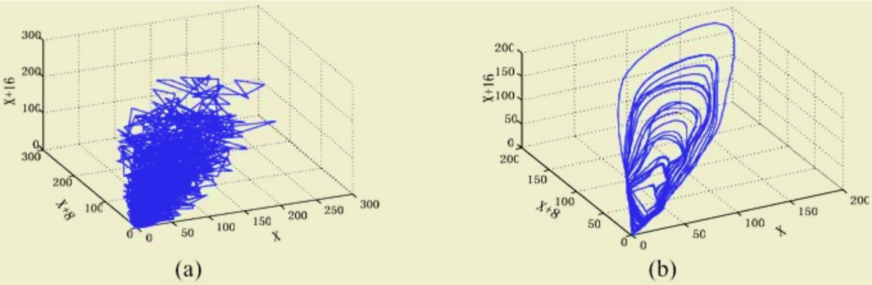


Fig. 5. Phase Space diagram of sunspots data (a) Raw data (b) 34% noise level removal data

Acknowledgements

This research is supported by the project (60374064) of the National Natural Science Foundation of China. It is also supported by the project (50139020) of the National Natural Science Foundation of China. All the support is appreciated.

References

1. Sivakumar, B., Phoon, K-K., Liong, S.-Y., Liaw, C.-Y.: A Systematic Approach to Noise Reduction in Chaotic Hydrological Time Series. Journal of Hydrology. **219** (1999) 103-135
2. Kantz, H., Schreiber T.: Nonlinear Time Series Analysis. Cambridge University Press, Cambridge (2000)
3. Porporato, A., Ridolfi, L.: Nonlinear Analysis of River Flow Time Sequences. Water Res. **33** (1997) 1353– 1367
4. Schouten, J.C., Takens, F., Van den Bleek, C.M.: Estimation of The Dimension of a Noisy Attractors. Phys. Rev. E**50** (1994) 1851-1861
5. Nagayama, I., Akamatsu, N.: Approximation of Chaotic Behavior by Using Neural Network. IEICE Inf. and Sys., E77 (1994) 450-458
6. Wan, E.: Finite Impulse Response Neural Network with Application in Time Series Prediction. Ph.D. Dissertation, Stanford University (1993)
7. Grassberger, P., Hegger, R., Kantz, H., Schaffrath, C., Schreiber, T.: On Noise Reduction Methods for Chaotic Data. Journal of Chaos, **3** (1993) 127-141
8. Jayawardena, A.W., Gurung, A.B.: Noise Reduction and Prediction of Hydrometeorological Time Series: Dynamical Systems Approach vs. Stochastic Approach. Journal of Hydrology, **228** (2000) 242-264

A New Speech Enhancement Method for Adverse Noise Environment

Xiaohong Ma, Yu Wang, Wenlong Liu, and Fuliang Yin

School of Electronic and Information Engineering
Dalian University of Technology, Dalian 116023, China
maxh@dlut.edu.cn

Abstract. A new speech enhancement method combined independent component analysis (ICA) with delay-sum beamforming is presented in this paper. The noise signals are separated from the speech signal by using ICA module whose inputs are the microphone array signals after time delay compensation. Then the output of the delay-sum beamforming and the separated noise signals are executed adaptive noise canceling according to the criterion of minimum output energy (MOE). Through this method, relatively 'pure' speech signal can be obtained in directional noise field or uncorrelated noise field. Some simulation results in the presence of different signal-to-noise ratio (SNR) are shown to demonstrate the validity of the proposed method especially in adverse noise environment.

1 Introduction

The speech signals received by microphone array are often disturbed by noise component in a speech communication system. Multi-microphone techniques are a growing filed in speech enhancement since beamforming and related techniques have a great potential for noise reduction. Spatial information is often exploited to enhance the speech signal in a noisy environment. The conventional method for spatial processing technique is beamforming; include delay-sum beamforming [1], adaptive beamforming [2], [3] and post-filter algorithm [4] and so on. The performance of the beamforming based methods will degrade if any priori information about the acoustical environment and the sources involved is unknown. Besides, a large number of microphones are generally required for good speech enhancement performance. ICA method is an alternative spatial processing technique to beamforming [5]. The weakness of the beamforming algorithm is the advantage of the ICA algorithm. The ICA method also has a disadvantage. When the number of the mixed signals is less than that of the source signals, it becomes a difficult case of the underdetermined problem, in which the complete separation is usually out of the question. To make use of the advantages of the beamforming and the ICA technique, a new speech enhancement method combined ICA with delay-sum beamforming is presented in this paper.

2 The Models of Array Signals

A lineal microphone array of M sensors is considered, which records the plane wave generated by a speaker in the presence of directional noise (such as the noise from air-condition). The array output signal $x_i(k)$ can be expressed as

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 586–591, 2005.

$$x_i(k) = \alpha_i s(k - \tau_i) + \beta_i n(k - \tau_i'), \quad i = 1, 2, \cdots, M \tag{1}$$

where $s(k)$ and $n(k)$ represent speech source signal and noise respectively, received by the array sensors, and $n(k)$ is assumed independent with $s(k)$. τ_i and τ'_i denote the time delay of speech and noise sources of each microphones. α_i and β_i are attenuation factors due to propagation effect.

When there doesn't exist a directional noise source, the environment noise is regarded as the main noise, in this situation the array output signal $x_i(k)$ can be simply expressed as

$$x_i(k) = \alpha_i s(k - \tau_i) + n_i(k) \tag{2}$$

where $n_i(k),\ i = 1, 2, \cdots, M$ is assumed uncorrelated with each other and with $s(k)$.

3 The Proposed Method

The schematic diagram of the proposed method is shown in Fig.1. There are five main stages, the time delay estimation module, the time delay compensation module, the linear weighting module, the FastICA module and the adaptive noise canceling module.

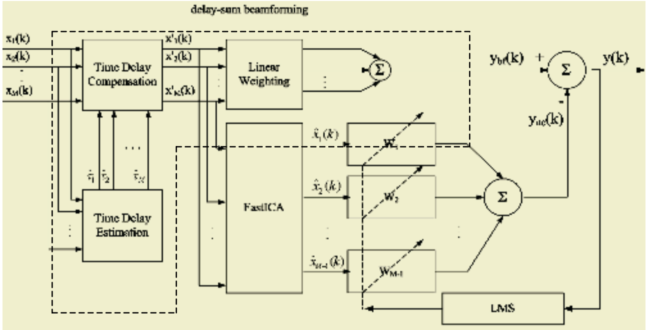


Fig. 1. Schematic diagram of the proposed method

3.1 Time Delay Estimation

We use adaptive eigenvalue decomposition algorithm in the wavelet domain to estimate time delay. Here, we assumed that the Mth microphone is a reference microphone. The time delays are referred to the Mth microphone relative to the other microphones.

The eigenvalue decomposition algorithm (AED) can estimate the time delays between microphone array signals shown in (1) and (2). But the only problem is that it considers little about noise, and the degradation occurs as noise increases. An improved method of AED in the wavelet domain is proposed with three parts. The first part is multi-resolution decomposition of two channel signals. The second part is adaptive eigenvalue decomposition in the wavelet domain. The last part is to change the estimated delay to time domain. Through this method, we can obtain the estimated time delays $\hat{\tau}_1, \hat{\tau}_2, \cdots, \hat{\tau}_{M-1}$, $\hat{\tau}_i$ is the relative time delay between the i^th^ microphone and the M^th^ microphone and $\hat{\tau}_M = 0$.

3.2 Time Delay Compensation

After time delay estimation, the signals $x_i(k)$ should be compensated with the time delays $\hat{\tau}_1, \hat{\tau}_2, \cdots, \hat{\tau}_M$ so that the desired speech signal can arrive at each microphone simultaneously. The compensated signals $x'_i(k), i = 1, 2, \cdots, M$ for (1) and (2) can be expressed separately as

$$x'_i(k) = \alpha_i s(k - \tau_i + \hat{\tau}_i + \tau_0) + \beta_i n(k - \tau'_i + \hat{\tau}_i + \tau_0), \quad i = 1, 2, \cdots, M \tag{3}$$

and

$$x'_i(k) = \alpha_i s(k - \tau_i + \hat{\tau}_i + \tau_0) + n_i(k + \hat{\tau}_i + \tau_0) \tag{4}$$

where τ_0 is a fixed time delay compensation for each received signal to prevent the signals $s(k)$ and $n(k)$ from becoming a noncausal signals. It means that τ_0 is selected to assure $k - \tau_i + \hat{\tau}_i + \tau_0 \geq 0$ and $k - \tau'_i + \hat{\tau}_i + \tau_0 \geq 0$ .If the time delay estimation was accurately, we could get $-\tau_1 + \hat{\tau}_1 + \tau_0 = -\tau_2 + \hat{\tau}_2 + \tau_0 = \cdots = -\tau_M + \hat{\tau}_M + \tau_0 = \tau$. Then we can simplify (3) and (4) as follows

$$x'_i(k) = \alpha_i s(k + \tau) + \beta_i n(k - \tau'_i + \hat{\tau}_i + \tau_0), \quad i = 1, 2, \cdots, M \tag{5}$$

and

$$x'_i(k) = \alpha_i s(k + \tau) + n_i(k + \hat{\tau}_i + \tau_0) \tag{6}$$

3.3 Linear Weighting

As shown in Fig.1, we can see that the time delay estimation module, the time delay compensation module and the linear weighting module constitute a conventional sum-delay beamforming technique. To simplify the calculation, all the weights are set to be $\frac{1}{M}$ to produce one channel signal $y_{bf}(k)$

$$y_{bf}(k) = \frac{1}{M} \sum_{i=1}^{M} x'_i(k) \tag{7}$$

3.4 Independent Component Analysis

From (5), we can see that if we considered the time delays of the noise part, there was *M+1* independent sources altogether. If we ignored the time delays of the noise part because they are smaller enough relative to whole duration of the signal $n(k)$, there was only two independent sources altogether. In both cases, there should be *M+ 1* independent source in (6). In (5) and (6), there are M observed mixtures $x_1', x_2', \cdots, x_M'$ of the source signals. At this time, the number of the mixed signals is less than that of the source signals, and it becomes a difficult case in which the complete separation is usually out of the question. But from the theory of ICA we may arrive at the conclusion that the part of speech signal and the part of noises may be separated if the non-gaussianity of the speech signal is stronger than that of the noises. We all know that the speech signal has a Laplacian distribution which belongs to the supergaussian case, so the condition above is easy to satisfy. The separated *M-1* noise signals are the linear combination of the *M* noises. These signals are useful to us, and we may use them to cancel the noise part in the output of the delay-sum beamforming. Here, we use FastICA method [6] to obtain the reference noises.

3.5 Adaptive Noise Canceling [7], [8]

We can derive one channel signal from delay-sum beamforming module and *M-1* noise signals from FastICA module. The noise signals can be denoted as $\hat{\mathbf{x}}(k)$where $\hat{\mathbf{x}}(k) = [\hat{x}_1(k), \hat{x}_2(k), \ldots, \hat{x}_{M-1}(k)]^T$.

According to the adaptive filter theory, the output of the filter is as follows

$$y_{nc}(k) = \sum_{l=1}^{K} [\mathbf{W}_l(k)]^T \hat{\mathbf{x}}(k-l) \tag{8}$$

where $\mathbf{W}_l(k)$ is a K dimension tap-weight vector. K is the filter length. We can obtain the enhanced speech signal $y(k)$ based on unconstrained LMS algorithm

$$y(k) = y_{bf}(k) - y_{nc}(k) \tag{9}$$

$$\mathbf{W}_l(k) = \mathbf{W}_l(k) + \mu y(k) \hat{\mathbf{x}}(k-l) \tag{10}$$

where μ is the step-size.

4 Experimental Results

The array of three microphones is used in our simulations. In each experiment, the speech signal, whose sampling rate is 8kHz, was delayed for appropriate times according to their spatial locations. Then they were added with uncorrelated or correlated noises at different SNR range form -13db to 0db and then truncated to 32ms analysis frames using rectangle window. The step-size μ of adaptive algorithm was set to be 0.01.

With the uncorrelated noise and the directional noise, the SNR comparisons between the proposed method and the GSC method [2] are shown in Fig.2 and Fig.3 respectively. From Fig.2 we can see that the SNR is improved obviously by using the proposed method than that of the GSC method. As shown in Fig.3, although the performance of the two methods is not as good as shown in Fig.2, the proposed method is still more effective than the GSC method.

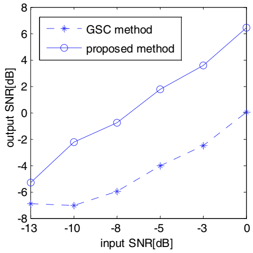


Fig. 2. SNR comparison for uncorrelated noise

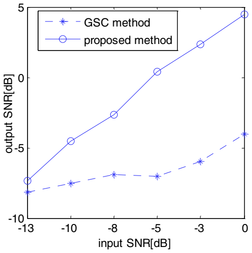


Fig. 3. SNR comparison for directional noise

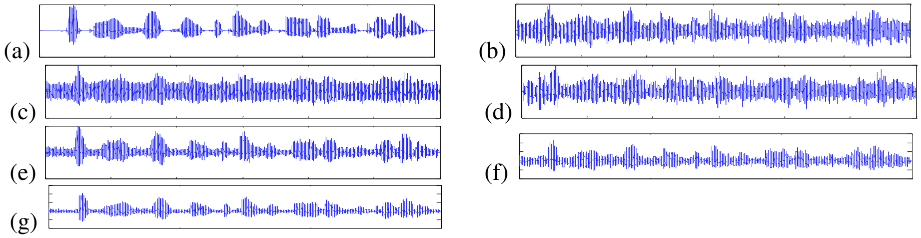


Fig. 4. Two examples of speech enhancement by the proposed method.(a) pure speech signal. (b)-(d) speech signal with -5db uncorrelated noise. (e) enhanced signal for uncorrelated noise by the proposed method. (f) speech signal with 0db directional noise (g) enhanced signal for directional noise by proposed method

Two examples of speech enhancement experiments by using the proposed method is shown in Fig.4. Fig.4 (a) shows the pure speech signal. Fig.4 (b)-(d) are the speech signal with −5 db uncorrelated noise. Fig.4 (e) is the enhanced result for uncorrelated noise using the proposed method. Fig.4 (f) is the speech signal with 0db directional noise. Fig.4 (g) is the enhanced result for directional noise using the proposed method Compared Fig.4 (b)-(d) with Fig.4(e) and Fig.4 (f) with Fig.4(g), we can see that the enhanced speech signal is more 'pure' than the noisy signals. And the hearing test shows that the proposed method does little harm to the speech signal.

5 Conclusion

A new speech enhancement algorithm based on independent component analysis and delay-sum beamforming is proposed in this paper. ICA is used to separate the noise signals from the speech signals. Then, we use adaptive noise canceling to enhance the output signal of the delay-sum beamforming module. The simulation results show the effectiveness of the proposed method in the adverse noise environment. Especially, the listening tests show that it does little harm to the speech signal.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. 60172073, and No. 60372082.

References

1. Flanagan, J.L., Johnston, J. D., Zahn, R., Elko, G. W.: Computer–steered Microphone Arrays for Sound Transduction in Large Rooms. J. Acoustic. Soc. Amer., **75** (1985) 1508–1518
2. Griffiths, L.J., Jim, C.W.: An Alternative Approach to Linearly Constrained Adaptive Beamforming. IEEE Transactions on Antennas and Propagation, **30** (1982) 27–34
3. Gannot, S., Burshtein, D., Weinstein, E.: Analysis of the Power Spectral Deviation of the General Transfer Function GSC. IEEE Transactions on Signal Processing, **52** (2004) 1115–1121
4. Zelinski, R.: A Microphone Array with Adaptive Post–filtering for Noise Reduction in Reverberant Rooms. Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '88), (1988) 2578–2581
5. Siow Yong Low, Nordholm, S., Togneri, R.: Convolutive Blind Signal Separation with Post-Processing. IEEE Transactions on Speech and Audio Processing, **12** (2004) 539–548
6. Hyvarinene, A.: Fast and Robust Fixed-point Algorithms for Independent Component Analysis. IEEE Transactions on Neural Networks, **10** (1999) 626–634
7. Widrow, B. W. et al: Adaptive Noise Canceling: Principles and Applications. Proceedings of the IEEE, **63** (1975) 1692–1975
8. Marvin R. S.: Adaptive Noise Canceling for Speech Signal. IEEE Transactions on Acoustics, Speech and Signal Processing, **26** (1978) 419–423

A Subband Adaptive Learning Algorithm for Microphone Array Based Speech Enhancement

Dongxia Wang[1,2] and Fuliang Yin[1]

[1] School of Electronic and Information Engineering, Dalian University of Technology
Dalian, Liaoning 116023, China
[2] Information Science and Engineering College
Liaoning Institute of Technology, Jinzhou, Liaoning 121001, China
wdx1976@sohu.com

Abstract. This paper describes a subband adaptive learning algorithm for enhancing microphone array speech signals degraded by a considerable amount of acoustic background noise. The subband multichannel adaptive learning algorithm is adopted to overcome the drawback of slow convergence as well as high computational complexity, which is associated with full band multichannel adaptive LMS algorithm. Simultaneously, oversampled Cosine-modulated filter banks instead of critical sampling filter banks are used to reduce the aliasing effects of subband itself. Simulations experiments show that in addition to fast convergence speed, the proposed microphone array speech enhancement method based on subband adaptive learning algorithm also exhibits a better noise reduction performance than well-known Generalized Sidelobe Canceller (GSC).

1 Introduction

In many speech communication applications such as hands-free teleconference, the recorded and transmitted speech signals are often corrupted by a considerable amount of acoustic background noise which causes signal degradation and further decreases the performance of following speech coding and speech recognition systems. Therefore an efficient noise reduction algorithm is required. Microphone array instead of single microphone, incorporating both spatial and temporal information, has been widely used to enhance the noisy speech signals in an enclosure environment [1-4].

These algorithms, providing better speech enhancement performance, however, are based on full band adaptive learning algorithm which leads to slow convergence speed with the condition number of input signal correlation matrix increased. It is known from adaptive filter theory that subband techniques present several advantages with respect to the full band approach [5], [6]. An important property is due to the splitting of the input signal, the eigenvalues spread of the subband signal's autocorrelation matrix is reduced and consequently least squares like adaptive learning algorithms present better convergence performance.

Therefore the aim of this paper is to present an application of subband adaptive learning algorithm to full band microphone array based speech enhancement method to achieve faster convergence speed. The main advantages of proposed method are

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 592–597, 2005.

both convergence speed significantly increased and noisy speech signal significantly enhanced by including oversampled Cosine modulated filter banks information into Generalized Sidelobe Canceller (GSC) and post-filtering process, which would greatly facilitate many applications, including automatic speech recognition and speaker identification.

The outline of this paper is as follows. In Section 2, we formulate the problem of microphone array based speech enhancement in the time domain. Section 3 presents the scheme of subband adaptive learning algorithm based oversampled Cosine-modulated GSC and post-filtering beamforming for speech enhancement. In Section 4, the experimental results of proposed method are given as well as comparison with fullband learning algorithm. Finally, Section 5 makes a summary of the proposed algorithm and draws a conclusion.

2 Problem Formulation

Let us consider a microphone array in the acoustic enclosure environment where desired speech signal and noise coexist. The *i*th microphone signal, comprised of two components, the nonstationary target speech signal and the stationary noise signal, can be modeled as

$$x_i(n) = h_i(n) * s(n) + n_i(n)\,. \tag{1}$$

where $s(n)$ is the target speech signal, $n_i(n)$ is the noise, $h_i(n)$ is the room impulse response between the speech source and the *i*th microphone. Noise is uncorrelated with target speech signal both time and spatial domain.

Our goal is to find an effective learning algorithm to reconstruct the speech signal component from all the microphone signals $x_i(n)$. The whole speech enhancement system structure proposed is shown in Fig.1. In addition before subband processing all microphone signals are processed with time delay compensation (TDC) after time delay estimation (TDE) which is assumed to be estimated correctly here.

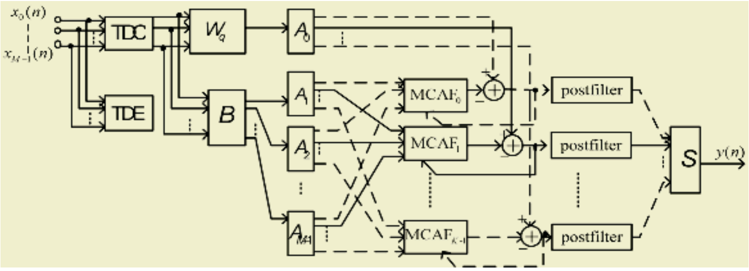


Fig. 1. Structure of microphone array based speech enhancement algorithm

3 Subband Adaptive Learning Algorithm

The proposed structure is composed of *M* oversampled Cosine modulated analysis filter banks which split both the output signals of block matrix and upper beamforming channel into K subbands each with a decimation factor of D. Likewise, a synthesis filter bank with interpolation factor D is used to reconstruct the output signals of all the subband multichannel adaptive filters into full band representation. Due to oversampling ($D < K$), a low alias level in the subband signals can be achieved. The analysis and synthesis filters are Cosine-modulated shifted versions of the linear-phase lowpass prototype filters, i.e.

$$h_k(n) = 2h(n)\cos((2k+1)\frac{\pi}{2K}(n-\frac{L-1}{2})+(-1)^k\frac{\pi}{4}). \tag{2}$$

$$f_k(n) = 2f(n)\cos((2k+1)\frac{\pi}{2K}(n-\frac{L-1}{2})-(-1)^k\frac{\pi}{4}). \tag{3}$$

where $h(n)$ and $f(n)$, analysis and synthesis prototype filter respectively, are both linear phase and the same filter length is assumed to be an integer multiple of 2K. The design of the prototype filters ensures minimization of the stopband energy [7].

The block matrix B decomposes the microphone signals both in the spatial domain and in the temporal domain, in which the rows vectors simultaneously perform a temporal filtering operation. Here blocking matrix $B \in \Re^{M\times(M-1)}$ is set up such that it can prevent a target signal from passing through to the auxiliary channels. Here hypotheses the time delay is compensated correctly. Then we have the block matrix

$$B = \begin{bmatrix} \beta_0 & \beta_1 & 0 & \cdots & 0 & 0 \\ 0 & \beta_0 & \beta_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \beta_0 & \beta_1 \end{bmatrix}_{M\times(M-1)} .$$

where apart from being shift one sensor per column, the values in the block matrix are identical. The structure of block matrix implies all input signal have passed a spatial band pass filter and $M-1$ spatial band limited filter consisting of two samples $g(n) = [\beta_0 \quad \beta_1]$ satisfying

$$\mathbf{1}^{\mathrm{T}} B_{M\times M-1} = 0 . \tag{4}$$

where $\mathbf{1} \in \Re^{M\times 1}$ is an M element vector containing ones.

Define at time $n = \left\lfloor \frac{n_1}{D} \right\rfloor$ which D is decimation factor, desired signal vector $\mathrm{d}(n) \in \Re^{K\times 1}$, input signal vector $\mathrm{u}_k(n) \in \Re^{(M-1)\times 1}$, spatial filter coefficients matrix $\mathrm{W}_k(n) \in \Re^{(M-1)\times L}$ of the kth subband multichannel adaptive filter as

$$\mathrm{d}(n) = [d_0(n),\cdots,d_k(n),\cdots,d_{K-1}(n)]^{\mathrm{T}}, \mathrm{u}_k(n) = [u_{0,k}(n),\cdots,u_{M-2,k}(n)]^{\mathrm{T}} .$$

$$\mathrm{W}_k(n) = [\mathrm{w}_k^{(0)}(n),\cdots,\mathrm{w}_k^{(p)}(n),\cdots,\mathrm{w}_k^{(L-1)}(n)]^{\mathrm{T}}, \mathrm{w}_k^{(p)}(n) = [w_{0,k}^{(p)}(n),\cdots,w_{M-2,k}^{(p)}(n)]^{\mathrm{T}} .$$

where L is the filter length. The kth subband output signal $v_k(n)$ is

$$v_k(n) = \sum_{p=0}^{L-1} [\mathrm{w}_k^{(p)}(n)]^{\mathrm{T}} \mathrm{u}_k(n-p) \,. \tag{5}$$

and error signal is equal to

$$e_k(n) = d_k(n) - v_k(n) \,. \tag{6}$$

Then the coefficient update equation based on subband learning algorithm is given by

$$\mathrm{w}_k^{(p)}(n+1) = \mathrm{w}_k^{(p)}(n) + \frac{\alpha}{P_k[\mathrm{u}_k(n)]} \mathrm{u}_k^*(n) e_k(n) \cdot \tag{7}$$

where

$$P_k[\mathrm{u}_k(n)] = \beta P_k[\mathrm{u}_k(n-1)] + (1-\beta) \sum_{l=0}^{M-2} \left\| u_{l,k}(n) \right\|^2 \,.$$

where β is a forgetting factor (typically $0.7 < \beta < 1$), $P_k[\mathrm{u}_k(n)]$ is the power estimate of input snapshot data vector $\mathrm{u}_k(n)$. The frequency response of postfilter is

$$H_k(\mathrm{e}^{j\omega}) = \frac{\dfrac{2}{(M-1)(M-2)} \displaystyle\sum_{i=0}^{M-3} \sum_{j=i+1}^{M-2} P_{u_i u_j}(\mathrm{e}^{j\omega})}{P_{e_k}(\mathrm{e}^{j\omega})} \,. \tag{8}$$

The output signal of the whole system is

$$y(n) = \begin{cases} \displaystyle\sum_{k=0}^{K-1} s_k(n) * f_k(n) = \sum_{k=0}^{K-1} [\mathrm{F}^{-1}(H_k(\mathrm{e}^{j\omega})) * \mathrm{e}_k(n)] * f_k(n), & n = mD \\ 0, & \text{otherwise} \end{cases} \,. \tag{9}$$

where $f_k(n)$ is defined as (3), F^{-1} and $*$ are the symbols of invertible Fourier transform and convolution respectively, m is integer and D is interpolation factor.

4 Experimental Results

To illustrate the performance of the proposed method, we carried out extensive computer experiments with different signal-to-noise ratios (SNR). The simulation room configuration is depicted with dimensions of $6m \times 4m \times 3m$. The linear equal spaced microphone array is used with $M = 6$ microphones and the distance between two adjacent microphones is 5cm. The target signal illuminates the array from the broadside 90° with sampling frequency 8000 kHz. The room impulse response is obtained using the image method [8], with the filter length of 1500 taps and for different reverberation times T_{60}. A prototype linear phase filter of length $L = 128$ for $K = 16$ subbands with decimation (interpolation) factor $D = 8$ is designed.

Fig.2 shows the convergence properties of subband adaptive learning algorithm as well as full band counterpart, where the step size in subband adaptation is set to $\mu = 0.02$, $\mu = 0.001$ respectively. From Fig.2, we can see that compared to the full band, the proposed method has a faster convergence speed, which is due to prewhitening effect of the analysis filter banks on the signals in each subband.

Fig.3 shows the waveforms of the speech component recorded by the microphone under the reverberation time $T_{60} = 300ms$ at SNR level of 0dB. It can be seen that residual noise is significantly reduced by the proposed algorithm, which acquires better speech quality and is proved further by the informal listening test. The proposed method can also serve as a speech enhancement scheme when the environmental noises are color noise due to the pre-whitening effect of analysis filter banks on the signals. Simultaneously, we also see that deverberation is dissatisfactory which has leads to our feature work.

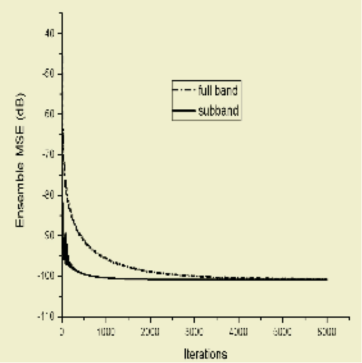


Fig. 2. Learning curves

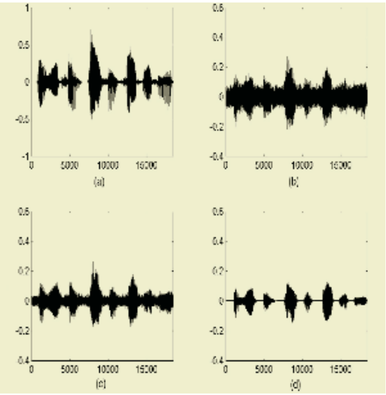


Fig. 3. Speech waveforms: (a) Clean signal (b) Noisy signal (c) Enhanced by GSC (d) Enhanced by proposed method

Table 1. Noise reduction performance under different SNR

Noise Reduction	$T_{60} = 300ms$				
	5dB	0dB	5dB	10dB	15dB
GSC	6.86dB	5.37 dB	3.87 dB	2.26 dB	1.58 dB
proposed	7.91 dB	6.43 dB	4.43 dB	3.39 dB	2.30 dB

Table.1 illuminates the noise reduction performance under different SNR with the same middle reverberation. We can see that the proposed algorithm produces higher noise reduction than full band GSC.

5 Conclusions

Subband adaptive learning algorithm has been shown to be applicable to the microphone array speech enhancement system by using a proper filter banks constructed by cosine modulated oversampling. Experiment results on algorithm convergence speed and speech enhancement performance demonstrate the effectiveness of the proposed method. Since subband adaptive learning algorithm here uses oversampled cosine modulated filter banks, only based on one desired prototype filter, its computational requirements for learning is substantially reduced compared with the DFT filter banks based method.

With reverberation time increased, we also see that deverberation ability of the proposed algorithm is dissatisfactory which will lead to our feature work.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. 60372082 and No. 60172073, Trans-Century Training Program Foundation for the Talents by the Ministry of Education of China.

References

1. Griffiths, L.J, Jim, C.W.: An Alternative Approach to Linearly Constrained Adaptive Beamforming. IEEE Transactions on Antennas and Propagation, **30** (1981) 27–34
2. Hoshuyama, O.: A Robust Adaptive Beamformer for Microphone Arrays with a Blocking Matrix Using Constrained Adaptive Filters, IEEE Transactions on Signal Processing, **47** (1999) 2677–2684
3. Gannot, S., Burshtein D., Weinstein, E.: Signal Enhancement Using Beamforming And Nonstationary with Applications To Speech. IEEE Transactions on Signal Processing, **49** (2001) 161–1626
4. Gannot, S, Cohen, I..: Speech Enhancement Based on the General Transfer Function GSC and Postfiltering. IEEE Transactions on Speech and Audio Processing, **12** (2004) 561–571
5. Fleige, N.J.: Multirate Digital Signal Processing. Wiley, New York (1994)
6. Yang.Y. G., Cho. N. I., Lee. S. U.: On the Performance Analysis and Applications of the Subband Adaptive Digital Filter. Signal Processing, **41** (1995) 279–294
7. Kliewer, J., Mertins, A.: Design of Paraunitary Oversampled Cosine-modulated Filter Banks. IEEE International Conference on Acoustics, Speech, and Signal Processing, **3** (1997) 2073–2076
8. Allen, J.B., Berkley, D.A.: Image Method for Efficiently Simulating Small Room Acoustics. Journal of the Acoustic Society of America, **65** (1979) 943–950

A Spiking Neuron Model of Auditory Neural Coding

Guoping Wang and Misha Pavel

OGI School of Science & Engineering at Oregon Health & Science University
20000 NW Walker Road, Beaverton, OR 97006, USA
{gpwang,pavel}@bme.ogi.edu

Abstract. The focus of this paper is to propose an explanation for how biological auditory mechanism is able to use spiking neurons to code high bandwidth information using information channels with very slow sampling rates (< 20 Hz). The general approach described in this paper is to decompose the signal into narrow band channels, each of which can be sampled at a frequency that is much lower than the center frequency of the corresponding narrow band filter. The new idea here is that the system can use non-uniform sampling to capture both the amplitude of the modulation and the phase of the carrier signal. In this paper, we first describe a system based on FFT analysis combined with overlap-add and a sampling process where magnitude is digitized but phase is represented using a temporal code of spiking neurons. The coding/decoding mechanism is based on the properties of the refractory period. We demonstrate that it is possible to reduce the bit rate to 50% by coding the carrier phase using the timing of the pulses. In the second part of this paper we show how a biological system may approximate the broadband auditory signal using spiking neurons in conjunction with a simple model of neural refractory period.

1 Introduction

A fundamental feature of an intelligent signal processing and pattern recognition system is its ability to adapt to dynamic and unpredictable conditions. Unlike the biological system that is quite robust when confronted with complex and unpredictable world [1], most engineering systems are sensitive to the contextual and environmental changes. For example, typical automatic speech recognition systems are designed to perform well in well-defined situations and contexts, but exhibit undue sensitivity to irrelevant changes. Similarly, image recognition that work well in numerous specific domains, cannot solve the general problem of object recognition plagued by its dependency on the specific representation, pose and lighting condition. Another example, although a number of DARPA-sponsored competitions over the years have led to decreasing error rates on increasingly difficult problems, the ability of a machine to recognize speech is still so inferior to the ability of a human [2,3]. Lippman (1997) gives us some comparisons between the error rate of machines and humans on a range of tasks [4]. Roughly, his results indicate that human error rates are an order of magnitude smaller than those obtained by ASR algorithms for clean speech, and two orders of magnitude smaller for typical noise conditions.

One possible explanation of the robustness of biological systems involves the representation of sensory signals as well as the prior knowledge. Biological perceptual

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 598–604, 2005.

systems seem to deploy flexible, multi-feature approaches that are redundant and highly adaptive with neither single most important cue, nor invariant cue. Instead, these recognizers seem to be able to use the most appropriate cues for any given situation. Such representations are typically redundant and not statistically efficient, but provide mechanisms to achieve robustness. In contrast, the traditional approach to statistically optimal pattern recognition takes a fixed and limited feature approach and strives to exploit as much as possible the information in the training data and remove all detectable redundancy. This approach, based on optimal use of training data, will perform poorly when information that is reliable in the training set is compromised in the test set [5].

One aspect common to most biological systems is that the sensory information used to make all the decisions is represented by action potential – spikes – carried by highly parallel neural subsystems. This neural code must be sufficiently rich to represent most possible inputs, e.g. acoustic stimuli such as produced during screaming, and at the same time must sufficiently structured to allow fast adaptation to specific contexts. In this paper we focus on the first of these requirements – completeness of the representation by spiking neurons.

This representation may be the result of the constraints of the processes and structures making up the neural system. On the other hand, we consider the possibility that this type of distributed, stochastic representation of the signals can support the superior resilience of these systems to the transformations of the physical signals and the changing environmental conditions.

Recently, there have been lots of efforts focused on the spike representation of signals [11, 12]. In this paper, we discuss a way that the natural auditory system represents high-bandwidth acoustic information with relatively slow sampling rates of the individual neurons. This notion provides a reason for the well-known narrow tuning of the peripheral auditory system. We explore the ability of the temporal code generated by spiking neurons to represent and communicate acoustic signals in narrow frequency bands. This type of representation is consistent with the findings that temporal envelope plays an important role in human speech recognition [6], and therefore, this type of representation may be useful for automatic speech recognition systems.

2 Signal Processing and Representation

The acoustic signal processing in mammal peripheral auditory systems can be characterized by three processing stages [7]:

- Narrow band filtering by the mechanical vibrations of the basilar membrane in the inner ear and transduction by the hair cells.
- The information in the narrow-band filter is coded in spike trains generated by cochlea neurons.
- Pattern recognition performed on the neural representation. Since the focus of this paper is the signal representation rather than pattern recognition, we use signal reconstruction instead of pattern recognition.

These three processing stages are represented in our model by the following three algorithms:

- Short time Fourier analysis to generate narrow band signals.
- Model individual neurons and the processes that generate spikes.
- Reconstruct the input signal from the spike train.

2.1 Narrow Band Signal Representation Using Short-Time Fourier Analysis

We use Fourier analysis to simulate the narrow band signal that serves as the input to the individual auditory neurons. We apply Short Time Fourier Transform (STFT) to the signal and then obtain the subband signal by inverse STFT analysis of each frequency component (other components are set to zero). With an appropriate type of window (Hanning) combined with an overlap-add process we can perfectly reconstruct the original signal. Therefore, this narrow-band representation preserves all the original signal information.

More specifically, given a digital signal $x(n)$, we first calculate the short time Fourier transform $X(l,k)$ of the l^{th} frame Hanning windowed signal. The k^{th} subband real time signal for the l^{th} frame can be obtained by inverse Fourier transform of the k^{th} component (others are set to zero), written as:

$$x_l(k,n) = real\left(IFFT\left(X\left(l,k\right)\right)\right) = real\left(X\left(l,k\right)e^{jw_k n}\right),\ l = 0,1,\cdots.$$

The whole range subband signal can be obtained by overlap adding $x_l(k,n)$, resulting $x(k,n) = \sum_{\substack{\forall l \text{ which covers} \\ \text{sample n}}} x_l(k, n \bmod H)\ l = 0,1,\cdots$, where H indicates the offset by each successive frame. Note that the resulting narrow-band signal can be represented in terms of a carrier signal with frequency at the center of the band amplitude-modulated by band-limited signal with frequency determined by the narrow band.

2.2 Sampling – Spike Train Generation

A complete model of a spiking neuron would include integrate and fire model [8, 9, 13] in combination with a process that represent the refractory period of the neuron. For the purpose of simplicity of this presentation, we focus on the effect of the refractory period. In fact, we assume that after firing, a neuron will slowly recover its sensitivity and we represent this process by a monotonically decreasing refractory function $g(t - t_{i-1})$ where t_{i-1} is the time of the previous firing – for the purpose of this presentation we ignore the short absolute refractory period. This function can be interpreted as a threshold which, when exceeded by the input signal, is responsible for a generation of a spike. The actual shape of the threshold function is not critical and for the purpose of this paper, we assumed that it is well approximated by an exponential, namely $g(t) = A\exp\left(-\frac{1}{\tau}\left(t - t_{i-1}\right)\right)$, where t_{i-1} is the previous firing time and τ controls the decay of the threshold over time. We assume that the membrane potential is generated proportional to the amplitude of the input stimuli, so that we generate a spike whenever the instantaneous amplitude of the input signal exceeds the value of the threshold function as shown as in Fig.1 (a) and (b).

Because of the non-linearity of this system, we examine its properties using simulation. The input is white Gaussian noise and the states of the subband neurons is randomized, so that the first firing time t_1 is in the range of 0 – 12.5msec (τ) after the signal onset. Each subsequent firing occurs at the peak of the carrier nearest to the point at which the threshold function was exceeded. After a spike is generated, the refractory function is initialized and the process is repeated. The resulting signal representation is a sample sequence with sample times $t_k^{samp} = [t_1^k \; t_2^k \; \cdots \; t_{N(k)}^k]$. Because of the nature of the sampling process, $N(k)$ and the spike train may be different for different bands. Since the function g is known, both the phase and the amplitude modulation of the narrow-band signal can be reconstructed – see Section 2.3. The critical aspect of this sampling process is that the carrier phase is preserved by triggering the spikes at the peaks of the carrier signal. This sampling process preserves the phase of the carrier, but introduces amplitude errors in the modulating signal. The sampling process can be modified by trading off between the phase and amplitude errors. An illustration of the tradeoff is shown in figure (Fig.2 (a)) where we compare phase-based and amplitude-based coding. This result illustrates the fact that the signal-to-noise ratios (SNR) are much higher when spikes are generated at the peaks of the signal, thereby preserving the phase of the carrier.

In order to represent the biological systems more accurately, we introduced additive noise that randomized the sampling times [9].

2.3 Reconstruction of the Signal

To assess the amount of information encoded in the spike train $t_k^{samp} = [t_1^k \; t_2^k \; \cdots \; t_{N(k)}^k]$, we compared the original signal to its estimate obtained by signal reconstruction from the samples. We note that the spike train provides amplitude information about the original signal because $\hat{a}_k^{samp} =$ $[g(t_1^k)\; \mathrm{g}(t_2^k)\; \mathrm{g}(t_3^k)\; \cdots\; \mathrm{g}(t_{N(k)}^k)]$ provides an approximation to the amplitude samples $a_k^{samp} = [x(t_1^k)\; x(t_2^k)\; x(t_3^k)\; \cdots\; x(t_{N(k)}^k)]$.

The first step of the reconstruction process involves interpolation of the amplitudes $\hat{a}_k^{samp}$ providing amplitude estimates at each point t $\tilde{a}_k(t)$, for $t = 1 \ldots N$. The phase is calculated using the assumption that the signal was sampled at the peaks of the subband signal. Theoretically, each subband signal can be approximated as $x_k(t) =$ $a_k(t)\cos(2\pi kt + \phi(t))$, at the spikes (peaks) $\cos(2\pi kt_k^{samp} + \phi(t_k^{samp})) = 1$, $2\pi kt_k^{samp} + \phi(t_k^{samp}) = 0$, $\phi(t_k^{samp}) = -2\pi kt_k^{samp}$ so the phase $\phi(t_k^{samp})$ at the sampled point t_k^{samp} is also known. We interpolate and smooth (unwrap) the phase to obtain the phase estimates at each sample point $\tilde{\phi}_k(t)$, for $t = 1 \ldots N$. The k^{th} subband signal is reconstructed using $\tilde{x}_k(t) = \tilde{a}_k(t)\cos(2\pi kt + \tilde{\phi}_k(t))$. The temporal encoding of the amplitude is transparent in the analytic expression for the spike signal, given by the expression:

$$a_k(t_i) = g(t_i) = A\exp\left(-\tfrac{1}{\tau}\left(t_i - t_{i-1}\right)\right) \approx A\left(1 - \tfrac{1}{\tau}\left(t_i - t_{i-1}\right)\right) = \tfrac{A}{\tau}\left(\tau - \left(t_i - t_{i-1}\right)\right).$$

3 Evaluation

As noted in Section 2 we evaluated the proposed signal representation using white Gaussian noise with zero mean and unity standard deviation, i.e. $x \sim N(0,1)$ as the input to the model. The motivation for using stochastic input signals is that this type of input contains most uncertainties, e.g., has the lowest possible autocorrelation and therefore it is most difficult to represent. Other signals such as speech are more predictable and should impose less stringent requirements on the representation. The evaluation was performed using segments of the signal with 4096 samples, at the sampling frequency of 16 kHz. These segments were filtered using a 512-point short time Fourier transform described in section 2. To simulate the spontaneous and random spikes of the neuron, we added random Gaussian noise to the subband signal and also controlled the probability of neuron firing at each cross point of $g(t)$ and $x_f(t)$. Fig.1 (a) shows subband signal centered at $f_c = 1000$ Hz after the inverse transform and overlap adding. It is very clear that although only a few spikes are generated (53 spikes) for the signal which consists of 4096 samples (Fig.1 (b)); the signal is still characterized very well (Fig.1(c)). Because we only need to allocate one bit for each spike, the total bytes for the spike train are 53*256/8 = 1696, and the bit rate is reduced to 1696/4096 = 41.41% of the original bit rate.

The histogram of interspike interval (Fig. 2(b)) from 20 neurons shows that the time intervals between nerve firings are approximately integral multiples of the period (1 millisecond for $f_c = 1000$ Hz) of the stimulating waveform, which is consistent with the prior experimental results of Rose et al. in [7,10].

4 Conclusions

In this paper, we have examined a possible neural model that underlies the mechanisms of the auditory neurons used to encode the acoustic input stimuli. The model based on the fact that the refractory period of each neuron represents a natural mechanism that converts amplitude to temporal code. We found that the narrow band envelope information could be encoded simply in the temporal interspike intervals. Using this model, auditory neurons can encode the narrow band signal in the neuron spiking trains and we demonstrated the utility of this model by comparing the reconstructed signal to that of the original input signal. Our results show that the auditory system could have processed the high bandwidth information both by spatially decomposing input so that each neuron only needs to respond to band limited signal, and by temporally encoding signal in spikes.

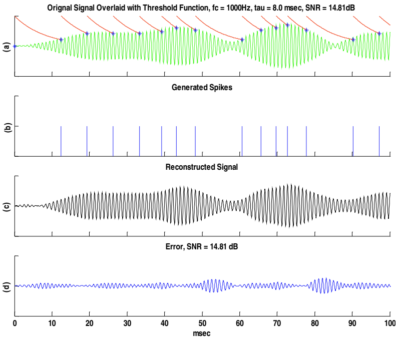


Fig. 1. Illustration of the spike generation and signal reconstruction (a) Subband signal $f_c = 1000$ Hz (b) Spike train (c) Reconstructed signal (d) Error

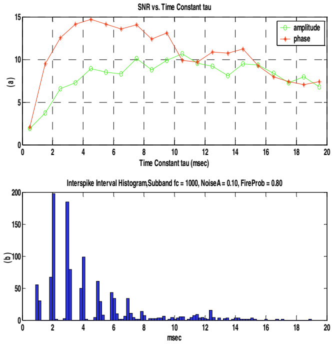


Fig. 2. (a) Comparison of the SNRs verse time constant τ when sampling at the peaks (phase) or at the amplitude cross point (amplitude) (b) Histogram of interspike interval for subband signal ($f_c = 1000$ Hz) extracted from 20 neurons

Acknowledgements

This research was supported by a NASA Grant NCC2-1218 to Oregon Health & Science University. We are grateful to Hynek Hermansky for fruitful discussions.

References

1. Pavel, M., Hermansky, H.: Information Fusion by Human and Machine. First European Conference on Signal Analysis and Prediction. Strahov Monastery, Prague, Czech Republic (1997)
2. DARPA's EARS Conference (2003)
3. DARPA's EARS Kickoff Meeting (2002)
4. Lippman, R.P.: Speech Recognition by Machines and Humans. Speech Communication **22** (1997) 1-15.
5. Wang, G., Pavel, M., Song, X.: Robust Recognition Based on Adaptive Combination of Weak Classifiers. Proceedings of the International Joint Conference on Neural Networks, (2003)
6. Shannon, R.V., Zeng, F., Kamath, V., Wygonski, J., Ekelid, M.: Speech Recognition with Primarily Temporal Cues. Science, New Series, **270** (1995) 303-304.
7. Moore, B.C.J.: An Introduction to the Psychology of Hearing. Academic Press (1989)
8. Gerstner, W., Kistler, W.: Spiking Neuron Models – Single Neurons, populations, plasticity. Cambridge University Press (2002)
9. Rieke, F., Warland, D., van Steveninck, R., Bialek, W.: Spikes: Exploring the Neural Code. MIT Press
10. Rose, J.E., Brugger, J.F., Anderson, D.J., Hind, J.E.: Patterns of Activity in Single Auditory Nerve Fibers of the Squirrel Monkey. Hearing Mechanism in Vertebrates (ed. A.V.S. de Reuck and J. Knight), Churchill, London

11. Lewicki, M.S.: Efficient Coding of Natural Sounds. Nature Neuroscience, **5** (2002) 354-363
12. Lewicki, M.S., Sejnowski, T.J.: Coding Time-Varying Signals Using Sparse, Shift-Invariant Representations. Advances in Neural Information Processing Systems, MIT Press **11** 730-736
13. Smith, L.S., Fraser, D.S.: Robust Sound Onset Detection Using Leaky Integrate and Fire Neurons with Depressing Synapses. IEEE Transactions on Neural Networks, **15** (2004) 1125-1134.

Blind Feature Extraction for Time-Series Classification Using Haar Wavelet Transform

Hui Zhang, Tubao Ho, and Wei Huang

School of Knowledge Science
Japan Advanced Institute of Science and Technology
Tatsunokuchi, Ishikawa 923-1292, Japan
{zhang-h,bao,w-hang}@jaist.ac.jp

Abstract. Time-series classification has attracted increasing interest in recent years, particularly for long time-series as those arising in bioinformatics and financial domain. Many dimensionality reduction algorithms have been proposed to attack the so-called *curse of dimensionality* problem. However, choosing the number of features is not a trivial task and has not been well considered. In this paper, we propose a novel *blind* feature extraction algorithm with Haar wavelet transform which can determine the feature dimensionality automatically. The algorithm takes the tradeoff of achieving lower dimensionality and lower sum of squared errors between the features and original time-series. Experimental results performed on several widely used time-series data demonstrate the effectiveness of the proposed algorithm.

1 Introduction

Time-series data accounts for a huge amount of data stored in financial, gene expression, medical and science databases. Classification is one of the most frequently used data mining techniques that is an supervised learning process of labelling the test data by a model built from training data set. When handling long time-series, the time required to compute the neighbors becomes prohibitive. Moreover, the *curse of dimensionality*, that affects any problem in high dimensions, causes highly biased estimates, thus decreases the classification performance.

Many dimensionality reduction algorithms have been proposed to attack the *curse of dimensionality* problem, such as Singular Value Decomposition (SVD) [2], Discrete Fourier Transform (DFT) [1], and Discrete Wavelet Transform (DWT) [5]. These techniques transform the time-series into a feature space and the features are ordered in importance. Thereby we can take the first few features to represent the whole time-series for the further classification task. All these algorithms work well for time-series on optimal dimensions because the high correlation between time series data makes it possible to remove huge amount of redundant information. However, determining appropriate dimensionality of the features is a challenging task. The feature extraction approaches by using

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 605–610, 2005.

these transforms can be divided into the *feedback* type and the *blind* type. Feedback feature extraction techniques use the feedback of classification accuracy with various feature dimensionality to determine optimal dimensionality. The blind approach reduces the feature dimensionality firstly then use these features for classification. It is obvious that the blind approach is faster than feedback methods without searching all candidate selections. Our work in this paper is belonged to the blind type.

Among the proposed dimensionality reduction techniques, SVD may be the most effective algorithm. But it is well known that SVD is time-consuming in computation. The DWT and DFT are good approximation of SVD, and both of them have fast computational algorithms. DFT maps the time-series data from the time domain to the frequency domain, and there exists a fast algorithm that can compute the DFT coefficients in $O(n\log n)$ time. Unlike the DFT takes the original signal from time domain and transforms it into frequency domain, the DWT transforms the time-series from time domain into time-frequency domain. Since the wavelet transform has the property of time-frequency *localization* of the time-series, it means most of the energy of the time-series can be represented by only a few wavelet coefficients. Moreover, if we use a special type of wavelets called *Haar* wavelet, we can achieve $O(n)$ time complexity (Most of the DWT transformation algorithms are implemented through DFT and have the same time complexity with DFT).

An ideal feature extraction has the capability of efficiently reducing the data into a lower-dimensional model, while preserving the properties of the original data. In practice, however, information is lost as the dimensionality is reduced. It is therefore desirable to formulate a method that reduces the dimensionality efficiently, while preserving as much information from the original data as possible. We propose a time-series feature extraction algorithm by using Haar wavelets. We define the features are the wavelet approximation coefficients within a specific scale. The problem of determination of feature dimensionality is circumvented by choosing the scale of wavelet transform. We take a tradeoff of lower dimensionality and lower errors by selecting the scale within which the detail coefficients have lower energy than that within the nearest higher scales.

The rest of this paper is organized as follows. Section 2 introduces our algorithm. The time complexity of the proposed algorithm is analyzed in Section 3. Section 4 contains a comprehensive experimental comparison of the K-NN algorithm with and without feature extraction. Section 5 concludes the paper with the summary of our contributions.

2 Wavelet-Based Feature Extraction Algorithm

Wavelet transform is a domain transform technique for hierarchically decomposing sequences. It allows a sequence to be described in terms of an approximation of the original sequence, plus a set of details that range from coarse to fine. The property of wavelets is that the broad trend of the input sequence is preserved in approximation part, whereas the localized changes are kept in detail parts. The number of decomposing scales for the input time series is $J = log_2(N)$, here N is the length of zero-padded input series. No information will be gained or lost during the decomposition process. The original signal can be fully reconstructed from the approximation part and the detail parts.

We use $\overrightarrow{A_0}$ to denote the original time-series $\overrightarrow{X}$. The approximation coefficients $\overrightarrow{A_j}$ and detail coefficients $\overrightarrow{D_j}$ within scale j are decomposed from $\overrightarrow{A_{j-1}}$, the approximation coefficients of scale $j-1$. Given $\overrightarrow{A_{j-1}} \in R^{2M}$, where M is the dimensionality of $\overrightarrow{A_j}$ and $\overrightarrow{D_j}$. The ith element of $\overrightarrow{A_j}$ is calculated as:

$$A_{i,j} = \frac{1}{\sqrt{2}}(A_{2i-1,j-1} + A_{2i,j-1}), i \in [1, 2, \ldots, M]. \tag{1}$$

The ith element of $\overrightarrow{D_j}$ is calculated as:

$$D_{i,j} = \frac{1}{\sqrt{2}}(A_{2i-1,j-1} - A_{2i,j-1}), i \in [1, 2, \ldots, M]. \tag{2}$$

After decomposing a time-series $\overrightarrow{X}$ at a specific scale $j \in [1, 2, \ldots, J]$, the wavelet coefficients $H(\overrightarrow{X})$ is represented by $\{\overrightarrow{A_j}, \overrightarrow{D_j}, \ldots, \overrightarrow{D_2}, \overrightarrow{D_1}\}$. From single processing point of view, the wavelet detail coefficients within higher scale correspond to lower frequency part of the signal. As the noises often exist in high frequency, we only need to keep first k ($k < N$) coefficients as the features [5]. We simply keep the whole approximation coefficients within a specific scale as the features.

Definition 1. *Given a time series $\overrightarrow{X}$, the features is the Haar wavelet approximation coefficients $\overrightarrow{A_j}$ decomposed from $\overrightarrow{X}$ within a specific scale $j(j > 0)$.*

We prefer the extracted features are similar to the original data as much as possible. A measurement for evaluating the similarity/dissimilarity between the features and the data is necessary. We use the widely used sum of squared errors (SSE) as the dissimilarity measure between a time-series and its approximation.

Definition 2. *Given a time-series $\overrightarrow{X} \in \mathbb{R}^N$ and its approximation $\widehat{\overrightarrow{X}} \in \mathbb{R}^N$, the Sum of Squared Errors between $\overrightarrow{X}$ and $\widehat{\overrightarrow{X}}$ is*

$$SSE(\overrightarrow{X}, \widehat{\overrightarrow{X}}) = \sum_{i=1}^{N}(x_i - \hat{x}_i)^2 \tag{3}$$

We hope to minimize the SSE between a time-series $\overrightarrow{X} \in \mathbb{R}^N$ and the extracted features $\overrightarrow{A_j} \in \mathbb{R}^M$ ($M < N$). As $M < N$, we can't calculate the SSE with Equ. (3) directly. One choice is to reconstruct a sequence $\widehat{\overrightarrow{X}} \in \mathbb{R}^N$ from $\overrightarrow{A_j}$ then calculate the similarity between $\overrightarrow{X}$ and $\widehat{\overrightarrow{X}}$. For instance, Kaewpijit et al. use correlation function of $\overrightarrow{X}$ and $\widehat{\overrightarrow{X}}$ to measure the similarity between $\overrightarrow{X}$ and

$\overrightarrow{A_j}$ [4]. Actually, we can prove that $SSE(\overrightarrow{X}, \widehat{\overrightarrow{X}})$ is the same as energy difference between $\overrightarrow{A_j}$ and $\overrightarrow{X}$ with Haar wavelet transform. This property lets it possible to design an efficient algorithm without reconstruction $\widehat{\overrightarrow{X}}$.

Definition 3. *Given a series $\overrightarrow{X} \in \mathbb{R}^N$, the Energy of $\overrightarrow{X}$ is:*

$$E(\overrightarrow{X}) = \sum_{i=1}^{N} X_i^2 \tag{4}$$

Definition 4. *Given a series $\overrightarrow{X} \in \mathbb{R}^N$, and its features $\overrightarrow{A_j} \in \mathbb{R}^M$, the Energy Difference (ED) between $\overrightarrow{X}$ and $\overrightarrow{A_j}$ is*

$$ED(\overrightarrow{X}, \overrightarrow{Y}) = \sum_{i=1}^{N} X_i^2 - \sum_{i=1}^{M} A_{i,j}^2 \tag{5}$$

Notice the $SSE(\overrightarrow{X}, \widehat{\overrightarrow{X}})$ will monotonically increase when increasing the scale of $\overrightarrow{A_j}$ (decreasing the dimensionality). We prefer lower dimensionality and lower SSE$(\overrightarrow{X}, \widehat{\overrightarrow{X}})$ at the same time, and these two objectives are in conflict. One must take a tradeoff between these two objectives. The $SSE(\overrightarrow{X}, \overrightarrow{A_j})$ is the sum of the energy of all discarded detail coefficients. Discarding the detail coefficients within a scale with lower energy ratio $(E(\overrightarrow{D_j})/\sum_{k=1}^{j} E(\overrightarrow{D_j}))$ will not decrease the SSE much. From noise reduction point of view, the noise normally locate in detail coefficients within finer scale (lower scale), and the energy of noise is much smaller than that of the *true signal* with wavelet transform. If the energy of the detail coefficients within a scale is small, their has much noise embedded in the detail coefficients, discarding the detail coefficients within this scale may remove more noise. Thus we leverage these two conflicted objectives by stopping the decomposition process in the scale j^*, when $E(\overrightarrow{D_{j^*}}) > E(\overrightarrow{D_{j^*-1}})$. The scale $j^* - 1$ is defined as the appropriate scale and the approximation coefficients in the scale $j^* - 1$ are kept as the appropriate features. Note that by this process, at least $\overrightarrow{D_1}$ will be removed, and the length of $\overrightarrow{D_1}$ is $\frac{N}{2}$. Thus the dimensionality of the features M will be $M \leq \frac{N}{2}$.

The pseudo code of the feature extraction algorithm is demonstrated in Algorithm 1 below.

3 Time Complexity Analysis

The time complexity of Haar wavelet transform is $2(N-1)$ proved in [5] bound by $O(N)$. Even in the extreme case of our algorithm (the selected scale $j = J$), except calculating the wavelet coefficients with time complexity $2N - 1$ and the energy of detail coefficients with time complexity N, we need to compare the $E(\overrightarrow{D_j})$ of different scales with time complexity $\log_2(N)$, the time complexity of the algorithm still bounded by $O(N)$.

Algorithm 1 The feature extraction algorithm

```
input: a time-series data set $X = \{\overrightarrow{X_1}, \overrightarrow{X_2}, \ldots, \overrightarrow{X_n}\}$
for $i = 1$ to $n$ do
  calculate $\overrightarrow{A_1}$ and $\overrightarrow{D_1}$ for $\overrightarrow{X_i}$
end for
calculate $\sum_n E(\overrightarrow{D_1})$
exitFlag = false
for $j = 2$ to $J$ do
  for $i = 1$ to $n$ do
    calculate $\overrightarrow{A_j}$ and $\overrightarrow{D_j}$ for $\overrightarrow{X_i}$
  end for
  calculate $\sum_n E(\overrightarrow{D_j})$
  if $\sum_n E(\overrightarrow{D_j}) > \sum_n E(\overrightarrow{D_{j-1}})$ then
    keep the $\overrightarrow{A_{j-1}}$ for $X$ as the appropriate features
    exitFlag = true
    exit loop
  end if
end for
if exitFlag == false then
  keep the $\overrightarrow{A_J}$ for $X$ as the appropriate features
end if
```

4 Experimental Evaluation

To show the effectiveness of our approach, we performed experiments on four publicly available data sets. We first evaluated the accuracy of classification on features within various scales to show the feasibility of using wavelet approximation coefficients within a specific scale as features. Then we compared the accuracy of classification with and without feature extraction. The used classification algorithm is one-nearest-neighbor algorithm (1-NN) with Euclidean distance, evaluated by *leave-one-out* cross validation. We used the datasets from the UCR Time Series Data Mining Archive [3]. We only took the classified data sets for experiments.

The accuracy of classification is measured by the classification error rates. The error rates of the 1-NN algorithm with A_j $(j \in [0, 1, \ldots, J])$ for all data sets are shown in Table 1. The classification error rates are different with various scales, and only Gun data has highest classification error rate on the original time-series (scale equals to 0). Dimensionality reduction can improve the classification accuracy on three datasets of the four datasets used.

The calculated appropriate scales for the four data sets are shown in Table 2. Compared with table 1, the appropriate features take the highest classification accuracy among features within various scales for all four data sets.

5 Conclusion

We have propose an algorithm of time-series feature extraction for classification by using Haar wavelets. The features are defined as the approximation Haar

Table 1. The classification error rates (%) for all four data sets

scale	CBF	CC	Trace	Gun
0	0.26	1.33	11	5.5
1	0.26	1	10.5	6
2	0	0.5	10.5	6
3	0.26	1	11.5	6
4	2.3	7.67	14	11.5
5	6.77	21.83	25	13.5
6	15.37	37.17	20.5	28
7	30.99		42.5	33
8			58	45
9			68	

Table 2. The selected appropriate scale and its corresponding classification error rate (%) for all four data sets

	CBF	CC	Trace	Gun
scale	2	2	1	1
error rate	0	0.5	10.5	6

wavelet coefficients within a specific scale. We leveraged the conflict of taking lower dimensionality and lower SSE simultaneously by finding the scale within which the energy of detail coefficients is lower than that within the nearest higher scale.

We conducted experiments on four time series data sets and compared the classification accuracy of the features within various scales and the original time series. The feature extraction algorithm successfully selected the scale with lowest error rate over all four data sets.

References

1. Agrawal, R., Faloutsos, C., Swami, A.: Efficient Similarity Search in Sequence Databases. In Proceedings of the 4th Conference on Foundations of Data Organization and Algorithms, (1993) 69-84
2. Korn, F., Jagadish, H., Faloutsos, C.: Efficiently Supporting ad hoc Queries in Large Datasets of Time Sequences. In: Proceedings of The ACM SIGMOD International Conference on Management of Data, (1997) 289-300
3. Keogh, E., Folias, T.,: The UCR Time Series Data Mining Archive. http://www.cs.ucr.edu/∼eamonn/TSDMA/index.html. Riverside CA. University of California - Computer Science & Engineering Department (2002)
4. Kaewpijit, S., Moigne, J. L., Ghazawi, T. E.: Automatic Reduction of Hyperspectral Imagery Using Wavelet Spectral Analysis. IEEE Trans. on Geoscience And Remote Sensing, **41** (2003) 863-871
5. Chan, K. P., Fu, A. W., Clement, T. Y.: Harr Wavelets for Efficient Similarity Search of Time-Series: with and without Time Warping. IEEE Trans. on Knowledge and Data Engineering, **15** (2003) 686-705

Prediction of Time Sequence Using Recurrent Compensatory Neuro-fuzzy Systems

ChiYung Lee[1] and ChengJian Lin[2,⋆]

[1] Dept. of Computer Science and Information Engineering
Nankai Institute of Technology
Nantou County, 542 Taiwan, China

[2] Dept. of Computer Science and Information Engineering
Chaoyang University of Technology
168 Gifeng E. Rd., Wufeng
Taichung County, 413 Taiwan, China
Tel: +886-4-23323000 Ext. 4408, Fax: +886-4-23742375
cjlin@mail.cyut.edu.tw

Abstract. In this paper, a recurrent compensatory neuro-fuzzy system (RCNFS) is proposed for prediction of time sequence. The compensatory-based fuzzy reasoning method is using adaptive fuzzy operations of neuro-fuzzy systems that can make the fuzzy logic systems more adaptive and effective. The recurrent network is embedded in the RCNFS by adding feedback connections in the second layer, where the feedback units act as memory elements. Also, an on-line learning algorithm is proposed to automatically construct the RCNFS. They are created and adapted as on-line learning proceeds via simultaneous structure and parameter learning.

1 Introduction

For a dynamic system, the output is a function of past input or past output or both, prediction of time sequence is not as direct as a static system, and to deal with temporal problem of dynamic system, the recurrent neural network and the recurrent neuro-fuzzy system have been attracting great interest [1]-[2].

In this paper, a recurrent compensatory neuro-fuzzy system (RCNFS) is proposed. The RCNFS is a recurrent multi-layer connectionist network for fuzzy reasoning and can be constructed from a set of fuzzy rules. In the RCNFS, adding feedback connections in the second layer develops the temporal relations. At the same time, the compensatory fuzzy inference method is using adaptive fuzzy operations of neuro-fuzzy system that can make the fuzzy logic system more adaptive and effective. An on-line learning algorithm is proposed to automatically construct the RCNFS. It consists of structure learning and parameter learning. The structure learning algorithm decides to add a new node which is satisfying the fuzzy partition of the input data. The back-propagation learning is then used for tuning input membership functions.

[⋆] Corresponding Author.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 611–617, 2005.

2 The Compensatory Operator

Zimmermann [3] first defined the essence of compensatory operations. Zhang and Kandel [4] proposed more extensive compensatory operations based on the pessimistic operation and the optimistic operation. The compensatory operation can map the pessimistic input x_1 and the optimistic input x_2 to make the relatively compromised decision for the situation between the worst case and the best case. For example, $c(x_1, x_2) = x_1^{1-r} x_2^r$, where $r \in [0, 1]$ is called the compensatory degree. Many researchers [5]-[6] have used the compensatory operation to fuzzy systems successfully. Therefore, we can define the fuzzy if-then rule as follows:

$$R_j : [\text{ IF } x_1 \text{ is } A_{1j} \cdots \text{ and } x_n \text{ is } A_{nj}]^{1-r_j+r_j/n}, \text{ THEN } y^{'} \text{ is } b_j \tag{1}$$

3 The Structure of RCNFS Model

The RCNFS realizes a fuzzy model of the following form:

$$Rule - j : [\text{ IF } h_{1j} \text{ is } A_{1j} \cdots \text{ and } h_{nj} \text{ is } A_{nj}]^{1-r_j+r_j/n}, \text{THEN } y^{'} \text{ is } w_j \tag{2}$$

where for $i = 1, 2, ..., n$, $h_{ij} = x_i + u_{ij}^{(2)}(t-1) \cdot \theta_{ij}$, $y^{'}$is output variable, A_{nj} is linguistic term of the precondition part, w_j is constant consequent part, and n is number of input variables. That is, the input of each membership function is the network input x_i plus the temporal term $u_{ij}^{(2)}\theta_{ij}$. Therefore, the fuzzy system, with its memory (terms feed-back units), can be considered a dynamic fuzzy inference system.

Next, we shall introduce the operation functions of the nodes in each layer of the RCNFS model are described. In the following description, $u^{(l)}$ denotes output of a node in the lth layer.

$Layer1(InputNode)$: No computation is done in this layer. Each node in this layer is an input node, which corresponds to one input variable, only transmits input values to the next layer directly.

$$u_i^{(1)} = x_i \tag{3}$$

$Layer2(InputTermNode)$: Nodes in this layer correspond to one linguistic label of the input variables in Layer1 and a unit of memory, i.e., the membership value specified the degree to which an input value and a unit of memory belongs a fuzzy set is calculated in Layer 2. The Gaussian membership function, the operation performed in Layer 2 is

$$u_{ij}^{(2)} = exp\{-\frac{[h_{ij} - m_{ij}]^2}{\sigma_{ij}^2}\} \tag{4}$$

where m_{ij} and σ_{ij} are, respectively, the mean and variance of Gaussian membership function of jth term of ith input variable x_i. In addition, the inputs of this layer for discrete time t can be defined by $h_{ij}(t) = u_i^{(1)}(t) + u_{ij}^{(2)}(t-1) \cdot \theta_{ij}$, where $u_{ij}(t-1)$ denotes the feedback unit of memory, which store the past information of the system, and θ_{ij} denotes the link weight of the feedback unit.

$Layer3(CompensatoryRuleNode)$: Nodes in this layer represents the precondition part of one fuzzy logic rule. And they receive the one-dimensional membership degrees of the associated rule from nodes of a set in Layer 2. Here we use a compensatory operator mentioned to perform IF-condition matching of fuzzy rules. As a result, the output function of each inference nodes is

$$u_j^{(3)} = [\prod_i u_{ij}^{(2)}]^{1-r_j+\frac{r_j}{n}} \tag{5}$$

where the $\prod_i u_{ij}^{(2)}$ of a rule node represents the firing strength of its corresponding rule, $r_j \in [0,1]$ is called the compensatory degree. By tuning r_j, the compensatory operator becomes more adaptive.

$Layer4(OutputNode)$: This layer acts a defuzzifier. The node in this layer is labeled Σ and its sums all incoming signals to obtain the final inferred result

$$u_k^{(4)} = \sum_j u_j^{(3)} w_{jk} \tag{6}$$

where the weight w_{jk} is output action strength of the kth output associated with the jth rule and $u_k^{(4)}$ is the kth output of the RCNFS.

4 The On-Line Learning Algorithm

In this section, we present an on-line learning algorithm for constructing the RCNFS. The proposed learning algorithm consists of structure learning phase and parameter learning phase. The structure learning is based on the degree measure to determine the number of fuzzy rules. The parameter learning is base upon supervised learning algorithms.

4.1 The Structure Learning Phase

The first step in the structure learning is to determine whether or not to extract a new rule from training data as well as the number of fuzzy sets on the universal of discourse of each input variable. Since one cluster in the input space corresponds to one potential fuzzy logic rule, with m_{ij} and σ_{ij} representing the mean and variance of that cluster. For each incoming pattern x_i the strength a rule is fired can be interpreted as the degree the incoming pattern belongs to the corresponding cluster. For computational efficiency, we can use compensatory operation of the firing strength obtained from $[\prod_i u_{ij}^{(2)}]^{1-r_j+\frac{r_j}{n}}$ directly as this degree measure

$$F_j = [\prod_i u_{ij}^{(2)}]^{1-r_j+\frac{r_j}{n}} \tag{7}$$

where $F_j \in [0, 1]$. Using this degree measure, we can obtain the following criterion for the generation of a new fuzzy rule of new incoming data is described as follows. Find the maximum degree

$$F_{max} = max_{1 \leq j \leq R_{(t)}} F_j \tag{8}$$

where $R_{(t)}$ is the number of existing rules at time t. If $F_{max} \leq \overline{F}$, then a new rule is generated where $\overline{F} \in (0, 1)$ is a prespecified threshold that decays during the learning process. Once a new rule is generated, the next step is to assign initial mean, variance, and weight of feedback for the new membership function. Since our goal is to minimize an objective function and the mean, variance, and weight of feedback are all adjustable later in the parameter learning phase. Hence, the mean, variance, and weight of feedback for the new membership function are set as follow: $m_{ij}^{(R_{(t+1)})} = x_i$, $\sigma_{ij}^{(R_{(t+1)})} = \sigma_{init}$, and $\theta_{ij}^{(R_{(t+1)})} = random$, where x_i is the new input data and σ_{init} is a prespecified constant.

The whole algorithm for the generation of new fuzzy rules as well as fuzzy sets in each input variable is as follows. Suppose no rules are existent initially:

```
Step 1: IF $x_i$ is the first incoming pattern THEN do
    { Generate a new rule
    with mean $m_{i1} = x_i$, variance $\sigma_{i1} = \sigma_{init}$, weight of feedback $\theta_{i1} = random$,
    compensatory degree $c_1 = random$, $d_1 = random$, weight $w_1 = random$
    where $\sigma_{init}$ is a prespecified constant.
    }
Step 2: ELSE for each newly incoming $x_i$, do
    { Find $F_{max} = max_{1 \leq j \leq R_{(t)}} F_j$
        IF $F_{max} \geq \overline{F}$
            do nothing
        ELSE
            $R_{(t+1)} = R_{(t)} + 1$ generate a new rule
            with mean $m_{ij}^{(R_{(t+1)})} = x_i$ , variance $\sigma_{ij}^{(R_{(t+1)})} = \sigma_{init}$ , weight of
            feedback $\theta_{i1}^{(R_{(t+1)})} = random$, compensatory degree $c_j^{(R_{(t+1)})}$ =random,
           $d_j^{(R_{(t+1)})}$ = random, weight $w_j^{(R_{(t+1)})}$ = random
            where $\sigma_{init}$ is a prespecified constant.
    }
```

4.2 The Parameter Learning Phase

After the network structure is adjusted according to the current training pattern, the network then enters the parameter learning phase to adjust the parameters of the net-work optimally based on the same training pattern. The learning process involves the determination of minimize a given cost function. The gradient of the cost function is computed and adjusted along the negative gradient. The idea of backpropagation algorithm is used for this supervised learning method. Considering the single output case for clarity, our goal is to minimize the cost function E is defined as

$$E = \frac{1}{2}[y - y^d]^2 \tag{9}$$

where y^d is the desired output and y is the current output. Using the steepest-descent gradient approach, the learning rule for a network weight in any one of the network layers is given by

$$\Delta W = -\frac{\partial E}{\partial W} \tag{10}$$

The weight is updated according to the following equation:

$$W(t+1) = W(t) + \eta_w \Delta W \tag{11}$$

where factor η_w is the learning rate parameter of the weight and t denotes the iteration number.

5 Prediction of Time Sequence

To clearly verify if the proposed RCNFS can learn the temporal relationship, a simple time sequence prediction problem found in [7] is used for test in the following example. The test bed used is shown in Figure 2(a). This is an 8 shape made up of a series with 12 points which are to be presented to the network in the order as shown. The RCNFS is asked to predict the succeeding point for every presented point. Obviously, a static network cannot accomplish this task, since the point at coordinate (0,0) has two successors: point 5 and point 11. The RCNFS must decide the successor of (0,0) based on its predecessor; if the predecessor is 3, then the successor is 5, whereas if the predecessor is 9, the successor is 11.

In this example, the RCNFS contains only two input nodes, which are activated with the two dimensional coordinate of the current point, and two output nodes, which represent the two dimensional coordinate of the predicted point. The learning rate $\eta_w = \eta_c = \eta_d = \eta_m = \eta_\sigma = \eta_\theta = 0.05$, and the prespecified threshold are chosen. After training, a root-mean-square (rms) error of 0.000237 is achieved, and the predicted values with 12 fuzzy logic rules ($\sigma = 0.08$) of RCNFS are shown in Figure 2(b). Simulation results show that we can obtain perfect prediction capability. Figure 2(c) shows the prediction results using the RFNN model [2]. In this figure, the RFNN also obtain prediction capability, but some time prediction points cannot be matched exactly. Figure 2(d) shows that a feedforward fuzzy neural network cannot predict successfully. Figure 2(e) shows the learning curves of the RCNFS model, the RFNN model and the FNN model. From the simulation results shown in Figure 2(d), we can see that the FNN is inappropriate for time sequence prediction because of its static mapping. To give a clear understanding of this performance comparison with the RFNN [2] and FNN [8] on the same problem is made in Table 1. Although the RCNFS needs more adjustable parameters than RFNN and FNN under the same fuzzy rules required, our model could obtain a smaller rms error and converge quickly.

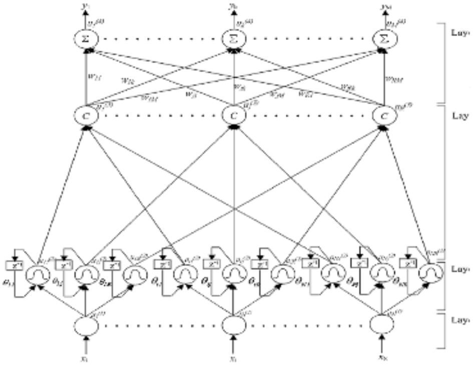


Fig. 1. Structure of the proposed RCNFS

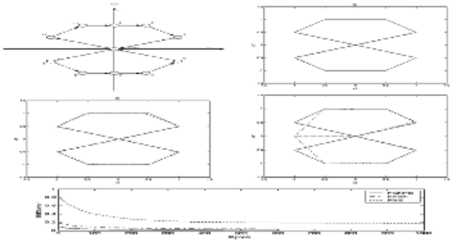
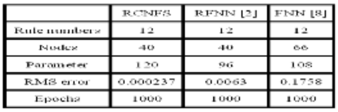


Fig. 2. Simulation results of time sequence prediction

Table 1. Performance comparison of various existing models



	RCNFS	RFNN [2]	FNN [8]
Rule numbers	12	12	12
Nodes	40	40	66
Parameter	120	96	108
RMS error	0.000237	0.0063	0.1758
Epochs	1000	1000	1000

6 Conclusion

A recurrent compensatory neuro-fuzzy system (RCNFS) is proposed in this paper. The compensatory operators are used to optimize fuzzy logic reasoning and select optimal fuzzy operators. Therefore, an effective neuro-fuzzy system should be able not only to adaptively adjust fuzzy membership functions but also to dynamically optimize adaptive fuzzy operators. An on-line learning algorithm is proposed to per-form the structure learning and the parameter learning. The simulation results show that the proposed learning algorithm converges quickly and requires a small number of tuning parameters.

References

1. Narendra, K. S., Parthasarathy, K.: Identification and Control of Dynamical Systems Using Neural Networks. IEEE Trans. on Neural Networks, **1** (1990) 4-27

2. Lee, C. H., Teng, C. C.: Identification and Control of Dynamic Systems Using Recurrent Fuzzy Neural Networks. IEEE Trans. on Fuzzy Systems, **8** (2000) 349-366
3. Zimmermann, H. J., Zysno, P.: Latent Connective in Human Decision. Fuzzy Sets and Systems, **4** (1980) 31-51
4. Zhang, Y. Q., Kandel, A.: Compensatory Neurofuzzy Systems with Fast Learning Algorithms. IEEE Trans. on Neural Networks, **9** (1998) 83-105
5. Lin, C. J., Chen C. H.: Nonlinear System Control Using Compensatory Neuro-Fuzzy Networks. IEICE Trans. On Fundamentals of Electronics, Communications and Computer Sci-ences, E86-A (2003) 2309-2316
6. Lin, C. J., Ho, W. H.: A Pseudo-Gaussian-Based Compensatory Neural Fuzzy System. Proceedings of the IEEE International Conference on Fuzzy Systems (2003)
7. Santini, S., Bimbo, A. D., Jain, R.: Block-Structured Recurrent Neural Networks. Neural Networks, **8** (1995) 135-147
8. Chao, C. T., Chen, T. J., Teng, C. C.: Simplification of fuzzy-neural systems using similarity analysis. IEEE Trans. Syst., Man, Cybern., pt. B. **26** (1996) 344-354

Study of Nonlinear Multivariate Time Series Prediction Based on Neural Networks

Min Han[1], Mingming Fan[1], and Jianhui Xi[1,2]

[1] School of Electronic and Information Engineering, Dalian University of Technology
Dalian, Liaoning 116023, China
minhan@dlut.edu.cn
[2] Department of Automation, Shenyang Institute of Aeronautical Engineering
Shenyang, Liaoning 110034, China

Abstract. A new method is brought forward to predict multivariate time series in this paper. Related time series instead of a single time series are applied to obtain more information about the input signal. The input data are embedded as the phase space points. By the Principle Component Analysis (PCA) the most useful information is extracted form the input signal and the embedding dimension of the phase space is reduced, consequently, the input of the neural networks is simplified. The recurrent neural network has a number of advantages for predicting nonlinear time series. Therefore, Elman neural network is adopted to predict multivariate time series in this paper. Simulations of nonlinear multivariate time series from nature and industry process show the validity of the method proposed.

1 Introduction

Variables in Multivariate time-varying processes are changing simultaneously as time goes by, moreover, different variables work on each other. The purpose to predict multivariate time series is to know about the evolution of the system. Though the multivariate time series processes are common in a lot of domains, they are less frequently studied. In this paper, neural network is applied to predict multivariate time series of complicated system.

Neural network has gain popularity for predicting multivariate nonlinear time series with the development of computer technology, such as capital markets and individual stock prices [1] and monthly temperatures from different sites [2]. Recurrent neural network is a specific type of neural networks. Since the activity pattern passes through the network more than once before it generates an output, RNN exhibits some intrinsic dynamics which is capable of performing more complex computations [3]. RNN has shown to be very useful in predicting nonlinear chaotic time series [4] and ARMA time series. It not only has the capability for handling a system of much higher complexity, but its superiority in time convergence can prove to be a valuable asset for time critical application [5]. In this paper, Elman recurrent neural network is applied to analyze the nonlinear multivariate time series in nature and industry process. Nowadays Principal Component Analysis (PCA) has been a powerful methodology for a wide variety of application. The principal component is the respective ei-

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 618–623, 2005.

genvectors corresponding to the largest eigenvalules of the autocorrelation matrix of the input signals [6]. And in this paper PCA is applied to extract adaptively the principal components from the input signal and consequently set a low dimension model of a system by statistical principle. And the performances of the prediction are evaluated by Root Mean Square Error E_{RMSE} and Prediction Accuracy E_{PA} .

2 Predicting Multivariate Time Series Using Recurrent Time Series

The algorithm is based on multivariate chaotic time series set $x_1(1), x_1(2),...x_1(N), x_2(1), x_2(2),...\ x_2(N),...x_n(1), x_n(2),...x_n(N)$, where N is the length of the data set, n is the dimension of the multivariate time series. If the embedding dimension of each chaotic time series is selected as D_i , $i = 1, 2,...n$, and the time delay as $\tau_i, i = 1, 2,...n$, so there are L phase space points $X_i(1), X_i(2),...X_i(L)$, are generated in the phase space, $i = 1,2,...n$, $L = N - \max_i[(D_i - 1)\tau_i]$.

$$\begin{aligned}
&X_1(1) = [x_1(1), x_1(1+\tau_1),..x_1(1+(D_1-1)\tau_1)]^T, X_2(1) = [x_2(1), x_2(1+\tau_2),..x_2(1+(D_2-1)\tau_2)]^T,...\\
&X_n(1) = [x_n(1), x_n(1+\tau_n),..x_n(1+(D_n-1)\tau_n)]^T;\\
&......\\
&X_1(L) = [x_1(L), x_1(L+\tau_1),..x_1(N)]^T, X_2(L) = [x_2(L), x_2(L+\tau_2),..x_2(N)]^T,...\\
&X_n(L) = [x_n(L), x_n(L+\tau_n),.....x_n(N)]^T
\end{aligned} \tag{1}$$

Then PCA is applied to modify the phase space, take the nonlinear time series $\vec{X}_1$ for example, the process is as following:

(1) let

$$\vec{X}_1 = [X_1(1), X_1(2),...X_1(L)]_{D_1 \times L} \ . \tag{2}$$

Where D_1 is the embedding dimension of nonlinear time series $x_1(1), x_1(2),...x_1(N)$

(2) Standardize the sample matrix $\vec{X}_1$ to $\tilde{X}_{D_1 \times L}$;

(3) Deal with $\tilde{X}_{D_1 \times L}$ based on singular value decomposition as equation (3)

$$\tilde{X} = U \Sigma V^T \ . \tag{3}$$

Where

$$\Sigma = diag\ [s_1\ s_2\ \dots\ s_p\ 0\dots0]. \tag{4}$$

P denotes the number of eigenvalue of $\tilde{X}_{D_1 \times L}$, $s_1 \geq s_2 \geq \cdots s_p$ are corresponding eigenvalue of $\tilde{X}_{D_1 \times L}$, U and V are both orthogonal matrixes. Where,

$U \in F^{D_1 \times D_1}, \Sigma \in F^{D_1 \times L}, V \in F^{L \times L}$

(4) The larger singular values are preserved as principle components. η_{k1} is the threshold value as equation (5), where $k1 = 1, 2, \cdots$,

$$\eta_{k1} = \sum_{j=1}^{k1} s_j^2 \Big/ \sum_{i=1}^{p} s_i^2 \quad . \tag{5}$$

Where, both s_i and s_j denote the corresponding eigenvalue. Choosing $0 < \eta_0 < 1$, if threshold $\eta_{k1} > \eta_0$, the first $k1$ singular values are preserved.

(5) The matrix $U_{D_1 \times D_1}$ is replaced by $\tilde{U}_{k1 \times D_1}$, which includes the first $k1$ row of $U_{D_1 \times D_1}$, then the principal component matrix is as equation (6),

$$\vec{Z}_{k1 \times L} \approx \tilde{U}_{k1 \times D_1} \vec{X}_{D_1 \times L} \quad . \tag{6}$$

With the above method, the principal component matrixes of other chaotic time series $x_2(1), x_2(2),...x_2(N),...x_n(1), x_n(2),...x_n(N)$ obtained as $\vec{Z}_{k2 \times L}, \vec{Z}_{k3 \times L},...\vec{Z}_{kn \times L}$, thus the final input matrix $\vec{Z}_{K \times L} = [Z_{k1 \times L}; Z_{k2 \times L};...Z_{kn \times L}]$, $K = k1 + k2 + ...kn$. Therefore, there are L embedding phase space points, they are $X(1), X(2), \cdots X(L)$, the RNN is trained using those obtained points. Assume $X(1) \sim X(M)$ as the input points and $X(M+T)$ as the outputs to train the RNN, where T is the prediction step, M is input number of RNN and $M < L$, n is the dimension of the multivariate time series, that means there are n outputs of the RNN.

In this paper, Elman network is the recurrent neural network model, which is shown in Fig.1.

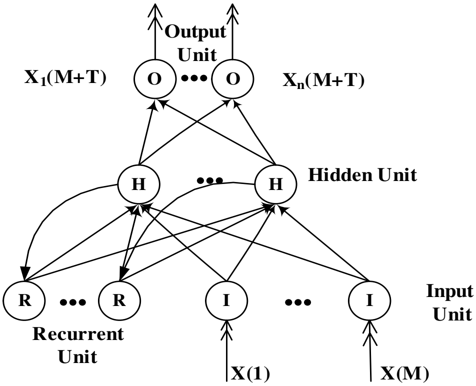


Fig. 1. The structure of Elman Recurrent Neural networks

There are four layers in Fig.1: input layer, hidden layer, output layer and recurrent layer in a basic Elman network. It is obvious that the Elman network differs from conventional two-layer networks in that the first layer has a recurrent connection. The recurrent connection allows Elman network to both detect and generate time-varying patterns. Elman network applied in this paper has tansig neurons in both hidden and output layers. The training function is resilient backpropagation algorithm as well as the Gradient descent with momentum weight and bias learning function is adopted.

In order to measure the network's performance, Root Mean Square Error E_{RMSE} and Prediction accuracy E_{PA} as equation (7) and equation (8) are adopted.

$$E_{RMSE} = \left(\frac{1}{N-1} \sum_{t=1}^{S} [\hat{y}(t) - y(t)]^2 \right)^{1/2} . \tag{7}$$

$$E_{PA} = \sum_{t=1}^{S} [(\hat{y}(t) - \hat{y}_m)(y(t) - y_m)] \Big/ (N-1)\sigma_{\hat{y}}\sigma_y . \tag{8}$$

$y(t)$ is the target value of certain variable, $\hat{y}(t)$ is the predicted value, y_m and σ_y denote averaged practical value and standard deviation of $y(t)$ correspondingly, $\hat{y}_m$ and $\sigma_{\hat{y}}$ are respectively the averaged practical value and standard deviation of $\hat{y}(t)$. E_{RMSE} can be used to describe how well the model accounts for the variation in the observed data. E_{PA} represents correlation coefficient between observed and predicted time series. In ideal situation, if there were no error in prediction, these parameters would indicate like that $E_{RMSE} = 0; E_{PA} = 1$.

3 Simulations on Predicting Multivariate Time Series

Simulations in different domains are adopted to show the validity of the new method.

A. Predicting Time Series of Sunspots and Runoff of the Yellow River Annually

The annual runoff of the Yellow River is influenced by the climate of the earth such as rainfall, and temperature. The studies show that the sunspots play an important role to change the climate of the earth, in other words, the activity of sunspots will affect the annual runoff of the river. The two-dimension time series of annual sunspots and runoff of the Yellow river from year 1753 to 1997 is adopted to train the Elman network, it means $N = 245$. All the studies show that the number of sunspots varies with a cycle of 11 years. So assume that $D_1 = D_2 = 11$, and $\tau_1 = \tau_2 = 1$, then $L = 234$. Both η_{k1} and η_{k2} are chosen as 0.9, then $k1 = 9; k2 = 6$ are obtained. So the original input matrix $\vec{X}_{22\times L}$ is replaced by the simplified input matrix $\vec{Z}_{15\times L}$, the first 232 points of the 244 points are chosen as training set and the other points as test set, the performances of the runoff and sunspots prediction are respectively as Fig.2 and Fig.3. Corresponding value of E_{PA} and E_{RMSE} are shown in Table.1.

From the figures and the data of table 1, it is obvious that the prediction of the sunspots and runoff approximate the observed data by using the new methodology.

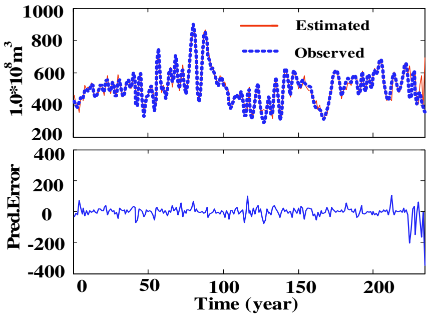


Fig. 2. Estimated and observed curves of Yellow River runoff time series and error

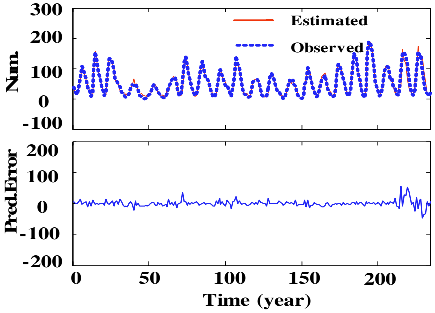


Fig. 3. Estimated and observed curves of sunspots time series and error

Table 1. Predict performance of annual Yellow River runoff and sunspots series (1753~1997)

Time series	E_{RMSE}	E_{PA}
Runoff	176.0531	0.9387
Sunspots	30.2931	0.9732

B. Predicting Quantity of Burnt Firedamp and Percentage of the Obtained Carbon Dioxide

In the industrial process, carbon dioxide is obtained when the firedamp is burning. If the percentage of the carbon dioxide is needed, used firedamp should be considered.

The two-dimension time series of the quantity of burnt firedamp and the percentage of the obtained carbon dioxide in certain timeslice are adopted to train the Elman network, and there are 23 nodes in the hidden units. $N = 240$. Assume that $D_1 = D_2 = 5$, $\tau_1 = \tau_2 = 1$, $L = N - \max_i[(D_i - 1)\tau_i] = 236$. After PCA, $k1 = 2; k2 = 2$, the first 109 points of the 236 points are chosen as the training set and the other 127 points as the test set, the performances of the prediction are respectively is as Fig.4. Corresponding value of E_{PA} and E_{RMSE} are shown in Table 2.

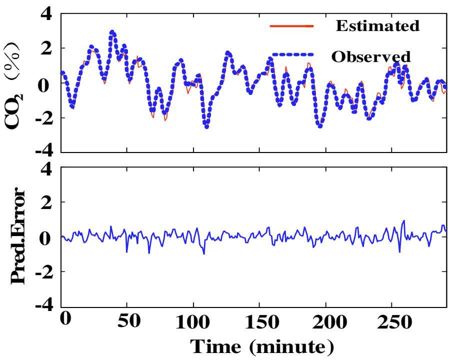


Fig. 4. Estimated and observed curves of carbon dioxide time series and error

Table 2. Predict performance of firedamp and carbon dioxide

Time series	E_{RMSE}	E_{PA}
CO_2(%)	0.2704	0.9682

The Fig.4 shows that the estimated curve approximate the observed curve. In Table 2, E_{RMSE} is 0.2704, it means the prediction error is slim, and E_{PA} is 0.9682, it also shows that the estimated curve is close to the observed curve. It could be obtained from the simulation that the result of the new methodology is excellent to predict the percentage of the carbon dioxide.

4 Conclusion

This paper focuses on the study of predicting the chaotic multivariate time series . A new method to predict the multivariate time series is proposed. Embedding phase space and PCA are adopted to simplify the input data of the neural network and recurrent neural network is applied to train the net. The simulation of the multivariate time series in nature and industry process show the validity of the method proposed.

Acknowledgements

This research is supported by the project (60374064) of the National Natural Science Foundation of China. It is also supported by the project (50139020) of the National Natural Science Foundation of China. All of these supports are appreciated.

References

1. Gan, W.-S., Ng, K.H.: Multivariate FOREX Forecasting Using Artificial Neural Networks. Proc. IEEE International Conference on Neural Networks, **2** (1995) 1018-1022
2. Valdes, J.J., Barton A.J.: Multivariate Time Series Model Discovery with Similarity-Based Neuro-Fuzzy Networks and Genetic Algorithms. Proceedings of the International Joint Conference on Neural Networks, **3** (2003) 1945-1950
3. Zhang, J., Tang, K.S., Man, K.F.: Recurrent NN Model for Chaotic Time Series Prediction. 23rd International Conference on Industrial Electronics, Control and Instrumentation. IECON, **3** (1997) 1108-1112
4. Han, M., Xi, J., Xu, S., Yin, F.-L.: Prediction of Chaotic Time Series Based on the Recurrent Predictor Neural Network**.** IEEE Transactions on Signal Processing, **52** (2004) 3409-3416
5. Zhang, J., Man, K.F.: Time Series Prediction Using RNN in Multi-Dimension Embedding Phase Space. IEEE International Conference on Systems, Man, and Cybernetics, **2** (1998) 1868-1873
6. Zhang, Q.F., Leung ,Y. W.: A Class of Learning Algorithms for Principal Component Analysis and Minor Component Analysis. IEEE Transactions on Neural Networks, **11** (2000) 200–204

Improving Ability of Passive Attacks of Chaotic Encryption by Using Neural Network

Xin Yang, Xiyue Huang, and Hanmin Huang

Automation College, Chongqing University, Chongqing 400044, China
luck_oo_sky@yahoo.com.cn

Abstract. Chaotic encryption have been widely studied and applied in cryptology field because its good randomicity and sensitivity to initial value. But nowadays it can't effectively defend passive attacks from outer world under development of hi-tech. Neural network be used into chaotic algorithm for its mnemonic function and self-study function can improve anti-passive-attacks ability of chaotic security system. This method can change length of secret-key at will also, which made the security system more flexible and practical.

1 Introduction

The chaotic phenomenon that systemic trochoid depend on precision of initial value are wildly applied in many fields[1]. Especially in information security and cryptology area, chaotic phenomenon are regarded as an important encryption technique because of its good randomicity and sensitivity to initial value. But nowadays the encryption techniques single used chaotic algorithm cannot defend attacks based on modern science and technology from outer. Through adding the trained neural network into chaotic encryption to extend secret-keys space, the encryption system can resist passive-attacks effectively without influencing the encrypting speed.

2 Passive-Attacks to Chaotic Logistic Algorithm

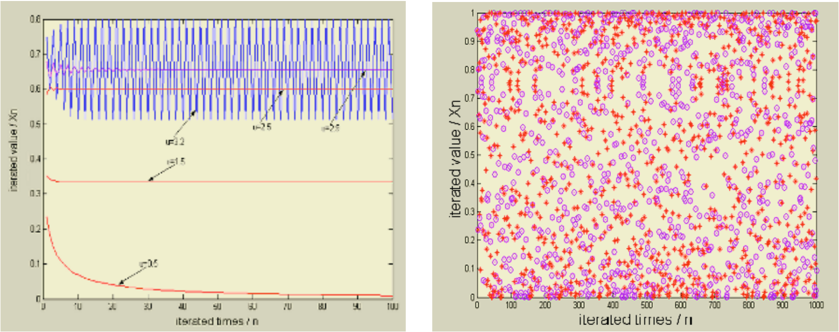
2.1 Chaotic Logistic Algorithm

In all encryption algorithm, classical one-time-pad encryption technique was recognized a perfect encryption and was the only one that can be proved a security encryption technique in mathematics. What ever other encryptions and devices were thought secure and reliable, but they all couldn't prove these through mathematics and only be concluded by counterexample. Chaotic Logistic algorithm can generate a fake random sequence stream, which have good randomicity, statistics characteristic and can be reproduced. The process that random sequence stream XOR with plaintexts to produce ciphertexts realize one-tine-pad technique in binary world. Discrete dynamic time system Logistic map defined as below [2]:

$$x_{n+1} = \mu x_n (1 - x_n) \qquad 0 < x_n < 1 \text{ , } 0 < \mu < 4 \text{.} \tag{1}$$

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 624–629, 2005.

x_n is state of system, μ is parameter, when dynamic system enter into iterative state with initial value x_0 ,there can generate a chaotic sequence stream through adjust parameter μ . Schuster deduced that when $\mu_\infty < \mu < 4$ and $\mu_\infty = 3.5699456$, Logistic map was in chaotic state, and the period of sequence approached infinite. Fig. 1(a) show that when μ gradually approach to μ_∞ in chaotic Logistic map, the iterative sequence gradually approach to chaotic state. And Fig. 1(b) show the sensitivity of initial value that nuance difference in initial value will result in two completely different chaotic sequences, where * is iterative dots when $\mu = 4$ and $x_0 = 0.6258$, ⊙ is iterative dots when $\mu = 4$ and $x_0^{'} = 0.6259$. So we can combine parameter μ and initial value x_0 as secret-keys of encryption.



(a) Iteration sequence in different μ

(b) Sensitive of chaos

Fig. 1. Characteristic of chaotic Logistic map

2.2 Passive-Attacks to Chaotic Logistic Algorithm

The main threatens to information's confidentiality in security system are passive-attacks. In passive-attacks, the attacker only spy the PDU (Protocol Data Unit) and warrant through "combine" [3]. Chaotic Logistic map' security is based on secret key, and details of encryption are published. The encryption precision lay on the space of secret-keys, and the attackers mostly adopted end-search secret-keys to attack chaotic Logistic encryption. To improve Chaotic Logistic map's anti-attack ability need increase encryption precision, but increase precision must occupy more system resource and consume more time. Thus the method that just depends on increase precision is unpractical and cannot defend passive-attacks effectively [4] .

3 Establishment of Neural Network

3.1 Architecture of Elman Neural Network

Elman neural network commonly is a several-layer network with feedback from the rear-layer output to the fore layer input. This recurrent connection allows the Elman network to both detect and generate time-varying patterns. The hidden (recurrent) layers are more big, and the running speed is more slow, so there adopted the neural network with one hidden layer as Fig.2. The function of feedback made inner nerve cell unknowable to exterior.

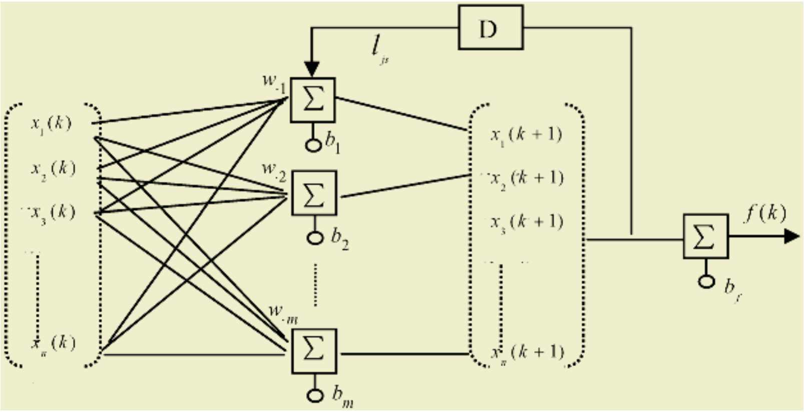


Fig. 2. Architecture of Elman neural network

And the principle of Elman neural network is as equation (2), $f(\bullet)$ is the output

$$f\left(x_i(k)\right)=\frac{\sum_{i,j}^{n,m} x_i(k-1)w_{(ij)}+b_j}{1+\sum_{i,j}^{n,m} x_i(k-1)w_{(ij)}\sum_{i,j,g}^{n,m,s} x_i(k-1)l_{(ij)}d_g}+b_f\,. \qquad (2)$$

of neural network, $x_i(k)$ is the input vector in input layer, n is the amount of input vectors, m is the amount of nerve cells in hidden layer, s is the amount of output vectors, $w_{(ij)}$ is the connection weight between input vectors and nerve cells, b_j is the biases of nerve cells , $l_{(ij)}$ is the feedback weight between feedback and nerve cells, d_g is feedback vector , b_f is the biases of output layer.

3.2 Training of Elman Neural Network

If training samples of neural network is $X'(0)=\left[x_1'(0) \quad x_2'(0) \quad \ldots\ldots \quad x_n'(0)\right]$,initial connection weights is $W'(0)=\begin{bmatrix} w_{1,1}' & w_{1,2}' & \cdots & w_{1,m}' \\ w_{2,1}' & w_{2,2}' & \cdots & w_{2,m}' \\ \cdots & \cdots & \cdots & \cdots \\ w_{n,1}' & w_{n,2}' & \cdots & w_{n,m}' \end{bmatrix}$, feedback matrix

$D = [d_1 \quad d_2 \quad ...... \quad d_m]^T$,feedback weights is $L' = \begin{bmatrix} l'_{1,1} & l'_{1,2} & \cdots & l'_{1,m} \\ l'_{2,1} & l'_{2,2} & \cdots & l'_{2,m} \\ \cdots & \cdots & \cdots & \cdots \\ l'_{n,1} & l'_{n,2} & \cdots & l'_{n,m} \end{bmatrix}$, output biases is $b_f = sign(f)$, then the training principle is :

$$W(k) = X(k)DL'W'(0) = \sum_{i,j}^{n,m} x_i d_j l'_{i,j} w'_{i,j} . \tag{3}$$

$$X(k+1) = \frac{X(k)W(k)}{1+X(k)W(k)} + b_j = \frac{\sum_{i,j}^{n,m} x_i(k) w_{(ij)}}{1+\sum_{i,j}^{n,m} x_i(k) w_{(ij)}} + b_j . \tag{4}$$

$$b_j = sign(f) = \begin{Bmatrix} 1 & \geq 0 \\ 0 & < 0 \end{Bmatrix} . \tag{5}$$

This neural network is special in that different initial weights and biases correspond to different inner relationship, and even if two Elman networks, with the same weights and biases, are given identical inputs at a given time step, their outputs can be different due to different feedback states in training. Thus Elman neural network can increase space of secret-keys and randomicity of selection to secret-keys. Furthermore feedback function hide relationship of nerve cells and make the hidden layer can be seen as a black box, attacker don't know how to get hands in it.

3.3 Cooperation Between Chaotic Encryption and Elman Neural Network

Fig.3 show the connection between chaotic Logistic algorithm and Elman neural network. First step is input initial value x and parameter μ to Logistic sequence generator, and set up the precision as p . Second step is to startup Logistic sequence generator to generate n entries iterative value as inspiring initial vector of neural network. Then the trained neural network generate q bit output as secret-keys stream and feedback input. Last the secret-keys stream XOR with plaintexts and give birth to ciphertexts [5].

4 Elman Neural Network Improve Ability of Passive Attacks of Chaotic Encryption

Elman neural network improve ability of passive attacks of chaotic Encryption from three aspects explained as below:

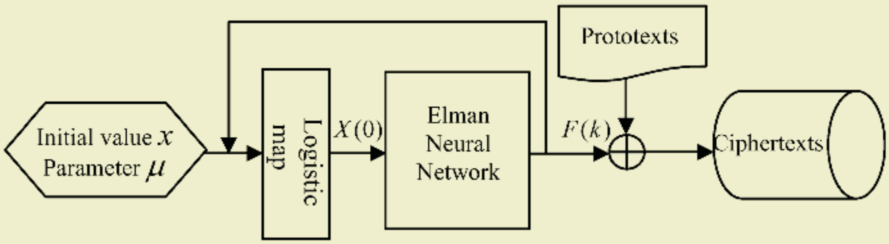


Fig. 3. Cooperation between chaotic encryption and Elman neural network

(1) The training and studying of Elman neural network increase randomicity of whole encryption system

When encryption systems are been used, the neural networks must be trained before. Random samples are selected to training neural networks, and different random samples will form different neural network. There random samples include Initial connection weights, feedback weights, initial input and training time, which insure the secret keys equiprobably distribute in a space that is big enough for attacker to waste a lot of cost and time that beyond importance of information they want to get.

(2) Extend secret keys' space, and reduce weak secret keys

Nowadays, attacks can use one or more computers connected by net to try every possible secret-keys of encryption system in little time. In chaotic Logistic encryption, if initial value and parameter were 16bit,there would have $2^{16\times2}$ possible secret-keys, attack online source site could easily capture the real secret-keys and unencrypt cryptograph. Neural network can extend secret-keys space, in figure3, amount of possible secret keys is $2^{2p+n\times m+m\times s}$. If attackers could try one trillion secret-keys in one second and need time T to try all possible secret-keys, the table 1show the comparison between chaotic Logistic encryption system and improved system by neural net work

(3) Changeable length of secret keys

Security of encryption system is a relative specification; require that the cost attacker wasted in unencrypting cryptograph is much more than encrypting information. So the length of secret-keys in encryption system showed in figure3 can be changed according to resource and requirements users had. The feedback channel in neural network could opened just in training in order to save system resource and encrypting time. These design are more flexible and practical.

5 Conclusions

To apply neural network in chaotic Logistic map is a better way in improving anti-passive-attacks ability of encryption system. This method could extend space of secret-keys largely, and the length of secret-keys could be changed, all these added the availability and practicability of encryption system without reducing security and reliability of encryption system.

Table 1. Comparison between chaotic Logistic encryption system and improved system by neural net work

Parameters	*p*	*n*	*m*	*s*	*T*	Encrypting speed
Chaotic Logistic encryption	24	----	----	----	3hours	Fast
	32	----	----	----	6months	Slow
	40	----	----	----	60years	Very slow
Using Elman neural network in ChaoticLogistic encryption	8	4	4	4	3hours	Very Fast
	8	8	4	4	6months	Fast
	8	8	4	8	10^{15}years	Fast

References

1. Jakimoski, G., Kocarev, L.: Chaos, Cryptography: Block Encryption Ciphers Based on Chaotic Maps. Circuits and Systems – I: Fundamental Theory and Applications, **2** (2001) 163–169
2. Zhao, J., Luo, S., Wen, J.: A Chaotic Encryption Basic on Neural Network. Computer Research and Development, **12** (2001) 1475–1479
3. John, M.C., Verhagonj, P.: Chaos In Cryptography. The Escape From The Strange Attractor (1991)
4. Liew, P., Yee, D., Silva, L.C.: Application of Multilayer Perceptron Networks in Symmetric Block Ciphers. Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on, **2** (2002) 1455–1458
5. Su, S., Lin, A., Yen, J.: Design and Realization of a New Chaotic Neural Encryption/Decryption Network. Circuits and Systems, 2000. IEEE APCCAS 2000. The 2000 IEEE Asia-Pacific Conference, (2000) 335–338

Chosen-Plaintext Cryptanalysis of a Clipped-Neural-Network-Based Chaotic Cipher

Chengqing Li[1], Shujun Li[2,⋆], Dan Zhang[3], and Guanrong Chen[2]

[1] Department of Mathematics, Zhejiang University, Hangzhou 310027, China
swiftsheep@hotmail.com

[2] Department of Electronic Engineering, City University of Hong Kong
Kowloon, Hong Kong, China
hooklee@mail.com, eegchen@cityu.edu.hk
http://www.hooklee.com

[3] College of Computer Science, Zhejiang University, Hangzhou 310027, China
zhangdan@etang.com

Abstract. At ISNN'04, a novel symmetric cipher was proposed, by combining a chaotic signal and a clipped neural network (CNN) for encryption. The present paper analyzes the security of this chaotic cipher against chosen-plaintext attacks, and points out that this cipher can be broken by a chosen-plaintext attack. Experimental analyses are given to support the feasibility of the proposed attack.

1 Introduction

Since the 1990s, the study of using chaotic systems to design new ciphers has become intensive [1]. In particular, the idea of combining chaos and neural networks has been developed [2], [3], [4], [5] and has been adopted for image and video encryption [6], [7]. In our recent work [8], it has been shown that the chaotic ciphers designed in [2], [3], [4], [6], [7] are not sufficiently secure from a cryptographical point of view.

This paper focuses on the security of a clipped-neural-network-based chaotic cipher proposed in ISNN'04 [5]. This chaotic cipher employs a chaotic pseudorandom signal and the output of a 8-cell clipped neural network to mask the plaintext, along with modulus additions and XOR operations. Also, the evolution of the neural network is controlled by the chaotic signal. With such a complicated combination, it was hoped that the chaotic cipher can resist chosen-plaintext attacks. Unfortunately, our analysis shows that it is still not secure against chosen-plaintext attacks. By choosing only two plaintexts, an attacker can derive an equivalent key to break the cipher. This paper reports our analyses and simulation results.

The rest of the paper is organized as follows. Section 2 is a brief introduction to the chaotic cipher under study. The proposed chosen-plaintext attack is described in detail in Sec. 3, with some experimental results. The last section concludes the paper.

[⋆] Corresponding author

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 630–636, 2005.

2 The CNN-Based Chaotic Cipher

First, the CNN employed in the chaotic cipher is introduced. The neural network contains 8 neural cells, denoted by $S_0, \cdots, S_7 \in \{1, -1\}$, and each cell is connected with other cells via eight synaptic weights $w_{ij} \in \{1, 0, -1\}$, among which only three are non-zeros. The synaptic weights between two connected cells are identical: $\forall\, i, j = 0 \sim 7$, $w_{ij} = w_{ji}$. The neural network evolves according to the following rule: $\forall\, i = 0 \sim 7$,

$$f(S_i) = \text{sign}\left(\widetilde{S}_i\right) = \begin{cases} 1, & \widetilde{S}_i > 0 \ , \\ -1, & \widetilde{S}_i < 0 \ , \end{cases} \tag{1}$$

where $\widetilde{S}_i = \sum_{j=0}^{7} w_{ij} S_j$. Note that $\widetilde{S}_i \neq 0$ holds at all times.

Now, let us see how the chaotic cipher works with the above CNN. Without loss of generality, assume that $f = \{f(i)\}_{i=0}^{N-1}$ is the plaintext signal, where $f(i)$ denotes the i-th plain-byte and N is the plaintext size in byte. Accordingly, denote the ciphertext by $f' = \{f'(i)\}_{i=0}^{N-1}$, where $f'(i)$ is a double-precision floating-point number corresponding to the plain-byte $f(i)$. The encryption procedure can be briefly depicted as follows[1].

- *The secret key* includes the initial states of the 8 neural cells in the CNN, $S_0(0), \cdots, S_7(0)$, the initial condition $x(0)$, and the control parameter r of the following chaotic tent map:

$$T(x) = \begin{cases} rx, & 0 < x \leq 0.5 \ , \\ r(1-x), & 0.5 < x < 1 \ , \end{cases} \tag{2}$$

  where r should be very close to 2 to ensure the chaoticity of the tent map.
- *The initial procedure*: 1) in double-precision floating-point arithmetic, run the tent map from $x(0)$ for 128 times before the encryption starts; 2) run the CNN for $128/8 = 16$ times (under the control of the tent map, as discussed below in the last step of the encryption procedure); 3) set $x(0)$ and $S_0(0), \cdots, S_7(0)$ to be the new states of the tent map and the CNN.
- *The encryption procedure*: for the i-th plain-byte $f(i)$, perform the following steps to get the ciphertext $f'(i)$:
  - evolve the CNN for one step to get its new states: $S_0(i), \cdots, S_7(i)$;
  - in double-precision floating-point arithmetic, run the chaotic tent map for 8 times to get 8 chaotic states: $x(8i+0), \cdots, x(8i+7)$;
  - generate 8 bits by extracting the 4-th bits of the 8 chaotic states: $b(8i+0), \cdots, b(8i+7)$, and then $\forall\, j = 0 \sim 7$, set $E_j = 2 \cdot b(8i+j) - 1$;
  - encrypt $f(i)$ as follows[2]:

$$f'(i) = \left(\left(\frac{f(i) \oplus B(i)}{256} + x(8i+7) \right) \bmod 1 \right) \ , \tag{3}$$

  where $B(i) = (b(8i+0), \cdots, b(8i+7))_2 = \sum_{j=0}^{7} b(8i+j) \cdot 2^{7-j}$;

[1] Note that some original notations used in [5] have been changed in order to provide a better description.

[2] In [5], $x(8i+7)$ was mistaken as $x(8)$.

- $\forall\, i = 0 \sim 7$, if $S_i \neq E_i$, update all the three non-zero weights of the i-th neural cell and the three mirror weights as follows: $w_{ij} = -w_{ij}$, $w_{ji} = -w_{ji}$.

– *The decryption procedure* is similar to the above one with the following decryption formula:

$$f(i) = (256 \cdot ((f'(i) - x(8i+7)) \bmod 1)) \oplus B(i) \ . \tag{4}$$

3 The Chosen-Plaintext Attack

In chosen-plaintext attacks, it is assumed that the attacker can intentionally choose a number of plaintexts to try to break the secret key or its equivalent [9]. Although it was claimed that the chaotic cipher under study can resist this kind of attacks [5, Sec. 4], our cryptanalysis shows that such a claim is not true. By choosing two plaintexts, f_1 and f_2, satisfying $\forall\, i = 0 \sim N-1$, $f_1(i) = \overline{f_2(i)}$, one can derive two masking sequences as equivalent keys for decryption.

Before introducing the chosen-plaintext attack, three lemmas are given, which are useful in the following discussions.

Lemma 1. *$\forall\, a, b, c \in \mathbb{R}, c \neq 0$ and $n \in \mathbb{Z}^+$, if $a = (b \mod c)$, one has $a \cdot n = ((b \cdot n) \bmod (c \cdot n))$.*

Proof. From $a = (b \mod c)$, one knows that $\exists\, k \in \mathbb{Z}$, $b = c \cdot k + a$ and $0 \leq a < c$. Thus, $\forall\, n \in \mathbb{Z}^+$, $b \cdot n = c \cdot n \cdot k + a \cdot n$ and $0 \leq a \cdot n < c \cdot n$, which immediately leads to $a \cdot n = ((b \cdot n) \bmod (c \cdot n))$ and completes the proof of this lemma. □

Lemma 2. *$\forall\, a, b, c, n \in \mathbb{R}$ and $0 \leq a, b < n$, if $c = ((a - b) \mod n)$, one has $a - b \in \{c, c - n\}$.*

Proof. This lemma can be proved under two conditions. i) When $a \geq b$, it is obvious that $((a-b) \bmod n) = a - b = c$. ii) When $a < b$, $((a-b) \bmod n) = ((n + a - b) \bmod n)$. Since $-n < a - b < 0$, one has $0 < n + a - b < n$, which means that $((a-b) \bmod n) = n + a - b = c$. That is, $a - b = c - n$. Combining the two conditions, this lemma is thus proved. □

Lemma 3. *Assume that a, b are both 8-bit integers. If $a = b \oplus 128$, then $a \equiv (b + 128) \pmod{256}$.*

Proof. This lemma can be proved under two conditions. i) When $0 \leq a < 128$: $b = a \oplus 128 = a + 128$, so $a \equiv (b + 128) \pmod{256}$. ii) When $128 \leq a \leq 255$: $b = a \oplus 128 = a - 128$, so $a \equiv (b - 128) \equiv (b + 128) \pmod{256}$. □

From Lemma 1, one can rewrite the encryption formula Eq. (3) as follows:

$$256 \cdot f'(i) = (((f(i) \oplus B(i)) + 256 \cdot x(8i+7)) \bmod 256) \ . \tag{5}$$

Given two plain-bytes $f_1(i) \neq f_2(i)$ and the corresponding cipher-blocks $f_1'(i)$, $f_2'(i)$, one has $256 \cdot (f_1'(i) - f_2'(i)) \equiv ((f_1(i) \oplus B(i)) - (f_2(i) \oplus B(i))) \pmod{256}$.

Without loss of generality, assume that $f_1'(i) > f_2'(i)$ and that $\varDelta_{f_{1,2}} = 256 \cdot (f_1'(i) - f_2'(i))$. It is true that $0 < \varDelta_{f_{1,2}} < 256$. Thus, one has

$$\varDelta_{f_{1,2}} = (((f_1(i) \oplus B(i)) - (f_2(i) \oplus B(i))) \bmod 256) \ . \tag{6}$$

Because $f_1(i) \oplus B(i)$ and $f_2(i) \oplus B(i)$ are 8-bit integers and $\varDelta_{f_{1,2}} \neq 0$, from Lemma 2, one of the following facts is true:

1. $(f_1(i) \oplus B(i)) - (f_2(i) \oplus B(i)) = \varDelta_{f_{1,2}} \in \{1, \cdots, 255\}$; (7a)
2. $(f_2(i) \oplus B(i)) - (f_1(i) \oplus B(i)) = \left(256 - \varDelta_{f_{1,2}}\right) \in \{1, \cdots, 255\}$. (7b)

For the above two equations, when $f_1(i) = \overline{f_2(i)}$ is satisfied, two possible values of $B(i)$ can be uniquely derived according to the following theorem.

Theorem 1. *Assume that a, b, c, x are all 8-bit integers, and $c > 0$. If $a = \bar{b}$, then the equation $(a \oplus x) - (b \oplus x) = c$ has an unique solution $x = a \oplus (1, c_7, \cdots, c_1)_2$, where $c = (c_7, \cdots, c_0)_2 = \sum_{i=0}^{7} c_i \cdot 2^i$.*

Proof. Since $a = \bar{b}$, one has $b \oplus x = \overline{a \oplus x}$. Thus, by substituting $y = a \oplus x$ and $\bar{y} = \overline{a \oplus x} = b \oplus x$ into $(a \oplus x) - (b \oplus x) = c$, one can get $y - \bar{y} = c$, which is equivalent to $y = \bar{y} + c$. Let $y = \sum_{i=0}^{7} y_i \cdot 2^i$, and consider the following three conditions, respectively.

1) When $i = 0$, from $y_0 \equiv (\bar{y}_0 + c_0) \pmod 2$, one can immediately get $c_0 = 1$. Note the following two facts: i) when $y_0 = 0$, $\bar{y}_0 + c_0 = 2$, a carry bit is generated for the next bit, so $y_1 \equiv (\bar{y}_1 + c_1 + 1) \pmod 2$ and $c_1 = 0$; ii) when $y_0 = 1$, $\overline{y_0} + c_0 = 1$, no carry bit is generated, so $y_1 \equiv (\bar{y}_1 + c_1) \pmod 2$ and $c_1 = 1$. Apparently, it is always true that $y_0 = c_1$. Also, a carry bit is generated if $c_1 = 0$ is observed.

2) When $i = 1$, if there exists a carry bit, set $c_1' = c_1 + 1 \in \{1, 2\}$; otherwise, set $c_1' = c_1 \in \{0, 1\}$. From $y_1 \equiv (\bar{y}_1 + c_1') \pmod 2$, one can immediately get $c_1' = 1$. Then, using the same method shown in the first condition, one has $y_1 = c_2$ and knows whether or not a carry bit is generated for $i = 2$. Repeat the above procedure for $i = 2 \sim 6$, one can uniquely determine that $y_i = c_{i+1}$.

3) When $i = 7$, it is always true that the carry bit does not occur, so $c_7' = 1$, and $y_7 \equiv 1$.

Combining the above three conditions, one can get $y = (1, c_7, \cdots, c_1)_2$, which results in $x = a \oplus (1, c_7, \cdots, c_1)_2$. □

Assume that the two values of $B(i)$ derived from Eqs. (7a) and (7b) are $B_1(i)$ and $B_2(i)$, respectively. The following corollary shows that the two values have a deterministic relation: $B_2(i) = B_1(i) \oplus 128$.

Corollary 1. *Assume that a, b, c, x are all 8-bit integers, $a = \bar{b}$ and $c > 0$. Given two equations, $(a \oplus x) - (b \oplus x) = c$ and $(b \oplus x') - (a \oplus x') = c'$, if $c' = 256 - c$, then $x' = x \oplus 128$.*

Proof. Since $c + \bar{c} = 255$, one has $c' = 256 - c = \bar{c} + 1$. Let $c = \sum_{i=0}^{7} c_i \cdot 2^i$, and observe the first condition of the proof of Theorem 1. One can see that $c_0 = 1$, so $c'_0 = \bar{c}_0 + 1 = 1$. Since there is no carry bit, one can deduce that $\forall\, i = 1 \sim 7$, $c'_i = \bar{c}_i$. Applying Theorem 1 for $(a \oplus x) - (b \oplus x) = c$, one can uniquely get $x = a \oplus (1, c_7, \cdots, c_1)_2$. Then, applying Theorem 1 for $(b \oplus x') - (a \oplus x') = c'$, one has $x' = b \oplus (1, c'_7, \cdots, c'_1)_2 = \bar{a} \oplus (1, \bar{c}_7, \cdots, \bar{c}_1)_2 = (a_7, \bar{a}_6 \oplus \bar{c}_7, \cdots, \bar{a}_0 \oplus \bar{c}_1)_2 = (a_7, a_6 \oplus c_7, \cdots, a_0 \oplus c_1)_2 = a \oplus (1, c_7, \cdots, c_1)_2 \oplus (1, 0, \cdots, 0)_2 = x \oplus 128$. Thus, this corollary is proved. □

For any one of the two candidate values of $B(i)$, one can further get an equivalent chaotic state $\hat{x}(8i+7)$ from $B(i)$, $f(i)$ and $f'(i)$ as follows:

$$\hat{x}(8i+7) = 256 \cdot f'(i) - (f(i) \oplus B(i)) \equiv 256 \cdot x(8i+7) \pmod{256} \ . \tag{8}$$

With $B(i)$ and $\hat{x}(8i+7)$, the encryption formula Eq. (3) becomes

$$f'(i) = \frac{((f(i) \oplus B(i)) + \hat{x}(8i+7)) \bmod 256}{256} \ , \tag{9}$$

and the decryption formula Eq. (4) becomes

$$f(i) = ((256 \cdot f'(i) - \hat{x}(8i+7)) \bmod 256) \oplus B(i) \ . \tag{10}$$

Assume that $\hat{x}_1(8i+7)$ and $\hat{x}_2(8i+7)$ are calculated by Eq. (8), from $B_1(i)$ and $B_2(i)$, respectively. Then, we have the following proposition.

Proposition 1. *$(B_1(i), \hat{x}_1(8i+7))$ and $(B_2(i), \hat{x}_2(8i+7))$ are equivalent for the above encryption procedure Eq. (9), though only one corresponds to the correct value generated from the secret key. That is,*

$$((f(i) \oplus B_1(i)) + \hat{x}_1(8i+7)) \equiv ((f(i) \oplus B_2(i)) + \hat{x}_2(8i+7)) \pmod{256} \ .$$

Proof. From $B_1(i) = B_2(i) \oplus 128$, one has $f(i) \oplus B_1(i) = (f(i) \oplus B_2(i) \oplus 128)$. Then, following Lemma 3, it is true that $(f(i) \oplus B_1(i)) \equiv ((f(i) \oplus B_2(i)) + 128) \pmod{256}$. As a result, $\hat{x}_1(8i+7) = (256 \cdot f'(i) - (f(i) \oplus B_1(i))) \equiv (256 \cdot f'(i) - ((f(i) \oplus B_2(i)) - 128)) \pmod{256} \equiv (\hat{x}_2(8i+7) + 128) \pmod{256}$, which immediately leads to the following fact: $((f(i) \oplus B_1(i)) + \hat{x}_1(8i+7)) \equiv ((f(i) \oplus B_2(i)) + \hat{x}_2(8i+7)) \pmod{256}$. Thus, this proposition is proved. □

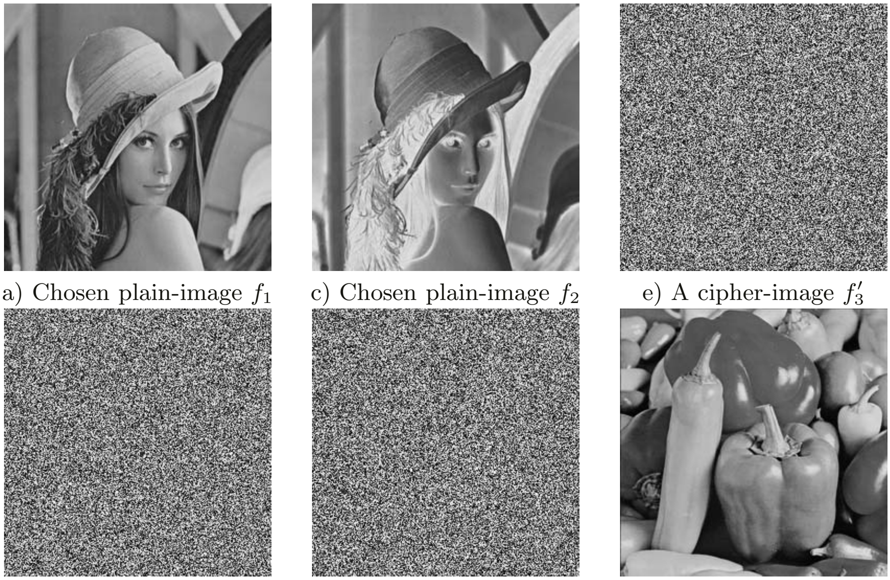
Considering the symmetry of the encryption and decryption procedures, the above proposition immediately leads to a conclusion that $(B_1(i), \hat{x}_1(8i+7))$ and $(B_2(i), \hat{x}_2(8i+7))$ are also equivalent for the decryption procedure Eq. (10).

From the above analyses, with two chosen plaintexts f_1 and $f_2 = \bar{f}_1$, one can get the following two sequences: $\{B_1(i), \hat{x}_1(8i+7)\}_{i=0}^{N-1}$ and $\{B_2(i), \hat{x}_2(8i+7)\}_{i=0}^{N-1}$. Given a ciphertext $f' = \{f'(i)\}_{i=0}^{N-1}$, $\forall\, i = 0 \sim N-1$, one can use either $(B_1(i), \hat{x}_1(8i+7))$ or $(B_2(i), \hat{x}_2(8i+7))$ as an equivalent of the secret key to decrypt the i-th plain-byte $f(i)$, following Eq. (10). This means that the chaotic cipher under study is not sufficiently secure against the chosen-plaintext attack.

To demonstrate the feasibility of the proposed attack, some experiments have been performed for image encryption, with secret key $r = 1.99$, $x(0) = 0.41$ and $[S_0(0), \cdots, S_7(0)] = [1, -1, 1, -1, 1, -1, 1, -1]$. One plain-image "Lenna" of

Table 1. The first ten elements of $\{B_1(i), \hat{x}_1(8i+7)\}_{i=0}^{256\times 256-1}$ and $\{B_2(i), \hat{x}_2(8i+7)\}_{i=0}^{256\times 256-1}$

i	0	1	2	3	4	5	6	7	8	9
$B_1(i)$	146	231	54	202	59	243	166	173	233	82
$B_2(i)$	18	103	182	74	187	115	38	45	105	210
$\hat{x}_1(8i+7)$	242.40	38.63	242.62	222.09	81.03	214.73	240.91	203.59	138.20	9.33
$\hat{x}_2(8i+7)$	114.40	166.63	114.62	94.09	209.03	86.73	112.91	75.59	10.20	137.33


a) Chosen plain-image f_1 c) Chosen plain-image f_2 e) A cipher-image f_3'

b) Cipher-image f_1' d) Cipher-image f_2' f) Recovered image f_3

Fig. 1. The proposed chosen-plaintext attack

size 256 × 256 is chosen as f_1 and another plain-image is manually generated as follows: $f_2 = \bar{f}_1$. The two plain-images and their cipher-images are shown in Fig. 1. With the two chosen plain-images, two sequences, $\{B_1(i), \hat{x}_1(8i+7)\}_{i=0}^{256\times 256-1}$ and $\{B_2(i), \hat{x}_2(8i+7)\}_{i=0}^{256\times 256-1}$, are generated by using the above-mentioned algorithm. The first ten elements of the two sequences are given in Table 1. $\forall\, i = 0 \sim (256\times 256-1)$, either $(B_1(i), \hat{x}_1(8i+7))$ or $(B_2(i), \hat{x}_2(8i+7))$ can be used to recover the plain-byte $f(i)$. As a result, the whole plain-image ("Peppers" in this test) can be recovered as shown in Fig. 1f.

4 Conclusion

In this paper, the security of a chaotic cipher based on clipped neural network has been analyzed in detail. It is found that the scheme can be effectively broken with only two chosen plain-images. Both theoretical and experimental analyses have been given to support the proposed attack. Therefore, this scheme is not suggested for applications that requires a high level of security.

Acknowledgements

This research was supported by the National Natural Science Foundation, China, under grant no. 60202002, and by the Applied R&D Centers of the City University of Hong Kong under grants no. 9410011 and no. 9620004.

References

1. Li, S.: Analyses and New Designs of Digital Chaotic Ciphers. PhD thesis, School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an, China (2003) available online at `http://www.hooklee.com/pub.html`.
2. Yen, J.C., Guo, J.I.: A chaotic neural network for signal encryption/decryption and its VLSI architecture. In: Proc. 10th VLSI Design/CAD Symposium. (1999) 319–322
3. Su, S., Lin, A., Yen, J.C.: Design and realization of a new chaotic neural encryption/decryption network. In: Proc. IEEE Asia-Pacific Conference on Circuits and Systems. (2000) 335–338
4. Yen, J.C., Guo, J.I.: The design and realization of a chaotic neural signal security system. Pattern Recognition and Image Analysis (Advances in Mathematical Theory and Applications) **12** (2002) 70–79
5. Zhou, T., Liao, X., Chen, Y.: A novel symmetric cryptography based on chaotic signal generator and a clipped neural network. In: Advances in Neural Networks - ISNN 2004: International Symposium on Neural Networks Proceedings, Part II. Volume 3174 of Lecture Notes in Computer Science. (2004) 639–644
6. Lian, S., Chen, G., Cheung, A., Wang, Z.: A chaotic-neural-network-based encryption algorithm for JPEG2000 encoded images. In: Advances in Neural Networks - ISNN 2004 Proceedings, Part II. Volume 3174 of Lecture Notes in Computer Science. (2004) 627–632
7. Lian, S., Sun, J., Li, Z., Wang, Z.: A fast MPEG4 video encryption scheme based on chaotic neural network. In: Neural Information Processing: ICONIP 2004 Proceedings. Volume 3316 of Lecture Notes in Computer Science. (2004) 720–725
8. Li, C., Li, S., Zhang, D., Chen, G.: Cryptanalysis of a chaotic neural network based multimedia encryption scheme. In: Advances in Multimedia Information Processing - PCM 2004 Proceedings, Part III. Volume 3333 of Lecture Notes in Computer Science., Springer-Verlag (2004) 418–425
9. Schneier, B.: Applied Cryptography – Protocols, Algorithms, and Souce Code in C. Second edn. John Wiley & Sons, Inc., New York (1996)

A Novel Identity-Based Key Issuing Scheme Based on Interacting Neural Network*

Tieming Chen, Bo Chen, and Jiamei Cai

College of Software, Zhejiang University of Technology
Hangzhou, Zhejiang 310014, China
{tmchen,cb,cjm}@zjut.edu.cn

Abstract. Identity-based public key cryptosystem may perfectly substitute the traditional certificate-based public key system if only the efficiency and security of key issuing are satisfied. Recently, interacting neural network has been studied with a novel result that the two neural networks can synchronize to a stationary weight space with the identical inputs. So we propose a tree parity machine model for secure key agreement purpose, and ID-based private key secure issuing over a complete public channel, as well as secure authentication to user, are all proposed in this paper.

1 Introduction

Traditional certificate-based public key infrastructure (PKI) has been widely applied for system security, but there exists grievous management cost expanding problems for public key certificates, revocation lists, and cross-certification, etc.. In addition, the requirement of PKI for pre-enrollment of all users limits its widespread application. On the other hand, Identity-based(ID-base) cryptosystem[1],[2] eliminates the need for certificates and overcomes above PKI obstacles by mapping a public key to a publicly known identifier of the receiver, such as email addresses, while the responsible private key is generated by a so-called key generator center. Thus a sender can send a secure message(encrypted with some receiver's public identity) to a receiver, and the receiver can decrypt the encrypted message using the private key. In fact, a sender can send an encrypted message even before the receiver obtains its private key from the KGC, and when to read the encrypted messages, the receiver then obtains its private key from the KGC by authenticating himself. So, it's most important for ID-based system that how KGC authenticates the users who submit key requests, and how to issue the keys to receivers securely.

At present, there exists several key ID-based key issuing protocols, most of them aim to process the key escrow problem. According to paper[3], the master key of the KGC is distributed into multiple authorities, and the private key of a user is computed in a threshold way, thus key escrow problem of a single authority can be prevented. In addition, paper[4] presents that the private key of a user can be generated by compounding multiple independent subkeys from multiple authorities, and the authorities

* Partially supported by Zhejiang Province Natural Science Fund (Y104158) and Technology Development Fund of Zhejiang University of Technology.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 637–642, 2005.

work simultaneously. However, both the approaches have the drawback that different authorities have to check and authenticate user's identity independently, which is quite a bottleneck to the system. Lately, a new scheme[5] has been proposed that the user's private key may be issued by the KGC while its privacy protected by multiple key privacy authorities, and the authorities work sequentially. Due to that only one authority designated to authenticate the user, the identification cost should be largely reduced. But the scheme also makes use of user-chosen secret information for constructing secure channel for user to retrieve private key securely, so it consumes lots of computation. Although some certificate-based encryption scheme has also been presented[6], it will short the ID-based cryptography advantages because that the public key is no longer identified by the user's identity.

So, the research on a convenient and secure ID-based key issuing scheme with rather high performance is kept on keen. Fortunately, the state-of-the-art research of interacting neural network[7],[8] has shown that both weights of such two networks can be synchronized by the same training under some Hebbian rules with the identical inputs. Expanding interacting neural network to multilayer network, such as tree parity machine, a secure key agreement model can be constructed based on the system characteristic of final synchronization, by which we should build a secure and authenticated public key issuing scheme over an open communication channel.

2 Interacting Neural Network Cryptography

2.1 Interacting Neural Network

It has been proven that neural network can learn from examples similar to human brain. So many useful models have been modeled as non-classical mathematical methods for some nonlinear problems. However, what happens if two neural networks learn from each other? Recently, an analytical solution has been presented to show a novel phenomenon: synchronization by mutual learning. The biological consequences of this phenomenon are not explored, yet, but it's really an important stone for cryptography application: secure generation of a secret key over a public channel. In next, we in brief introduce the synchronization mechanism based on simple mutual learning network[7].

Fig.1 illustrates a simple model system that wo perceptrons receive a common random input vector $\underline{x}$ and modify their weights $\underline{w}$ according to their mutual bit σ. The output bit σ of a single perceptron is given by the equation:

$$\sigma = \mathrm{sign}(\underline{w} \cdot \underline{x})$$

$\underline{x}$ is an N-dimensional input vector with components which are drawn from a Gaussian with mean 0 and variant 1. $\underline{w}$ is a N-dimensional weight vector with continuous components which are normalized:

$$\underline{w} \cdot \underline{w} = 1$$

The initialization is to choose a random weight vectors $w_{Ai}(w_{Bi})$, i = 1, ...N. At each training step a common random input vector is presented to the two networks which generate two output bits σ_A and σ_B. From paper[7], the perceptron learning rule for weight updating listed as below:

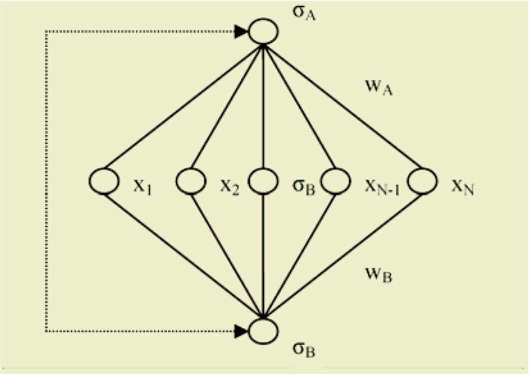


Fig. 1. Two perceptrons receive an identical input $\{x_i\}$ and learn mutual output bits σ_A, σ_B with their weights w_A,w_B

$$\underline{w}_A(t+1) = \underline{w}_A(t) + \frac{\eta}{N}\underline{x}\sigma_B\Theta(-\sigma_A\sigma_B)$$

$$\underline{w}_B(t+1) = \underline{w}_B(t) + \frac{\eta}{N}\underline{x}\sigma_A\Theta(-\sigma_A\sigma_B)$$

where Θ (x) is the step function, η is the learning rate.

Due to analytics in [7], we know that above a critical rate □c the networks relax to a weight antiparallel state, $w_A = -w_B$. This is the key stuff of interacting neural network synchronization.

2.2 Key Agreement Using Tree Parity Machine (TPM)

Some modifications and improvements have been done in paper[8] for utilizing tree parity machine to secret key generator. At first, discrete weight vectors are constructed that $w_{A(B)i} \in \{-L,-L+1, ...,L-1,L\}$,which makes each component of the weight vectors perform a kind of random walk with reflecting boundary L. In addition, the learning rule is also modified in order to obtain parallel synchronization with two weight vectors:

$$\underline{w}_A(t+1) = \underline{w}_A(t) - \underline{x}\sigma_A\Theta(-\sigma_A\sigma_B)\ ,$$

$$\underline{w}_B(t+1) = \underline{w}_B(t) - \underline{x}\sigma_B\Theta(-\sigma_A\sigma_B)\ .$$

From the Fig.2 of TPM, two equations below are met:

$$\tau_A = \sigma_A^1\sigma_A^2\sigma_A^3\ ,$$

$$\tau_B = \sigma_B^1\sigma_B^2\sigma_B^3\ .$$

The two machines A and B receive identical input vectors x_1, x_2, x_3 at each training step. The training algorithm is that: Only if the two output bits are identical, $\tau_A = \tau_B$, the weights do be modified otherwise go to next round. During the weight modification, only the hidden unit σ_i which is identical to the output bits do modify its weights using the following Hebbian rule:

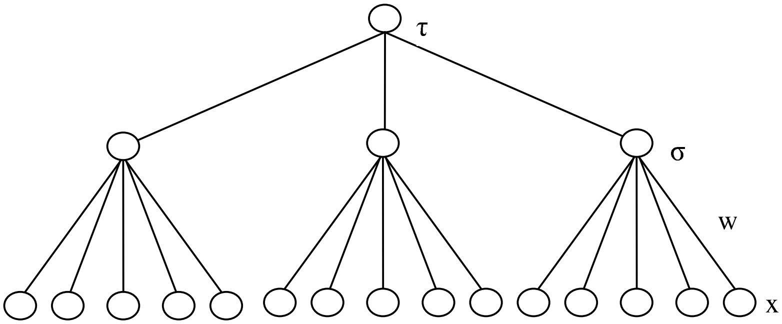


Fig. 2. Tree parity machine with three hidden units

$$\underline{w}_A^i(t+1) = \underline{w}_A^i(t) - \underline{x}_i \tau_A$$

Above brief discussions referenced from [8] which finally gives the conclusion that about 400 training steps later, two partners A and B with their respective initial weights and identical input vectors synchronize to a stationary state, $\underline{w}_B^i(t) = \underline{w}_B^i(t)$. But non-identical inputs lead to non-synchronization.

So it implies that this synchronization phenomenon can be used for secure key agreement. The two partners can agree on a common secret key over a public channel by finally mapping the synchronized weight space into such secret.

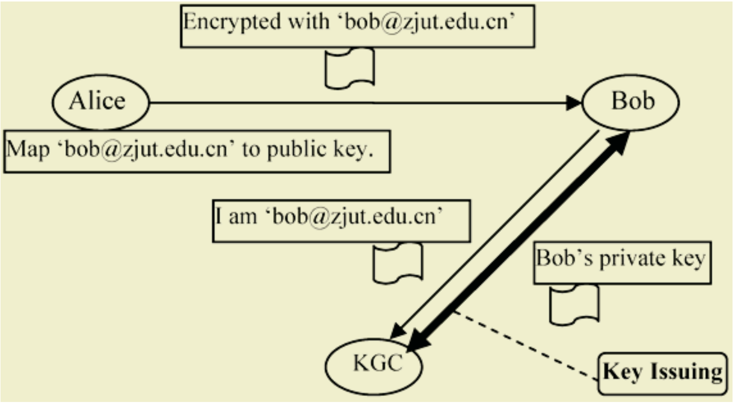

Encrypted with 'bob@zjut.edu.cn'
Alice
Bob
Map 'bob@zjut.edu.cn' to public key.
I am 'bob@zjut.edu.cn'
Bob's private key
KGC
Key Issuing

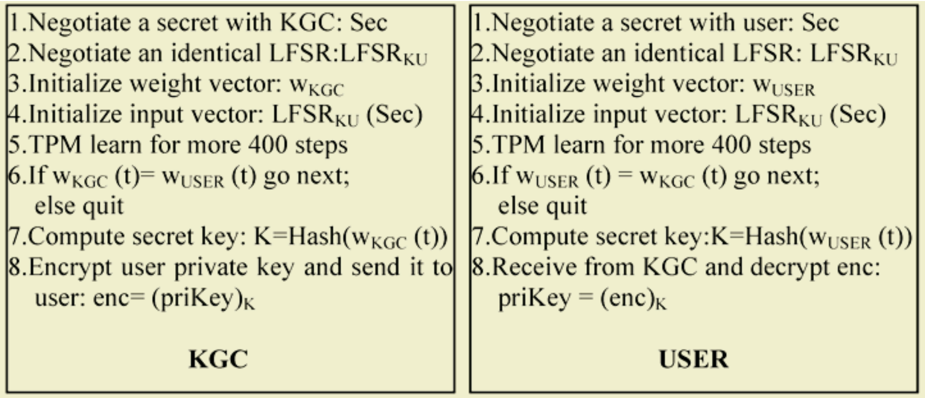
Fig. 3. A typical ID-based application framework

3 Proposed Key Issuing Scheme

3.1 ID-Based Key Issuing

A typical running of ID-based cryptsystems is shown in Fig.3. It's clear that the key issuing is the focused problem on which whether ID-based public key system is more advanced than PKI depends. As for designing a secure communication channel, we consider that the proposed scheme requires several conditions to met:(1) user can be authenticated by KGC;(2) two partners can agree with a unique key, namely key exchange; (3) any attacker can not compute the exchanged key between two partners. Finally, KGC encrypts the private key using the secret key, sends to user for secure receiver.

Based on section 2, we just describe the steps of our key issuing scheme as below.



1.Negotiate a secret with KGC: Sec
2.Negotiate an identical LFSR:LFSR_{KU}
3.Initialize weight vector: w_{KGC}
4.Initialize input vector: LFSR_{KU} (Sec)
5.TPM learn for more 400 steps
6.If w_{KGC} (t)= w_{USER} (t) go next;
else quit
7.Compute secret key: K=Hash(w_{KGC} (t))
8.Encrypt user private key and send it to user: enc= $(\text{priKey})_{\text{K}}$

KGC

1.Negotiate a secret with user: Sec
2.Negotiate an identical LFSR: LFSR_{KU}
3.Initialize weight vector: w_{USER}
4.Initialize input vector: LFSR_{KU} (Sec)
5.TPM learn for more 400 steps
6.If w_{USER} (t) = w_{KGC} (t) go next;
else quit
7.Compute secret key:K=Hash(w_{USER} (t))
8.Receive from KGC and decrypt enc:
priKey = $(\text{enc})_{\text{K}}$

USER

Note that Sec is securely negotiated between KGC and USER(may select on-line or off-line method) therefore is only known to KGC and USER while it's pre-stored secret information in KGC and USER as well as the weight vectors. LFSR_{KU} is a selected publicly known LFSR as pseudo-random number generator for both identical inputs production. If the user presents the correct secret Sec, in other words two partners have the same inputs, their weights must synchronize, thus a final exchanged secret key K can be either obtained by simple hash function on the identical weight vectors. Now user private key issuing over complete public channel based on TPM is done. The structure also sketched as Fig.4.

3.2 Analysis

Some concrete analytical results stay in [7],[8], from which we know that TPM model with hidden layer is rather secure than simple interacting neural network, since even in case of an opponent breaking into network and obtaining the change of output bits it still can not identify which one of weight vectors is changed. On the other hand, the time for opponents getting synchronization is longer than the partners do, so the agreed secret key is practical secure for private key encryption. In addition, it is true that if the inputs are not identical, the system can never synchronize. so if the USER is a personator without the exact secret Sec, it's initial input vector then also should not be identical to KGC's, therefore this 'USER' should not be authenticated by KGC for synchronization not obtaining.

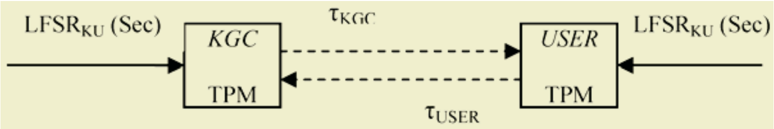


Fig. 4. ID-based key issuing scheme

4 Conclusions

Recently, so many researchers work on ID-based cryptsystems but seldom practical results outcome. The main problem lies in the private key issuing management. As we know that nobody has present any ID-based cryptography related to neural network. Here we propose a key issuing scheme based on generating secure key by TPM. Although the security of using such interacting neural network synchronization should be further analyzed, in contrast to traditional number theoretical methods this neural network model are analytically very fast and more attractive for development in future.

References

1. Shamir, A.: Identity-base Cryptosystems and Signature Schemes. Proc. of Crypto'84, Lecture Notes in Computer Science, Springer-Verlag, Berlin, **196** (1984) 47-53
2. Boneh, D., Franklin, F.: Identity-based Encryption from the Weil Pairing. Advances in Cryptology- Crypto'2001, LNCS 2139. Springer-Verlag, Berlin (2001) 213-229
3. Chen, L., Harrison, K., Smart, N.P., Soldera, D.: Applications of Multiple Trust Authorities in Pairing Based Cryptosystems. InfraSec 2002, LNCS 2437.Springer-Verlag, Berlin (2002) 260-275
4. Chow, Sherman, S.M., Hui, Lucas, C.K., Yiu, S.M.: Identity-based Threshold Ring Signature. In 7th International Conference on Information Security and Cryptology (ICISC 2004), Seoul, Korea (2004)
5. Lee, B., Boyd, C., Dawson, E., Kim, K., Yang, J., Yoo, S.: Secure Key Issuing in ID-Based Cryptography. ACM Second Australasian Information Security Workshop (AISW 2004), New Zealand (2004) 69-74
6. Al-Riyami,S., Paterson, K.: Certificateless Public Key Cryptography. Advances in Cryptology - Asiacrypt'2003, LNCS 2894. Springer-Verlag, Berlin (2003) 452-473
7. Metzler, R., Kinzel, W.: Interacting Neural Networks. Phys. Rev. E., **62** (2000) 2555-2562
8. Kinzel, W., Kanter, I.: Interacting Neural Network and Cryptography. Advances in Solid State Physics, **42** (2002) 383-391

The Projection Pursuit Learning Network for Nonlinear Time Series Modeling and Forecasting

Zheng Tian[1,2], Zi Jin[3], Fang He[1], and Wei Ling[1]

[1] Department of Applied Mathematics, Northwestern Polytechnical University
Xi'an, Shaanxi 710072, China
[2] National Key Laboratory for Pattern Recognition
Institute of Automation Chinese Academy of Sciences, Beijing 100080, China
[3] Department of Statistics, Toronto University, Toronto, Canada
zhtian@nwpu.edu.cn

Abstract. Nonlinear time series modeling and forecasting is one of important problems of nonlinear time series analysis. In this paper, we prove that projection pursuit learning network can approximate to nonlinear autoregression at any given precision in L^k space, where k is some positive integer. The mathematical formulation, training strategy and calculative procedures are also presented. The results of application to real data show that the projection pursuit learning network outperforms the traditional methods.

1 Introduction

Projection pursuit regression (PPR) has been proved to be an efficient way of coping with the problems of the 'curse of dimensionality' in nonparametric regression and has been intensively investigated. The projection pursuit learning network (PPLN) is based on PPR and possesses the structure of a two-layer feed forward network structure. PPLN was first introduced by Barron and Barron [1]. Jones [2] established the nonsampling convergence rate for projection pursuit regression and neural network under mild assumptions. Hwang et al ([3]) studied several two-dimensional examples to compare PPR and one layer sigmoidal neural network.

On the other hand, an important goal of nonlinear time series is to nonlinear time series predication and modeling. However, is there any simple and more straightforward method than the traditional method [4],[5]….[10] for nonlinear time series predication and modeling? In this paper, we prove that PPLN can approximate to nonlinear autoregression at any given precision in L^k space, where k is some positive integer. The mathematical formulation, training strategy and calculative procedures are also presented. The results of the application to real data show that not only the predication error with PPLN is much smaller than the ones with traditional methods but also the computation with PPLN is simpler and faster than the others as well.

In section 2, we introduce mathematical formulation, training strategy and calculative procedures for nonlinear time series predication and modeling by PPLN. Section

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 643–650, 2005.

3 proves that PPLN can approximate to nonlinear autoregression at any given precision in L^k space, where k is some integer. Section 4 contains some numerical results examining the properties of PPLN. The results of the application to real data show that the PPLN outperforms the traditional methods.

2 The Learning Strategies and Calculative Procedures of PPLN for Nonlinear Time Series

2.1 Projection Pursuit Learning Network

A projection pursuit learning network is the network that combines the advantages of both neural network and improving PPR. Let $x_1,\cdots,x_p$ be the input of PPLN, $y_1,\cdots,y_q$ be the desired output and $\hat{y}_1,\cdots,\hat{y}_q$ be the actual output, then the input vector $X=(x_1,\cdots,x_p)^{\mathrm{T}}$ and the actual output $\hat{y}_i(1\le i\le q)$ satisfy the mathematics model is as follows:

$$\hat{y}_i=\sum_{k=1}^{m}\beta_{ik}g_k(\sum_{j=1}^{p}\alpha_{kj}x_j)=\sum_{k=1}^{m}\beta_{ik}g_k\left(\alpha_k^T X\right),i=1,\cdots,q. \tag{1}$$

where $\alpha_k^{\mathrm{T}}=(\alpha_{k1},\alpha_{k2},\cdots,\alpha_{kp})^{\mathrm{T}}(k=1,\cdots,m)$ denotes the weight vector liked between the input layer and hidden layer, α_{kj} denotes the input layer weight linked between the k^{th} activation function $g_k(\cdot)$ of the hidden layer and the j^{th} neuron of the input layer; β_{ik} denotes the output layer weight linked between the $i-th$ output neuron $\hat{y}_i$ and the k^{th} hidden active function $g_k(\cdot)$ $k=1,\cdots,m,$ $i=1,\cdots q$. $\{g_k\}$ are unknown smooth function. Three sets of parameters, projection directions $\alpha_k^{\mathrm{T}}=(\alpha_{k1},\alpha_{k2},\cdots,\alpha_{kp})$, projection "strengths" β_{ik} and the unknown smooth activation functions $\{g_k\}$ are estimated via the minimizing the criteria function

$$L_2=\sum_{i=1}^{q}w_iE(y_i-\hat{y}_i)^2=\sum_{i=1}^{q}w_iE\left(y_i-\sum_{k=1}^{m}\beta_{ik}g_k\left(\alpha_k^T X\right)\right)^2,$$

where $w_i(1\le i\le q)$ denotes the learning rate what is decided by user and the expectation operator E denote the sample average over all the n training data $(y_l,x_l),\ \ l=1,\cdots,n.$ The topological structure is shown in Figure1.

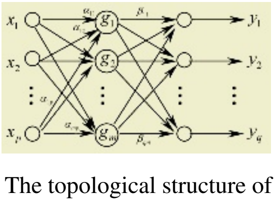


Fig. 1. The topological structure of PPLN

2.2 The Learning Strategies of the PPLN for Univariate Nonlinear Time Series Modeling and Forecasting

The learning strategies of PPLN use the alternative optimization method to decide the three kinds of parameters, including the input layer weights $\alpha_{kj},\ 1\le k\le m,\ 1\le j\le p,$ the activation functions of hidden layer $g_k(\cdot),\ 1\le k\le m$ and the output layer weights $\beta_{ik}, 1\le k\le m, 1\le i\le g$. All the parameters to be estimated are divided into m groups (each associated with one hidden neuron), and each group, say the k^{th} group, is further divided into three subgroups: the output-layer weights $\{\beta_{ik}, i=1,\cdots,q\}$ connected to the k^{th} hidden neuron; and the input-layer weights $\{\alpha_{kj}, j=1,\cdots,p\}$ connected to the k^{th} hidden neuron. The PPLN procedure starts from updating the parameters associated with the first hidden (group) by updating each subgroup, $\{\beta_{i1}\}$, f_1, and $\{\alpha_{1j}\}$ consecutively (layer-by-layer) to minimize the loss function L_2. And then it updates the parameters associated with the second hidden neuron by consecutively updating $\{\beta_{i2}\}$, f_2 and $\{\alpha_{2j}\}$. A complete updating pass ends at the updating of the parameters associated with the m^{th} (the last) hidden neuron by consecutively updating $\{\beta_{im}\}$, f_m , and $\{\alpha_{mj}\}$. Repeated updating passes are made over all the groups until the precision satisfies the demands. The L_2 can be written as follows:

$$L_2 = L_2^{(k)} = \sum_{i=1}^{q} w_i E[R_{i(k)} - \beta_{ik} g_k(\alpha_k^T X)]^2 . \tag{2}$$

where

$$R_{i(k)} = y_i - \sum_{l\neq k} \beta_{il} g_l(\alpha_l^T X) \tag{3}$$

doesn't include the action of the k^{th} group parameters to the L_2 .

The estimations β_{ik} ***of output layer weights*** β_{ik} . Given $R_{i(k)}$ and fixed g_k and α_k , the estimations $\hat{\beta}_{ik}$ of the parameters β_{ik} , computed by setting the derivatives of L_2 with respect to β_{ik} equal to zero, are:

$$\hat{\beta}_{ik} = \frac{E[R_{i(k)} g_k(\alpha_k^T X)]}{E[g_k(\alpha_k^T X)]^2}, (1\le i\le g) . \tag{4}$$

The estimations $\hat{g}_k(\cdot)$ ***of activation function*** $g_k(\cdot)$. Rewrite $L_2^{(k)}$:

$$L_2^{(k)} = E\{E[\sum_{i=1}^{q} w_i (R_{i(k)} - \beta_{ik} g_k(\alpha_k^T X))^2 \big| \alpha_k^T X]\} . \tag{5}$$

then

$$\hat{g}_k(\alpha_k^T X) = \frac{E[\sum_{i=1}^{q} w_i \beta_{ik} R_{i(k)} \big| \alpha_k^T X]}{\sum_{i=1}^{q} w_i \beta_{ik}^2} = \frac{\sum_{i=1}^{q} w_i \beta_{ik} R_{i(k)}}{\sum_{i=1}^{q} w_i \beta_{ik}^2} . \tag{6}$$

The Hermite polynomials can be considered to approximate to activation function $g_k(\cdot)$. The resulting $\hat{g}_k(\cdot)$ is

$$\hat{g}_k(X)=\sum_{r=1}^{R}\hat{C}_{kr}h_r(X). \tag{7}$$

The estimation $\hat{\alpha}_k$ ***of input layer weights*** α_k**.** Let L_2 be rewrite as follows:

$$L_2(\alpha_k)=\sum_{i=1}^{q}w_iE[\psi_i(\alpha_k)]^2. \tag{8}$$

where

$$\psi_i(\alpha_k)=R_{i(k)}-\sum_{l\neq k}\beta_{il}g_l(\alpha_l^{\mathrm{T}}X). \tag{9}$$

It is noted that L_2 loss function is not a quadratic function of α_k, The Gauss-Newton method is used to obtain $\hat{\alpha}_k$. The procedure will be continued until the error satisfied the demand.

Now we consider the problem about the univariate nonlinear autoregression modeling and forecasting via PPLN. The above learning strategy ([3]) of PPLN is suitable for univariate nonlinear time series modeling and forecasting. Let (Ω,F,P) be a probability space, (E,R) a measurable space, E a bounded closed set, R is a Borel on E. If $\{x_{t,}\ t\in T\}$ is a univariate stationary process defined on (Ω,F,P) and satisfies

$$x_t=f(x_{t-1},\cdots,x_{t-p})+\varepsilon_t. \tag{10}$$

where $f:R^p\rightarrow R$ is unknown measurable function. $\varepsilon_t\sim WN(0,\sigma^2)$, $\{x_{t,}\ t\in T\}$ is univariate nonlinear autoregression process of order p (NLAR (p)).

The learning criterion is

$$\min_{\alpha_j,\beta_j,g_j}\left\|f(X_t)-\sum_{j=1}^{m}\beta_jg_j\left(\alpha_j^TX_t\right)\right\|,$$

where $X_t=(x_{t-1},x_{t-2},\cdots x_{t-p}),\alpha_j^{\mathrm{T}}=\left(\alpha_{j1},\alpha_{j2},\cdots,\alpha_{jp}\right)$ $\alpha_j(1\leq j\leq m)$ is the weigh vector of input layer, $g_j(\cdot)(1\leq j\leq m)$ is the activation function of hidden layer, $\beta_j(1\leq j\leq m)$ is the weight of output layer.

2.3 The Learning Strategies of the PPLN for Multivariate Nonlinear Time Series Modeling and Forecasting

Definition 1 The n-dimensional stationary series $\{x(t),\quad t=0,\pm1,\pm2,\cdots\}$ is said to be n-dimensional nonlinear autoregression, if and only if $\{x(t)\}$ satisfies

$$x(t)=h(x(t-1),x(t-2),\cdots,x(t-p))+\varepsilon(t). \tag{11}$$

where

$$x(t) = \left(x_1(t), x_2(t), \cdots, x_n(t)\right)^{\mathrm{T}} \in R^n,$$
$$x(t-i) = \left(x_1(t-i), x_2(t-i), \cdots, x_n(t-i)\right)^{\mathrm{T}} \in R^n, i = 1, 2, \cdots, p,$$

$h(\cdot): R^{np} \to R^n$ is unknown smooth function, $\varepsilon_t \sim WN(0, \sigma^2)$.

Now we consider the problem about the n-dimensional nonlinear autoregression modeling and forecasting via PPLN. The learning criteria of minimizing the error loss function:

$$L_2 = \sum_{i=1}^{n} w_i E\left[x_i(t) - \sum_{k=1}^{m} \beta_{ik} g_k \left(\sum_{j=1}^{p} \sum_{l=1}^{n} \alpha_{jl}^{k} x_l(t-j)\right]^2,$$

where $\alpha_{jl}^{i}\left(1 \le i \le m,\ \ 1 \le l \le n,\ \ 1 \le k \le m\right)$ denotes the input-layer weight linked between the $i-th$ hidden activation function $g_i(\cdot)$ $\left(1 \le i \le m\right)$ and the $lj-th$ neuron $x_l(t-j)$ $\left(1 \le l \le n, 1 \le j \le p\right)$ of the input layer, $\beta_{ik}\left(1 \le k \le p,\ \ 1 \le i \le n\right)$ denotes the output layer weight linked between the $i-th$ output neuron $x_i(t)$ $\left(1 \le i \le n\right)$ and the k^{th} hidden activation function $g_k(\cdot)(1 \le k \le m), w_i(1 \le i \le n)$ is the learning rate that is decided by user. The calculating procedure for multivariate nonlinear time series is simulative to that for univariate nonlinear time series. The topological structure of the PPLN is shown in Figure 2.

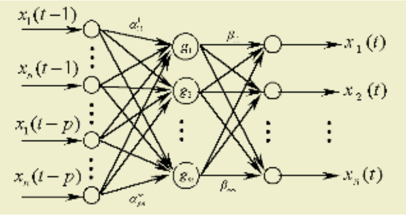


Fig. 2. The topological structure of the PPLN for n-dimensional NLAR (p)

3 The Convergence Property of Approximation of PPLN for Nonlinear Time Series

Suppose $\left(E^p, R^p, P_x(x_t)\right)$ be a probability space, where E^p is a bounded closed subset on R^p, R^p is a Borel on the E^p, $R^p = \overbrace{R \times R \times \cdots \times R}^{p}$, $E^p = \overbrace{E \times E \times \cdots \times E}^{p}$, $P_t(x_t)$ is the probability distribution of x_t, which is also the probability measure on R^p. Let S be a linear space spayed by $f_1, f_2, \cdots, f_n, \cdots,$ $\int_{B_t} |f(X_t)|^k \, dP_X(X_t) < \infty,$ $i = 1, 2, \cdots,$ $k \ge 1$, where for any $B_t \in R^p$, and S is a dense set in $L^k(E)$. The convergence property of approximation of nonlinear time series is as follows.

Theorem 1 Let $\{x_t, t \in T\}$ be an univariate nonlinear autoregression process

$$x_t = f(x_t) + \varepsilon_t ,$$

where $\int_{B_t} |f(x_t)|^k \, dP_X(x_t) < \infty, k \geq 1,$ then there are α_j, β_j and $g_j(\alpha_j^T X_t)$ such that

$$\int_{B_t} | f(X_t) - \sum_{j=1}^{m} \beta_j g_j (a_{\mathrm{j}}^{\mathrm{T}} X_t) |^k dP_X(X_t) \to 0, m \to \infty . \qquad (12)$$

where $g_j(\cdot) \in S, \beta_j$ are real constants, $j = 1, \cdots, m,$ α_j are real d-dimensional vector, $\|\alpha_j\| = 1,\ j = 1, \cdots, m$.

Theorem 2 Let $\{x_t, t \in T\}$ be a n-dimensional autoregression process

$$x_t = h(x(t-1), x(t-2), \cdots, x(t-p)) + \varepsilon(t) ,$$

where $\int_{B_t} |h(x_t)|^k \, dP_x(x_t) < \infty, k \geq 1$, then there are $\alpha_{jl}^k, \beta_{ik}$ and $g_k(\cdot)$ such that

$$\int_{B_t} | h(X_t) - \sum_{k=1}^{m} \beta_{ik} g_k (\sum_{j=1}^{p} \sum_{l=1}^{n} \alpha_{lj}^k x_j(t-j)) |^k \, dP_X(X_t) \to 0, m \to \infty ,$$

where $X_t = (x(t-1), x(t-2), \cdots, x(t-p))$.

4 Experimental Results and Analysis

4.1 The Annual Sunspot Numbers (1700 - 1987)

Wölfer sunspont numbers (1700-1979) are fitted and the numbers are predicated (1980-1987) with PPLN, whose number of input layer nodes is 4 and the that of hidden is 4. The results see Figure 3 and Figure 4, respectively. The weights of input layer nodes are as follows: 0.027, -0.278, -1.118, 2.500;2.836, -1.711,3.146, 1.875; 2.447, -1.761, -2.161, 5.325. The weights of output layer nodes are as follows: 0.696, 0.603, 0.678, and 0.1356. The predication mean square error is 488.788, and the mean square error of prediction for Tong (pp.421) is 683.960.

Table 1. The Prediction and Error of the Wölfer sunspont numbers (1980-1987)

Year	Observation	Predication		Error	
		SETAR	PPLN	SETAR	PPLN
1980	154.7	160.1	148.1	-5.4	-6.6
1981	140.5	141.8	138.8	-1.3	-1.7
1982	115.9	96.4	105.1	19.5	-10.8
1983	66.6	61.8	67.0	4.8	0.4
1984	45.9	31.1	46.8	14.8	0.9
1985	17.9	18.1	27.1	-0.2	9.2
1986	16.4	18.9	21.3	-5.5	7.9
1987	29.2	29.9	15.9	-0.7	13.3

4.2 The Canada Lynx Data from 1820 to 1934

The Canada lynx data (1820-1914) are fitted and the numbers are predicated (1921-1934) with the PPLN(see Figure 5 and Figure 6).

The mean square error of predication is 0.083, and the mean square error of prediction for Tong (1977pp.466) is 0.224(see table 1 and table 2).

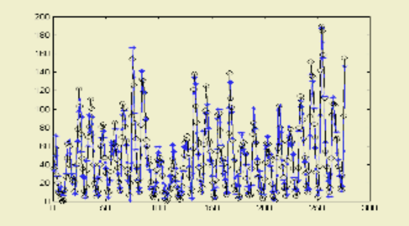


Fig. 3. The fitted results of the sunspont numbers (1700-1979) (__): actual data; (….): PPLN

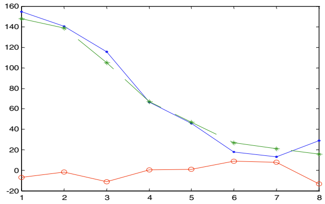


Fig. 4. The Prediction and error of the Wӧlfer sunspont numbers (1980-1987)

Table 2. The Prediction and Error of the Canada lynx data (1820-1914) of numbers (1980-1987)

Year	Observa-tion	Predication			Error		
		SETAR	AR (12)	PPLN	SETAR	AR (12)	PPLN
1921	2.3598	2.3109	2.4559	2.3476	0.0489	-0.0952	-0.0122
1922	2.6010	2.8770	2.8070	2.7297	-0.276	0.2060	0.1287
1923	3.0539	2.9106	2.8990	2.9882	0.1433	-0.1549	-0.0657
1924	3.3860	3.3703	3.2310	3.3548	-0.0157	-0.1550	-0.0312
1925	3.5532	3.5875	3.3880	3.5691	0.0343	-0.1652	0.0160
1926	3.4676	3.4261	3.3320	3.4778	0.0415	-0.1356	0.0102
1927	3.1867	3.0936	3.0070	3.1094	0.0931	0.1797	-0.0773
1928	2.7235	2.7706	2.6880	2.7840	0.0471	-0.0355	-0.1779
1929	2.6857	2.4217	2.4280	2.5079	0.2640	-0.0355	0.0606
1930	2.8209	2.7644	2.7650	2.7524	0.0565	-0.2577	-0.0685
1931	3.0000	2.9397	2.9840	3.0260	0.0603	-0.0559	0.0260
1932	3.2014	3.2462	3.2170	3.1970	-0.448	0.0156	-0.0044
1933	3.4244	3.3701	3.3650	3.3292	0.0543	0.0594	-0.0952
1934	3.53097	3.4468	3.5030	3.4628	0.08417	0.02797	-0.0682

5 Conclusions

A projection pursuit learning network is based on projection pursuit regression (PPR) and possesses the structure of a two-layer feed forward network structure. PPLN has the most successful learning network strategies: (1) it adaptively grows the network structure using all the observation data; (2) it uses network structures which were not limited in their approximation capability. Unfortunately, PPLN doesn't show ability in localized learning, and pre-selection of the activation function's nonlinearity (e.g., the order of Hermite polynomials) plays a critical role. To remedy the two defects in a PPLN, we propose a new learning network. This research work is now progress.

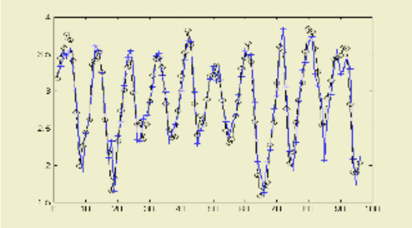


Fig. 5. The fitted results of the Canada lynx data (1821-1919) numbers (__):actual data; (….): PPLN**.**

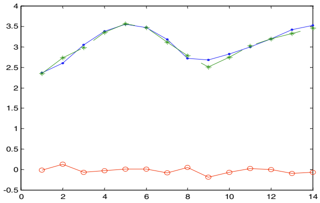


Fig. 6. The predicators of the lynx data (1921-1934)

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No.60375003) and Aviation Foundation of China (No.03I5309).

References

1. Barron, A. R., Barron, R. L.: Statistical Learning Networks: A Unifying View. Proceedings 20th Symposium Inferface, Ed. Wegman, Ed. Washington, D.C.: American Statist. Assoc (1988) 192-203
2. Jones, L.K.: A Simple Lemma on Greedy Approximation in Hilbert Space and Convergence Rates for Projection and Neural Networks. Ann. Statist, **20** (1992) 608-613
3. Hwang, J.: Regression Modeling in Back-Propagation and Projection Pursuit Learning. IEEE Trans. Neural Networks, **5** (1994) 342-353
4. Tong, H.: Non-linear Time Series: A Dynamic System Approach. Clarendon Press Oxford (1990)
5. Tong, H.: Threshold Autoregression, Limit Cycles And Cyclical Data. J. R. Statist. Soc.B, **42** (1980) 245-292
6. Tong, H.: Dimension Estimation and Models. Singapore; River Edge, World Scientific (1993)
7. Tong, H.: Chaos and Forecasting. Proceedings of Royal Society Discussion Meeting, Singapore, World Scientific (1994) 2-3
8. Perter, J. B, Richad, A. D.: Time Series：Theory and Methods. Springer-Verlag (1990)
9. Tian, Zh.: The Study of Computer-Intensive Methods：Non-Parametric Regression and Non-Linear Time Series Models. University Dortmund Germany (1994)
10. Chan, K, Tong, H.: Chaos: A Statistical Perspective. New York Springer (2001)

A New Scheme for Blind Decomposition of Mixed Pixels Based on Non-negative Matrix Factorization*

Hao Zhou[1], Bin Wang[1,2], and Liming Zhang[1]

[1] Department of Electronics Engineering, Fudan University, Shanghai 200433, China
{032021038,wangbin,lmzhang}@fudan.edu.cn
[2] Key Laboratory of Wave Scattering and Remote Sensing Information
(Ministry of Education), Fudan University, Shanghai 200433, China

Abstract. In this paper, we propose a new blind decomposition scheme for mixed pixels in multichannel remote sensing images. This scheme does not depend on information from spectrum database or pure endmember pixels and can be used to decompose mixed pixels by using Non-negative Matrix Factorization (NMF). Principal Component Analysis (PCA) is proposed to determine the number of endmembers in remote sensing images and a constraint for NMF that the sum of percentages concerning each endmember equals one is introduced in the proposed scheme. Experimental results obtained from both artificial simulated and practical remote sensing data demonstrate that the proposed scheme for decomposition of mixed pixels has excellent analytical performance.

1 Introduction

Usually, ground objects in remote sensing images are detected in unit of pixels. Due to the limit of spatial resolution, in most cases, one pixel may cover hundreds of square meters with various ground objects and becomes a mixed pixel. The mixed pixel problem not only influences the precision of object recognition and classification, but also becomes an obstacle to quantitative analysis of remote sensing images. This problem can be overcome by precisely obtaining the percentages of object of interest. In fact, the exact decomposition of mixed pixels is very important in the field of subpixel classification of multichannel remote sensing image as well as detection and identification of ground objects.

A variety of methods using pure ground object pixel obtained from multichannel remote sensing images or ground object spectral database are widely used in the field of decomposition of mixed pixels [1]-[6]. These methods demand complete spectral database or pure pixels obtained from multichannel remote sensing images, which in fact is very difficult. For example, some ground objects are not included in spectral database, or pure pixel does not exist in some remote sensing images. In these circumstances, it is very significant if we can make the mixed pixel decomposed blindly.

* This research was supported in part by the grants from the Major State Basic Research Development Program of China (No. 2001CB309401), the National Natural Science Foundation of China (No. 30370392 and No. 60171036), Hang Tian Support Techniques Foundation (No. 2004-1.3-03), and Shanghai NSF (No. 04ZR14018).

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 651–658, 2005.

Independent component analysis (ICA) is a method to decompose multichannel signals blindly [7]. But ICA cannot meet the constraints that the percentage of each decomposed result should be non-negative and the sum of percentages of decomposed results should be 1.

Non-negative matrix factorization (NMF), a new method proposed by Daniel. D. Lee *et al* [8], [9], decomposes a positive matrix into a product of two positive matrices and has been used only to resolve some problems in the fields of face recognition and semantic analysis of text documents [8][10]. In this paper, we propose a scheme based on constrained NMF to blindly decompose mixed pixels in multichannel remote sensing images. Applying NMF method to try to solve the problem of decomposition of mixed pixels constitutes the first originality of our research. Additionally, we adopt principal component analysis (PCA) to determine the number of endmembers in multichannel remote sensing images which constitutes the second originality of our work. Finally, some experiments on artificial simulated images and practical remote sensing images are performed to test the proposed scheme. From the experimental results we can find that this new scheme is very effective in finishing the blind decomposition of mixed pixels, and also it operates very fast.

The remainder of this paper is organized as follows. Section 2 is used to describe the proposed scheme. Some experimental results are shown in section 3. Conclusion is given in section 4.

2 The Proposed Scheme

2.1 Linear Spectral Mixture Analysis

Over the past years, linear spectral mixture analysis (LSMA) has been widely used for mixed pixel decomposition. It assumes that the spectral signature of an image pixel is linearly mixed by the spectral signatures of objects present in the image. Define $\boldsymbol{X}$ as a multichannel vector of a single pixel in multichannel remote sensing images, and $\boldsymbol{A}$ as a reflectance characteristic matrix composed of reflectance of each object in each spectral band, $\boldsymbol{S}$ as a vector composed of the percentage of each object, and $\boldsymbol{N}$ as a vector realized from spatially white Gaussian noise. So we can obtain the equation

$$\boldsymbol{X} = \boldsymbol{A} \times \boldsymbol{S} + \boldsymbol{N} . \tag{1}$$

For additive white Gaussian noise, it is usually weak for some mature multichannel imaging techniques, especially for multispectral imaging techniques. Here, we omit the effect of additive white Gaussian noise for the convenience of analysis.

If the multichannel remote sensing images have n bands and m sorts of interesting objects, then $\boldsymbol{X}$ is a $n \times 1$ vector, $\boldsymbol{A}$ is a $n \times m$ matrix and $\boldsymbol{S}$ is a $m \times 1$ vector. In this model, the selection of matrix $\boldsymbol{A}$ is important to the precision of decomposition results.

2.2 Constrained Conditions

In practice, the decomposition results $\boldsymbol{S}$ based on LSMA should satisfy the following three constraints.

(a) The sum of percentages s_i of ground objects in every single pixel should be 1, *i.e.*

$$\sum_{i=1}^{m} s_i = 1 \cdot \tag{2}$$

(b) The percentages s_i of ground objects should be non-negative, *i.e.*

$$s_i \geq 0\,,\ (i = 1,2,...,m)\,. \tag{3}$$

(c) The decomposition results $\boldsymbol{S}$ should satisfy the condition of LSMA model, *i.e.*

$$\boldsymbol{X} = \boldsymbol{A} \times \boldsymbol{S}\,. \tag{4}$$

In addition, because the spectral reflectances of ground objects are a sort of energy, they cannot be negative, the elements a_{ij} of matrix $\boldsymbol{A}$ should be non-negative, *i.e.*

$$a_{ij} \geq 0\,. \tag{5}$$

2.3 Non-negative Matrix Factorization

Non-negative matrix factorization is a new method to decompose a positive matrix into a product of two positive matrices using positive constraint. Given an initial positive $n \times m$ matrix ***V***, by iterative operation, we can find two positive matrices ***W*** and ***H***, and make them satisfy the following equation:

$$V = W \times H\,. \tag{6}$$

In Equation (6), ***W*** is a $n \times r$ matrix and ***H*** is a $r \times m$ matrix. *r* should be given a fixed value in advance. Usually, *r* is less than *m* and *n*.

In NMF, no negative entries are allowed in matrix factors ***W*** and ***H*** whereby non-negativity constraint is imposed in factorizing the data matrix ***V*** limiting data manipulation only to additions (no subtractions are allowed). Each column in the matrix ***W*** is called a basis image, and a column in the matrix ***H*** is called an encoding. An image in ***V*** can be reconstructed by linearly combining basis image with the coefficients in an encoding. The encoding influences the activation of pixels in the original matrix via basis images.

Given a data matrix ***V***, Lee *et al*[8][9] developed a technique for factorizing the ***V*** to yield matrices ***W*** and ***H***. In order to find the factorization matrices, an objective function is defined as

$$\left|V - W \times H\right|^2 = \sum_{i,j} \left(V_{i,j} - (W \times H)_{i,j}\right)^2. \tag{7}$$

Deal with the Equation (7) and set iterative step η as

$$\eta_{i,j} = \frac{H_{i,j}}{(W^T W H)_{i,j}}\,, \quad (i=1,2,\ldots,r;\ j=1,2,\ldots,m). \tag{8}$$

The following iterative learning rules are used to find the linear decomposition

$$H_{i,j} \leftarrow H_{i,j} \frac{\left(W^T V\right)_{i,j}}{\left(W^T W H\right)_{i,j}}, \ (i=1,2,\ldots,r;\ j=1,2,\ldots,m), \tag{9}$$

$$W_{i,j} \leftarrow W_{i,j} \frac{\left(V H^T\right)_{i,j}}{\left(W H H^T\right)_{i,j}}, \ (i=1,2,\ldots,n;\ j=1,2,\ldots,r). \tag{10}$$

Here, r represents the number of basis images and the number of coefficients in an encoding.

The above unsupervised multiplicative learning rules are used iteratively to update ***W*** and ***H***. The initial values of ***W*** and ***H*** are fixed randomly. At each iteration, a new value for ***W*** and ***H*** is evaluated. Each update consists of a multiplication and sums of positive factors. With these iterative updates, the quality of the approximation of the equation (6) improves monotonically with a guaranteed convergence to optimal matrix factorization.

2.4 Apply NMF to Decomposition of Mixed Pixels

Usually, ***V*** is considered as a matrix composed of n vectors, ***W*** is a matrix composed of a set of basis vectors and ***H*** is a matrix composed of weight values of ***V*** projected into ***W*** space. In the problem of decomposition of mixed pixels, we see ***V*** as a n-band multichannel remote sensing images, and then, ***W*** is a reflectance characteristic matrix composed of reflectance of each object in each spectral band, and ***H*** is a matrix composed of the percentage of each object in each pixel.

We can find two positive matrices ***W*** and ***H*** by using NMF, and the two matrices satisfy the equation $\boldsymbol{V}=\boldsymbol{W}\times\boldsymbol{H}$, so the result of NMF satisfies the constraints (3), (4) and (5). In order to make the result satisfy the constraint (2), we propose to apply the following iterative operation to NMF algorithm:

$$H_{(i,j)} \leftarrow \frac{H_{(i,j)}}{\sum_{i=1}^{n} H_{(i,j)}}, \ (i=1,2,\ldots,r;\ j=1,2,\ldots,m). \tag{11}$$

The application of a constrained NMF method to the problem of decomposition of mixed pixels constitutes the first originality of our work.

Because r should be given in advance in NMF, before applying the NMF method to decompose mixed pixels, we propose to obtain r by a preprocessing step which constitutes the second originality of our work. Principal component analysis (PCA) can be used to solve this problem. Because the pixels in images satisfy the equation $\boldsymbol{V}=\boldsymbol{W}\times\boldsymbol{H}$, so we can get r by observing the number of large eigenvalues.

Finally, we simply summarize the computation process of the proposed algorithm as follows:

Step 1. Rectify the data to minimize the influence of atmosphere.

Step 2. Preprocess the n-band multichannel remote sensing images with PCA to obtain r.

Step 3. Set ***W*** and ***H*** randomly, and set total iterative number.
Step 4. Update ***W*** and ***H*** with the equation (9) and (10).
Step 5. Update ***H*** with the equation (11).
Step 6. If the iterative number reaches the predefined value, the iterative process ends; otherwise, return to Step 4.

3 Experimental Results

In this section, the experiments are performed on artificial simulated images, practical Landsat multispectral images and hyperspectral images.

3.1 Experiments for Simulated Images

In order to test the performance of our algorithm, first, the experiment is carried out for simulated images. The process is shown in the following:

1. Produce three 1×10000 positive vectors randomly and use them to simulate the percentages of three ground objects in each pixel.

2. Make the three vectors satisfy the constraint (2) by preprocessing.

3. By defining the artificial matrix ***W***, we can obtain simulated images with LSMA model and use them to simulate the mixed pixels in multispecrtral or hyperspectral images. Here we define 5×3 matrix ***W*** below

$$W = \begin{bmatrix} 20 & 50 & 120 \\ 160 & 80 & 200 \\ 100 & 180 & 60 \\ 70 & 30 & 150 \\ 210 & 160 & 230 \end{bmatrix}.$$

4. Process and analyze the data with the proposed algorithm. The eigenvalues obtained by PCA are 6.3616×10^8, 3.1824×10^7, 5.4832×10^7, 1.1528×10^{-8} and 8.1256×10^{-9}, so we set r=3 here.

5. Compare the experiment results with the original data. Calculate their correlation coefficients and use them to evaluate the performance of the proposed algorithm.

In the experiment, we produce 3 random images. For the convenient observation, we cut 3 images (25×25) from original images and show them in Fig. 1. The mixed 5 simulation images are shown in Fig. 2. The experimental results obtained by the proposed algorithm are shown in Fig. 3. From the images, we can see that the proposed algorithm works very well in decomposing mixed pixels. The decomposed result strongly resembles the original data. Their correlation coefficients are shown in Table.1. In this experiment, we set the total iterative number as 500 and it takes Pentium 4 1.70GHz CPU 7.0360 seconds to decompose 5-band images (100×100).

Table 1. Correlation coefficients between the 3 decomposed images and the 3 original images

Random image 1	Random image 2	Random image 3
0.9982	0.9886	0.9993

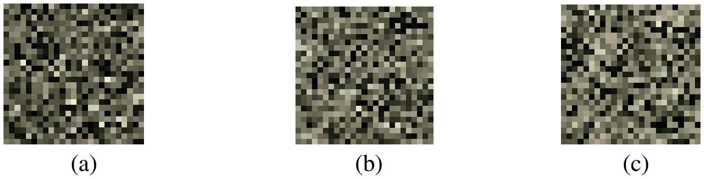


Fig. 1. Three original images (25×25) produced randomly

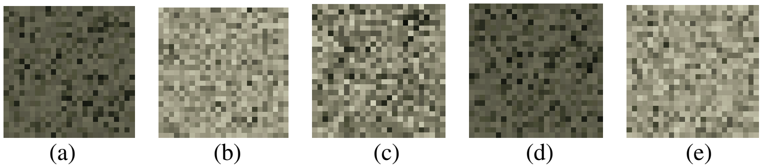


Fig. 2. Simulated multichannel remote sensing images (25×25)

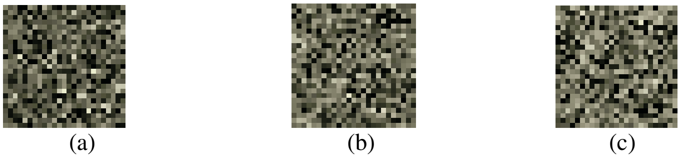


Fig. 3. Three decomposed images (25×25)

3.2 Experiments for Landsat Multispectral Images

In this part, we used 1st – 5th and 7th bands of Landsat multispectral images obtained by Landsat 7 ETM+ on July 14, 2000. We use the multispectral images which covers an area of Shanghai as experimental data (the image size is 256×256). Fig. 4 shows a Landsat false colour image of this area, with the band 3 displayed as blue, the band 4 as green, and the band 5 as red. As shown in Fig. 4, this area mainly includes 3 sorts of typical ground objects: water, plant and soil (include artificial architecture).

In the experiment, we apply the proposed algorithm to the Landsat multispectral images. The eigenvalues obtained by PCA are 4.5698×10^8, 1.0534×10^7, 3.2636×10^6, 2.6823×10^{-3}, 3.0569×10^{-5} and1.0277×10^{-5}, so we set r=3, and it is identical with ground fact.

The decomposition results of the proposed algorithm are shown in Fig. 5. In these images, pure black denotes that the percentage of a certain sort of object in this pixel is 0, while pure white denotes 1. From the results we can see that the proposed algorithm is successful in decomposing the Landsat Multispectral Images into 3 ground objects. In this experiment, we still set the total iterative number as 500 and it takes Pentium 4 1.70GHz CPU 9.6168 seconds to decompose 6-band images.

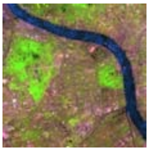


Fig. 4. Landsat remote sensing image covering Shanghai region

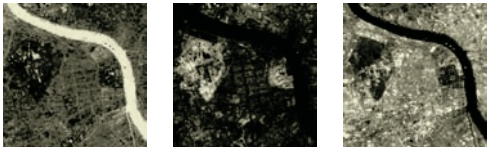


Fig. 5. The results obtained by the proposed scheme from the multispectral images

3.3 Experiments for Hyperspectral Images

In the final experiment, we apply the proposed algorithm to 80-band GPS hyperspectral images (the image size is 256×256). These images covering another area of Shanghai were acquired by a platform at altitude 1632.5m. The graph of logarithm of the largest 20 eigenvalues obtained by PCA is shown in Fig. 6. The horizontal axis presents the number of eigenvalues and the vertical axis dose the logarithm of magnitudes of eigenvalues. Here, we set *r*=4. The decomposition results of the proposed algorithm are shown in Fig. 7.

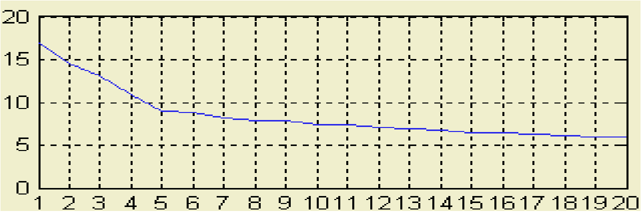


Fig. 6. The graph of logarithm of the largest 20 eigenvalues

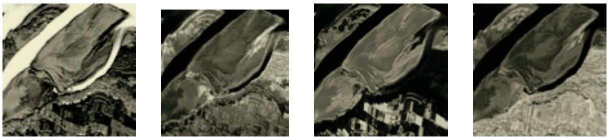


Fig. 7. The results obtained by the proposed scheme from the hyperspectral images

4 Conclusions

In this paper, we have improved the NMF algorithm and make it satisfy the constrained conditions for the problem of decomposition of mixed pixels. With the proposed algorithm, we can blindly decompose the mixed pixels in multichannel remote sensing images. In order to determine the number of ground objects, we preprocess the data by using PCA. The experiments of simulated data and real-world data confirmed the validity of the proposed algorithm. In addition, we point out that the proposed algorithm is simple, fast and precise for the blind decomposition of mixed pixel in multichannel remote sensing images, and should be very useful for the applications of object detection and recognition.

References

1. Chang, C.I., Zhao, X.L.: Least Squares Subspace Projection Approach to Mixed Pixel Classification for Hyperspectral Images. IEEE Transactions on Geoscience and Remote Sensing, **36** (1998) 898-912
2. Muramatsu, K., Furumi, S., Fujiwara, N., Hayashi, A., Daigo, M., Ochiai, F.: Pattern Decomposition Method in the Albedo Space for Landsat TM and MSS Data Analysis. International Journal of Remote Sensing, **21** (2000) 99-119
3. Peng, T., Li, B., Su, H.: A Remote Sensing Image Classification Method Based on Evidence Theory and Neural Networks. Proceedings of the 2003 international Conference on Neural Networks and Signal Processing, **1** (2003) 240-244
4. Heermann, P. D., Khazenie, N.: Classification of Multispectral Remote Sensing Data Using a Back-Propagation Neural Network. IEEE Transactions on Geoscience and Remote Sensing, **30** (1992) 81-88
5. Chang, C.I., Ren, H., Chang, C.C., D'Amico, F., Jensen, J.O.: Estimation of Subpixel Target Size for Remotely Sensed Imagery. IEEE Transactions on Geoscience and Remote Sensing, **42** (2004) 1309-1320
6. Du, Q., Chang, C.I.: An Interference Rejection-Based Radial Basis Function Neural Network for Hyperspectral Image Classification. International Joint Conference on Neural Networks (1999) 2698-2703
7. Hyvärinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. Wiley, New York (2001)
8. Daniel, D., Lee, H., Seung, S.: Learning the Parts of Objects by Non-negative Matrix Factorization. Nature, **401** (1999) 788–791
9. Daniel, D., Lee, H., Seung, S.: Algorithms for Non-negative Matrix Factorization. Advances in Neural Information Processing Systems, **13** (2001) 556-562
10. Rajapakse, M., Wyse, L.: NMF V.S. ICA for Face Recognition. Proceedings of the 3rd International Symposium on Image and Signal Processing and Analysis (PROC.ISPA03) (2003) 605-610

Representing Image Matrices: Eigenimages Versus Eigenvectors

Daoqiang Zhang[1,2], Songcan Chen[1], and Jun Liu[1]

[1] Department of Computer Science and Engineering
Nanjing University of Aeronautics and Astronautics
Nanjing 210016, China
{dqzhang,s.chen}@nuaa.edu.cn
[2] National Laboratory for Novel Software Technology
Nanjing University, Nanjing 210093, China

Abstract. We consider the problem of representing image matrices with a set of basis functions. One common solution for that problem is to first transform the 2D image matrices into 1D image vectors and then to represent those 1D image vectors with *eigenvectors*, as done in classical principal component analysis. In this paper, we adopt a natural representation for the 2D image matrices using *eigenimages*, which are 2D matrices with the same size of original images and can be directly computed from original 2D image matrices. We discuss how to compute those eigenimages effectively. Experimental result on ORL image database shows the advantages of eigenimages method in representing the 2D images.

1 Introduction

Principal component analysis (PCA) [3] is a well-known data representation technique widely used in pattern recognition and signal processing [2], [4], [5]. When using PCA to represent 2D images, we have to first transform the 2D image matrices into 1D image vectors, then compute the corresponding 1D *eigenvectors* from the sample covariance matrix to represent any image vector as a weighted sum of a set of eigenvectors, and finally retransform the 1D sum vector back to a 2D matrix to obtain the reconstructed image. However, the vectorized representation of image has the following disadvantages. Firstly, concatenating a 2D image often leads to a corresponding high-dimensional 1D vector, which makes it very difficult and time-consuming to compute the corresponding eigenvectors. Secondly, such a concatenation for 2D image may cause the loss of some structure information hiding in original 2D image.

To overcome those problems, Yang *et al.* proposed two-dimensional PCA (2DPCA) [6] which directly compute eigenvectors of the so-called *image covariance matrix* without matrix-to-vector conversion. However, 2DPCA still use 1D eigenvectors to represent image matrices which will be detailed in the next section, if we view each row in the 2D image matrices as an image vector, then 2DPCA can be approximately seen as conventional PCA operated on those row vectors. Thus the obtained

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 659–664, 2005.

1D eigenvectors only reflect the row information in original images, and we call them as *row eigenvectors*. And each row of original 2D images is represented as a weighted sum of a set of those row eigenvectors. That is, 2DPCA represents 2D image matrices using 1D eigenvector in essence. A more recent development on that aspect is the generalized low rank approximations of matrices (GLRAM) [7] proposed by Ye. In GLRAM, two groups of 1D eigenvectors $L=\{l_1, l_2, \ldots, l_p\}$ and $R=\{r_1, r_2, \ldots, r_q\}$ are jointly used to represent 2D images. However, unlike in PCA and 2DPCA, GLRAM cannot find the global optimal solutions for L and R, and instead an iterative algorithm is used to find locally optimal solutions. Thus GLRAM needs a relatively more computational cost than 2DPCA.

In this paper, we make further step along with 2DPCA. As in 2DPCA, we compute the row eigenvectors r_j $(1 \le j \le p)$ by viewing each row as an image vector. Similarly, we can obtain the *column eigenvectors* l_i $(1 \le i \le q)$ by viewing each column as an image vector. Then we define an *eigenimage* as the outer-product between l_i and r_j and represent original images with those 2D eigenimages. Although the proposed method is very similar to GLRAM, there do exist at least two differences. First, although GLRAM also compute the two groups of eigenvectors L and R, there is no explicit definition for the eigenimage and thus no discussion on the interesting characteristics on eigenimages, as shown in the next section. Moreover, the eigenvectors L and R in GLRAM are computed with iterative steps and a good initial value is needed, while in our method L and R are computed both in closed-form and in a parallel manner. Experimental result on ORL image database shows the advantages of the proposed method in representing the 2D images compared with PCA, 2DPCA and GLRAM.

2 Image Representation with Eigenimages

Let $A^k \in R^{r\times c}, k = 1,\ldots,n$ denote original image matrices, compute the mean image as $\overline{A} = \frac{1}{n}\sum_{k=1}^{n} A^k$, then we get the corresponding centered images $\overline{A}^k = A^k - \overline{A}$. We first compute the row eigenvectors as follows.

Concatenating the n image matrices $\overline{A}^k$ into an nr by c single matrix according to the following order: $\overline{A}_{row} = \left((\overline{A}^1)^T, (\overline{A}^2)^T, \ldots, (\overline{A}^n)^T\right)^T$, then

$$\left(\overline{A}_{row}\right)^T \left(\overline{A}_{row}\right) = \sum_{k=1}^{n} \left(\overline{A}^k\right)^T \left(\overline{A}^k\right) = \sum_{k=1}^{n}\sum_{j=1}^{r} \left(\overline{a}_j^k\right)^T \left(\overline{a}_j^k\right) = \sum_{i=1}^{nr} b_j^T b_j \, . \tag{1}$$

Here $\overline{a}_j^k$ is the j-th row of image matrix $\overline{A}^k$ and b_j is the j-th row of $\overline{A}_{row}$. Note that Eq. (1) is in fact n times of the *image covariance matrix* $\Sigma = \frac{1}{n}\sum_{k=1}^{n}\left(\overline{A}^k\right)^T \left(\overline{A}^k\right)$

used in 2DPCA and nr times of the *covariance matrix* $\Sigma' = \frac{1}{nr}\sum_{i=1}^{nr} b_j^T b_j$ when viewing each row b_j as an image vector. However, according to matrix theory multiplying a matrix by a constant does not change its eigenvectors [1], so Σ has the same eigenvectors as Σ'. That is, what 2DPCA really computes is only the eigenvectors corresponding to nr row vectors. We name those *c* eigenvectors as *row eigenvectors* and denote them as $r_1, r_2, ..., r_c$. Here r_1 is the eigenvector corresponding to the largest eigenvalue and r_c the eigenvector corresponding to the smallest eigenvalue.

Following a similar procedure, we can obtain the column eigenvectors. By concatenating the *n* image matrices $\overline{A}^k$ into an *r* by *nc* single matrix according to the following order: $\overline{A}_{col} = \left(\overline{A}^1, \overline{A}^2, ..., \overline{A}^n\right)$, we have

$$\left(\overline{A}_{col}\right)\left(\overline{A}_{col}\right)^T = \sum_{k=1}^{n}\left(\overline{A}^k\right)\left(\overline{A}^k\right)^T = \sum_{k=1}^{n}\sum_{j=1}^{c}\left(\overline{a}_j^k\right)\left(\overline{a}_j^k\right)^T = \sum_{i=1}^{nc} b_j b_j^T . \qquad (2)$$

Here $\overline{a}_j^k$ is the *j*-th column of image matrix $\overline{A}^k$ and b_j is the *j*-th column of $\overline{A}_{col}$. By computing the eigenvectors of Eq. (2), we obtain the *column eigenvectors*. And we denote the *r* column eigenvectors as $l_1, l_2, ..., l_r$. Here l_1 is the eigenvector corresponding to the largest eigenvalue and l_r the eigenvector corresponding to the smallest eigenvalue.

Now we are in a position to introduce the concept of *eigenimage*. Given *c* row eigenvectors $r_1, r_2, ..., r_c$ and *r* column eigenvectors $l_1, l_2, ..., l_r$, define *eigenimage* as

$$E_{ij} = l_i \cdot r_j^T, (1 \le i \le r, 1 \le j \le c). \qquad (3)$$

It is easy to verify that the *eigenimage* E_{ij} has the following characters:

(1) E_{ij} is a 2D matrix with the same size of original image, i.e. $r \times c$.

(2) The *intrinsic dimensionality* or *rank* of E_{ij} is no more than 1.

(3) Any two eigenimages E_{kl} and E_{mn}, satisfying

$$trace\left(\left(E_{kl}\right)^T\left(E_{mn}\right)\right) = \begin{cases} 1, & k = m \ and \ l = n \\ 0, & otherwise \end{cases} . \qquad (4)$$

(4) Any image A^k ($1 \le k \le n$) can be represented by a weighted sum of eigenimages plus the mean image $\overline{A}$ as

$$A^k = \sum_{i=1}^{r}\sum_{j=1}^{c} D_{ij}^k E_{ij} + \overline{A}\,. \tag{5}$$

(5) The coefficients D_{ij}^k in Eq. (5) is computed using

$$D_{ij}^k = l_i^T \overline{A}^k r_j, (1 \le i \le r, 1 \le j \le c)\,. \tag{6}$$

From Eqs. (5) and (6), we accurately represent the original image A^k with rc eigenimages. In fact, we can approximately represent image A^k with partial ($< rc$) eigenimages. If we choose the $q(<c)$ row eigenvectors corresponding to the largest q eigenvalues of Eq. (1) and the $p(<r)$ column eigenvectors corresponding to the largest p eigenvalues of Eq. (2). Then we can approximately represent image A^k as $\hat{A}^k = \sum_{i=1}^{p}\sum_{j=1}^{q} D_{ij}^k E_{ij} + \overline{A}$ with only pq eigenimages. And $\hat{A}^k$ is also called reconstructed image in image compression.

We conclude this section by giving the detailed description of algorithm Eigenimage as follows.

```
Algorithm 'Eigenimage':
Step 1: Input $n$ $r \times c$ image matrices $A^1, A^2, ..., A^n$, and fix
    $p(\le r)$ and $q(\le c)$.
Step 2: Compute the mean image $\overline{A} = \frac{1}{n}\sum_{k=1}^{n} A^k$ and generate $\overline{A}^k = A^k - \overline{A}, (1 \le k \le n)$.
Step 3: Concatenate $\overline{A}^k (1 \le k \le n)$ into
    $\overline{A}_{row} = \left((\overline{A}^1)^T, (\overline{A}^2)^T, ..., (\overline{A}^n)^T\right)^T$, and compute the $q$ row eigenvectors $\{r_1, r_2, ..., r_q\}$ corresponding to the $q$ largest eigenvalues of $\left(\overline{A}_{row}\right)^T \left(\overline{A}_{row}\right)$.
Step 4: Concatenate $\overline{A}^k (1 \le k \le n)$ into $\overline{A}_{col} = \left(\overline{A}^1, \overline{A}^2, ..., \overline{A}^n\right)$,
    and compute the $p$ row eigenvectors $\{l_1, l_2, ..., l_p\}$ corresponding the $p$ largest eigenvalues of $\left(\overline{A}_{col}\right)\left(\overline{A}_{col}\right)^T$.
Step 5: Compute the eigenimages
    $E_{ij} = l_i \cdot r_j^T, (1 \le i \le p, 1 \le j \le q)$.
Step 6: Compute the coefficients
    $D_{ij}^k = l_i^T \overline{A}^k r_j, (1 \le i \le p, 1 \le j \le q, 1 \le k \le n)$.
```

```
Step 7: Compute the reconstructed image
    $\hat{A}^k = \sum_{i=1}^{p}\sum_{j=1}^{q} D_{ij}^k E_{ij} + \bar{A}, (1 \le k \le n)$ .
```

3 Experimental Results

In this section, we compare the performances of the proposed Eigenimage method with those of PCA, 2DPCA, and GLRAM on ORL face database. We are more concerned on the image reconstruction quality measured by PSNR and the time required by algorithms. All of our experiments are carried out on a PC machine with P4 1.7GHz CPU and 256MB memory.

We use the cropped ORL database in our first experiment. The cropped ORL face database consists of 400 different grey scale images of 40 different persons, with a resolution of 64×64. PCA first concatenating each 64×64 image to a 4094-dimensional vector, and then compute the corresponding eigenvectors, which is a very time-consuming procedure because the 4094×4094 covariance matrix is very huge. In our experiments, we adopt a trick introduced in [1] which need only computing the eigenvectors of an n×n matrix, and n is the size of face database. The left part of Fig. 1 shows the 36 eigenvectors of PCA corresponding to the largest 36 eigenvalues. Note these 1D eigenvectors are retransformed back to 2D images for visualization. We can see from the figure that these eigenvectors looks like the faces, i.e. they are dependent to the input data. On the other hand, the right part of Fig. 1 shows the 36 eigenimages of the Eigenimage method. We are surprised to see that, unlike PCA, these eigenimages do not look like faces any more, but show themselves with some 'regular' strip or block structures.

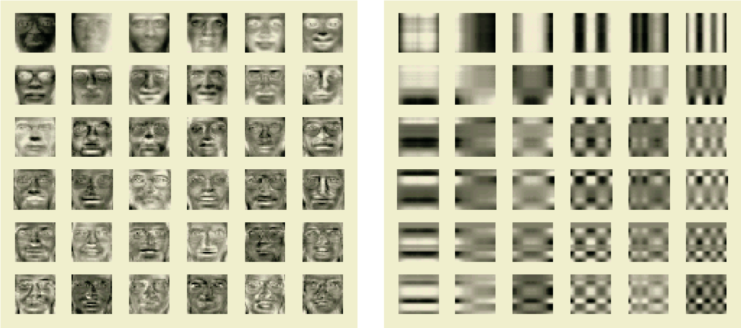


Fig. 1. Eigenvectors (Left) and Eigenimages (Right) on ORL database

Table 1 gives the detailed comparisons of four methods concerning the image reconstruction quality (measured with PSNR), compression ration (CR) and the required time. Here PCA and 2DPCA use 32 and 6 eigenvectors respectively, while GLRAM and Eigenimage both use 18 row and column eigenvectors. From Table 1,

Table 1. Comparisons of performances of four methods on ORL database

Methods	PSNR	CR	Time (s)
PCA	25.19	11.39	8.75
2DPCA	24.96	10.64	0.28
GLRAM	28.60	12.42	6.14
Eigenimage	28.58	12.42	0.59

2DPCA has the fastest speed but its compressed image quality is the worst under the same compression ratio, because it needs too many coefficients to represent an image. On the opposite, GLRAM has the best compressed image quality but it takes relatively more time, because there exist iterative steps in its solving procedure. The proposed Eigenimage achieves a good tradeoff between the compressed image quality and the speed. As seen from Table 1, the compressed image quality of Eigenimage is only slightly (about 0.02 db) worse than that of GLRAM, but its speed is tens of times higher than that of GLRAM. In fact, the required time of Eigenimage is about 2 times of 2DPCA, because Eigenimage can be seen as performing 2 times of 2DPCA in row and column, respectively.

Acknowledgements

This work was supported by China Postdoctoral Science Foundation, Jiangsu Planned Projects for Postdoctoral Research Funds, and National Science Foundation of China under the Grant No. 60473035.

References

1. Golub, G.H.: Van Loan, C.F.: Matrix Computations. The Johns Hopkins University Press, MD (1996)
2. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice-Hall (1999)
3. Jolliffe, I.T.: Principal Component Analysis. Springer-Verlag, New York (1986)
4. Kirby, M., Sirovich, L.: Application of the KL Procedure for the Characterization of Human Faces. IEEE Trans. PAMI, **12** (1990) 103-108
5. Turk, M., Pentland, A.: Eigenfaces for Recognition. J. Cognitive Neurosci, **3** (1991) 71-86
6. Yang, J., Zhang, D., et al.: Two-dimensional PCA: A New Approach to Appearance-based Face Representation and Recognition. IEEE Trans. PAMI, **26** (2004) 131-137
7. Ye, J.: Generalized Low Rank Approximation of Matrices. Proceedings of ICML (2004)

A SIMD Neural Network Processor for Image Processing

Dongsun Kim[1], Hyunsik Kim[1], Hongsik Kim[2],
Gunhee Han[2], and Duckjin Chung[3]

[1] DMB Project Office, Korea Electronics Technology Institute
455-6 MaSan-Ri, JinWi-Myon, PyungTaek-Si, KyungGi-Do 451-865, Korea
{dskim,hskim}@keti.re.kr
[2] Department of Electrical & Electronic Engineering, Yousei University, Seoul, Korea
{hskim,gunhee}@yonsei.ac.kr
[3] Information Technology and Telecommunications, Inha University
253 Younghyun-Dong, Nam-Gu, Incheon 402-751, Korea
djchung@inha.ac.kr

Abstract. Artificial Neural Networks (ANNs) and image processing requires massively parallel computation of simple operator accompanied by heavy memory access. Thus, this type of operators naturally maps onto Single Instruction Multiple Data (SIMD) stream parallel processing with distributed memory. This paper proposes a high performance neural network processor whose function can be changed by programming. The proposed processor is based on the SIMD architecture that is optimized for neural network and image processing. The proposed processor supports 24 instructions, and consists of 16 Processing Units (PUs) per chip. Each PU includes 24-bit 2K-word Local Memory (LM) and a Processing Element (PE). The proposed architecture allows multichip expansion that minimizes chip-to-chip communication bottleneck. The proposed processor is verified with FPGA implementation and the functionality is verified with character recognition application.

1 Introduction

Massively parallel computation of simple operator in neural networks and image processing suggest analog implementation as an attractive choice. Analog implementation of neural-network processor has advantages in low-power and small silicon area [1,2]. Although many analog implementations have been reported and commercialized as well, the reliability of analog computation is severely degraded as the array size increases due to component mismatch. The digital implementations of neural network processor have been reported in 90s [4,5]. This approach requires large silicon area while application range of the specialized neural network processor is limited. Drastic improvement of general purpose DSP performance and price makes the DSP based implementation more attractive than neural network processor development. Late 90s and early 2000, digital neural processor development is revived because the fabrication technology and

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 665–672, 2005.

CAD tool advance allow the implementation of a complex system on a chip in relatively low cost and several commercial products are available. However their application range is still limited. Data intensive operators in neural networks and image processing need high computing power and have a great potential for SIMD parallel processing. SIMD architectures have been adopted effectively on the applications of image processing, matrix operations, partial differential equations, artificial neural networks, multimedia processing, etc. Previously, SIMD architecture was realized in the form of massively parallel computer system starting from the first SIMD machine project, ILLIAC IV. They usually consisted of high performance host computer and data parallel unit including few hundreds or thousands of simple processing elements, memory system, and a global array control unit [2,3]. To overcome the limits of the previous SIMD processors, this paper proposes a flexible SIMD Processor with distributed memory taking full advantage of the application specific instruction set and hardware resources; Address Modifier (AM), Non-linear Functional Unit (NFU), ring and global buses, and multi-chip expansion. These features are optimally customized for achieving high computational efficiency in data intensive applications while providing flexibility.

2 Processor Architecture

The proposed processor employs an 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 1. 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. Global Register File (GRF) 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. 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, an l6-bit logical arithmetic unit (LALU) is embedded for basic logical operations (AND, OR, XOR, and NOT). For MAC operation, the result of multiplier is bypassed to adder. 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 weights, coefficients, image, and other data according to the applications. Followings are key features of the proposed processor.

2.1 Address Modifier (AM)

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.

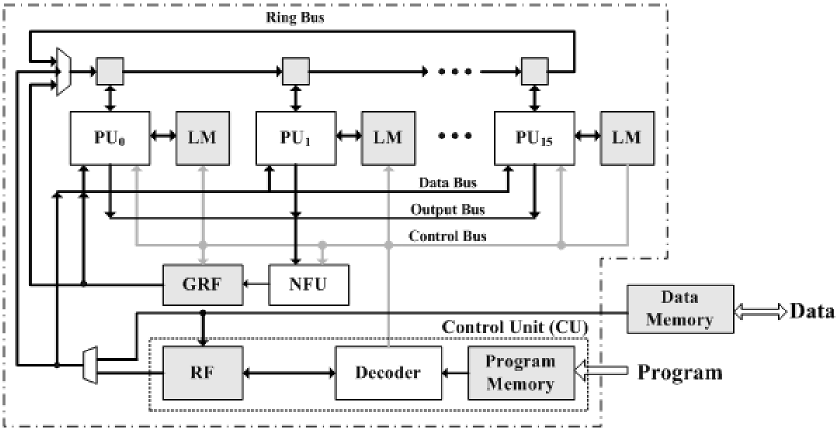
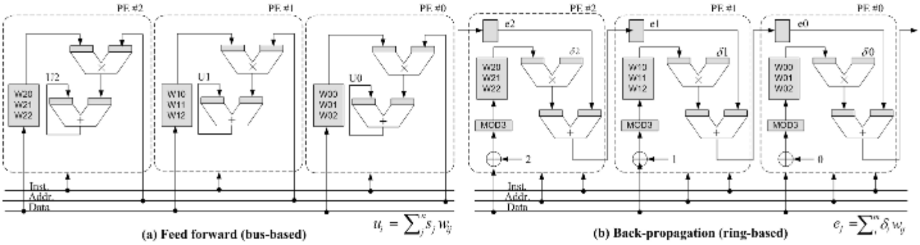


Fig. 1. Block diagram of the proposed architecture

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. Figure 2 shows an operational model of how an AM works on multilayer perception (MLP) with back-propagation (BP). Here, a row of the forward weight matrix is allocated to each PU. The first is feed-forward (FF) operation, in which the network computes the equation, $u_i = \sum_j^b s_j \omega_y$. The second is error BP that computes the equation,$e_j = \sum_i^m \delta_i \omega_y$. From these two equations, the weights distributed over LMs should be accessed in two different modes; the row-wise for the (FF)and the column-wise for the error BP. In the proposed processor, three mechanisms, ring, bus, and AM, are used for effective memory access for BP. For the process of FF operation, the address is broadcast to all PUs simultaneously through bus as shown on Figure 2 (a) since the weights are stored in local memories in row order. In error BP 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 as shown in Figure 2 (b). Therefore, the proposed architecture enables both row and column wise memory access without many overheads.

2.2 Multi-chip Expansion

The expandability of the proposed processor is essential because most scientific computations require large sizes of parallel processing. The multi-chip expansion through a register ring is used for increasing the network size. This is called as multi-chip ring shift operation mode which is decided by the flag attached to instructions. In this mode, the ring ready register is set to '1' and the program is stalled. After all the ring ready registers in chips are set (or the signal,*ext_shift_en*



(a) Feed forward (bus-based)

(b) Back-propagation (ring-based)

Fig. 2. Row and column mode memory access with AM

becomes '1'), the chip-to-chip ring shift operation is performed forming a larger ring across all the chips. This simplifies the chip-to-chip data transfer in multi-chips expansion. The contents of program memory of each chip are duplicated, and share the external data memory for multi-chip SIMD operation.

2.3 Instruction Set

The function of the proposed processor is programmed by means of 24 customized instructions for target applications. They include instructions for memory access, data transfers, arithmetic operations, and flow controls. Table 1 shows the instruction list including several special purpose instructions such as *BR, WLD*, etc. *BR* is used for broadcasting data through ring or bus. The selection is made by the flag bit appended to *BR* instruction. *WLD* loads data from an LM of its neighbor PUs or its own PU to RF, at this time, the AM can be selected to operate on column-wise or row-wise memory access.

3 Character Recognition System

The functionality and the performance of the proposed processor are verified with the character recognition application based on ANNs including image processing. Figure 3 shows the overall architecture of the proposed character recognition system. Generally, the character recognition application is separated into three phases [6]. The first phase is the image pre-processing using translation, dilation, rotations, thinning, and so on to bring a character to a standardized form. The second phase is the feature extraction that corresponds to linear or non-linear filtering. The third phase is the classification based on features that are obtained in the second phase. If the recognition system requires learning or adaptation, then additional learning or training stage is required.

3.1 Preprocessing and Feature Extraction

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 [7]. The *thinning* skeletonizes an input image

Table 1. Instruction Set

Instruction	**Description**	
NPLD CRn,@(disp:21),D	CRn and (RR or DBR) ←DM@(disp:21+CR0)	Load a data to data bus or ring bus from external memory (D=0: ring bus, D=1: data bus)
NPST CRn,	DM@(disp:13+CR0) @(disp:13),S	Store a data in an *OR* into external memory ← OR@CRn
NPMV Rn,Rm,P,S	Rn← Rm	Move a data in a register *Rn* to a register *Rm*
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*
NPADDim Rj, ♯(imm:16)C S	Rj ←Rj+ ♯(imm:16)	Add *Rn* with an immediate value (C=0: add in PU, C=1: add in CU)
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)

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 the *image filtering* certain lines may be broken. These broken lines have to be reconnected by the *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 BP learning.

3.2 Multi-layer Perception (MLP)

MLPs are well-established multipurpose classifying algorithm, and they are frequently employed in recognition systems. On the proposed system, the MLP consists of three layers; input, output, and one hidden layer. Each layer consists of 16 nodes. To improve recognition performance, the input character sets are grouped into one of four sub-nets according to the number of strokes in the character. Figure 4 shows the general structure of MLP with memory access

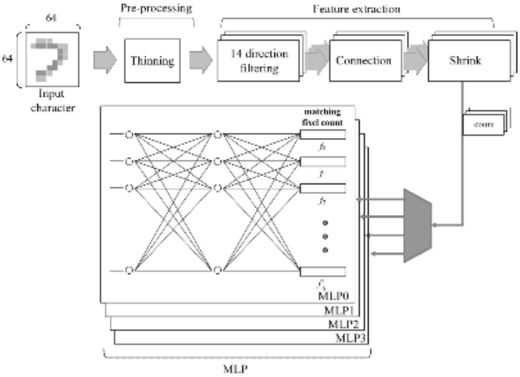


Fig. 3. Pattern recognition application using the proposed processor

mode on FF and BP stages. Each PU is assigned for one neuron on a layer. Therefore, one PU holds two weight sets, one for the first layer and the other for second layer. The FF path is processed with one input element at a time. The first input element z_1 is broadcasted through the data bus and all PUs compute the corresponding synaptic weight using v_{ji} stored in each PU, and then the second input element z_2 is broadcasted through the data bus. The same operations are repeated for all input elements and the result of synaptic weight for every input is accumulated at AR in each PU. This is the process between the input layer and the hidden layer, which can be express as the equation, $net_j = \sum_{j=0}^{j-1} v_{ji} z_i$. After that net_j is moved to NFU through output register (OR) to calculate the output value $y_i = f_i(net_j)$ for the hidden layer. At this time, y_i is also stored at global memory since it is going to be read again on BP stage. The same sequence of computations for the second layer is repeated to calculate the o_k in output neuron using the input y_i. This process can be expressed by the equations $net_k = \sum_{k=0}^{k-1} \omega_{kj} y_j$ and $o_k = f_k(net_k)$. In the BP stage, the produced output o_k is compared with the desired output d_k and an error value $\delta_{ok} = (d_k - o_k) f'_k(net_k) = (d_k - o_k) o_k (1 - o_k)$ is propagated backward to update weight values. The process is expressed as following equations; $\delta_{ok} = (d_k - o_k) f'_k(net_k)$ and $\omega_{kj} = \omega_{kl} + \eta \delta_{ok} y_i$ between the output layer and the hidden layer; $\delta_{yj} = (\sum_{k=0}^{k-1} \delta_{ok} \omega_{kj}) f'_k(net_k)$ and $v_{ji} = v_{ji} + \eta \delta_{yi} z_i$ between the hidden layer and the input layer. Finally, the weight is updated using δ_{yj}. These operations can be summarized as follows. First, the weight is expressed in the form of two dimensional matrix and stored in local memory in row order. Second, the input values are broadcasted to all PUs through the bus on the FF stage. Therefore, all PUs read weights at the same memory location and executes MAC operation. Third, on BP stage the AM modifies the memory address and the weights are read and calculated. And the desired values are broadcasted through the ring. This allows the proposed processor to operate on both row and column mode memory access without overhead. Another feature is that the calculation of non-linear function. The non-linear function requires

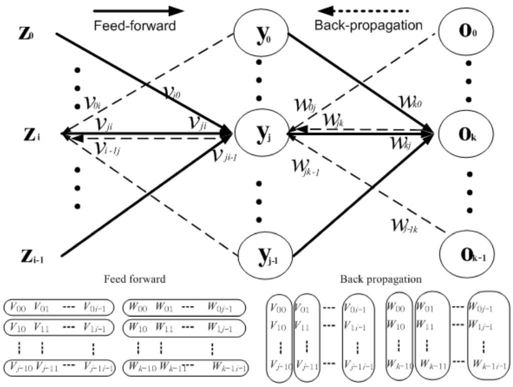


Fig. 4. MLP with back-propagation learning

complex computation or large look-up table. Implementation of such block in each PU significantly increases the hardware comp1exity. Therefore, only one NFU is implemented as a look-up table. In this case, the computation of nonlinear function win may be a bottleneck of the over all performance. However, the proposed architecture allows effective bus management which eliminated the bottleneck to use NFU. Furthermore, the processing time also can be reduced by storing the output values between each layer in global memory because they are expected to be fetched again on the BP stage.

4 Implementation Result and Conclusion

Figure 5 shows the physical layout of the prototype system board using the proposed processor architecture embodied in FPGA chip. The operating clock is 55MHz and operating voltage is 3.3V. The overall size is utilizing 19,914 logic elements equivalent to 320,000 gate level. CU is implemented with 32-bit × 4K-word 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 1,770 iterations on incremental learning mode. 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 14 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 0.18-micron 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. This paper proposed a high performance acceleration processor optimized for scientific computations such as image processing and ANNs. Highly customized 24 instructions were devised to improve the performance and the programmability of the processor on the target applications. From the architectural point of view, the characteristic of the proposed processor is SIMD with 16 PUs and special hardware resources such as NFU, a ring, and buses. Each PU includes arithmetic ALU (32-bit adder/subtracter/multiplier) and logical ALU (32-bit bitwise operators: AND,OR, XOR). The proposed architecture is suitable for applications that require heavy memory access relatively low computational complexity. Furthermore, the AM in each PU enables the proposed processor to have the ability to operate on column wise and row wise memory access, which can be exploited by many linear algebra applications.

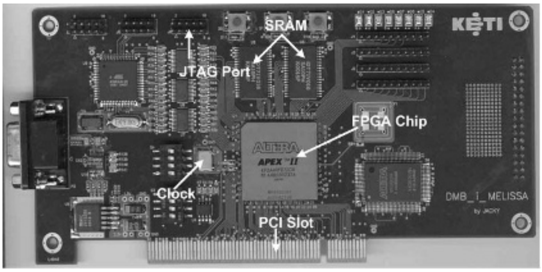


Fig. 5. A Prototype system board

References

1. Shiva S.G.: 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. Hicklin J., Demuth H.: Modeling Neural Networks on the MPP. Proc. 2nd Symposium on the Frontiers of Massively Parallel Computation. (1988) 39–42
4. 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
5. 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
6. Salembier P., Brigger P., Casas J. R., Pardas M.: Morphofogical Operators for Image Ellld Video Compression. IEEE Trans Image Process. **5** (1996) 881–898
7. Yentis R., Zaghloul M. E.: VLSl Implementation of Focally Connected Neural Networks for Solving Partial Differential Equations. IEEE Trans. Circnits Syst. I, Fundaln. Theory Appl. **43** (1996) 687–690

MRF-MBNN: A Novel Neural Network Architecture for Image Processing

Nian Cai[1], Jie Yang[1], Kuanghu Hu[2], and Haitao Xiong[2]

[1] Institute of Image Processing and Pattern Recognition, Shanghai Jiaotong University
1954 Huashan Road, Shanghai 200030, China
{cainian,jieyang}@sjtu.edu.cn
[2] Lab of Image Analysis and Pattern Recognition, Institute of Biophysics
Chinese Academy of Sciences, 15 Datun Road, Chaoyang District, Beijing 100101, China
{hukh,xionghaitao}@sun5.ibp.ac.cn

Abstract. Contextual information and a priori knowledge play important roles in image segmentation based on neural networks. This paper proposed a method for including contextual information in a model-based neural network (MBNN) that has the advantage of combining a priori knowledge. This is achieved by including Markov random field (MRF) into the MBNN and this novel neural network is termed as MRF-MBNN. Then the proposed method is applied to segmenting the images. Experimental results indicate the MRF-MBNN is superior to the MBNN in image segmentation. This study is a successful attempt of incorporating contextual information and a prior knowledge into neural networks to segment images.

1 Introduction

Although neural networks have been extensively studied in the field of image segmentation, two challenging problems should be further studied, which are the excellent incorporation of contextual information and the inclusion of a priori knowledge [1].

The main features of the model-based neural network (MBNN), over traditional neural networks, are that inclusion of (global) a priori knowledge in the network, combination of a priori knowledge and adaptive learning, and exclusion of combinatorial explosion that is inherent in other existing methods of modeling intellect and, in particular, performing automatic recognition [2], [3], [4]. For its features, especially the advantage of combining a priori knowledge, the MBNN was attempted to segment the images [5]. The contextual information, however, was not incorporated in the literature.

Markov random field (MRF) is a very popular method for image modeling and plays an important role in image analysis. It has the advantages of the ability to catch contextual information and tractable computation [6], [7], [8]. Therefore, in this paper, we introduce MRF into the MBNN to segment the images and use the term MRF-MBNN for this novel network. To decrease the computation burden, we employ the technique of preassigning a class number [5]. The experimental results show a significant improvement over the MBNN. We conclude that the reason is that contextual information, that is, neighborhood information, is incorporated in the MRF-MBNN, which is not included in the MBNN.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 673–678, 2005.

2 MBNN

The MBNN, in this paper, is based on the maximum likelihood adaptive neural system (MLANS) proposed by Perlovsky [9]. It has all the available information as its input, including the observed values $\mathbf{X}_n$, a priori knowledge and environmental interrogation feedback, if available. The output contains the estimated parameters of all classes defined as follows:

$$N_{km} = \sum_{n=1}^{N} W_{nkm} \ . \tag{1}$$

$$\mathbf{M}_{km} = \sum_{n=1}^{N} \frac{W_{nkm}\mathbf{X}_n}{N_{km}} \ . \tag{2}$$

$$\mathbf{C}_{km} = \sum_{n=1}^{N} \frac{W_{nkm}\left(\mathbf{X}_n - \mathbf{M}_{km}\right)^T \left(\mathbf{X}_n - \mathbf{M}_{km}\right)}{N_{km}} \ . \tag{3}$$

where W_{nkm} is the weight of the network, N_{km} is the estimation of object numbers of each type, $\mathbf{M}_{km}$ is the estimation of mean vector of each type, and $\mathbf{C}_{km}$ is the covariance matrix of each type; k is the class of objects, m is the type of the objects within the class.

If all available information is determined by probability terms, the weights are defined as posteriori Bayesian probabilities as follows:

$$W_{nkm} = P\left(k, m \middle| n\right) = \mathrm{pdf}\,(\mathbf{X}_n | k, m) \Big/ \sum_{k^{'}, m'} \mathrm{pdf}\,(\mathbf{X}_n | k^{'}, m') \ . \tag{4}$$

where $\mathrm{pdf}\,(\mathbf{X}_n \mid k, m)$ is a probability density function (pdf).

A Gaussian mixture model is used in this paper, thus the total pdf for all observations $\{\mathbf{X}_{\mathrm{n}}, n\text{=}1,...,N\}$ is a product of individual pdf($\mathbf{X}_n$).

$$\mathrm{pdf}\left\{\mathbf{X}_n, n = 1, \ldots, N\right\} = \prod_{n=1}^{N} \mathrm{pdf}\,(\mathbf{X}_n) \ . \tag{5}$$

$$\mathrm{pdf}\,(\mathbf{X}_n) = \sum_{k=1}^{K} \sum_{m=1}^{M} \mathrm{pdf}\,(\mathbf{X}_n | k, m) \ . \tag{6}$$

$$\mathrm{pdf}\,(\mathbf{X}_n | k, m) = (2\pi)^{-d/2} (\det \mathbf{C}_{km})^{-1/2} \exp(-\tfrac{1}{2} \mathbf{D}_{nkm} \mathbf{C}_{km}^{-1} \mathbf{D}_{nkm}^{T}) \ . \tag{7}$$

$$\mathbf{D}_{nkm} = \mathbf{X}_n - \mathbf{M}_{km} \ . \tag{8}$$

3 MRF-MBNN

Because MRF models express global statistics in terms of the local neighborhood potentials, most approaches have used MRF to accurately model the unknown images. And most of these are distinguished by the choice of potential function that assigns cost to differences between neighboring pixels.

Here we model the image X by the Gaussian MRF (GMRF) with a symmetry neighboring η, then X can be written in terms of a non-causal autoregressive (AR) representation with correlation coefficients θ [10],

$$X_s = \sum_{r \in \eta_s} \theta_r X_{s+r} + e_s \ . \tag{10}$$

$$B_\theta X = e \ . \tag{11}$$

where B_θ is a $M \times N$ matrix (M, N is the width and height of the image X respectively), and e is a zero mean Gaussian noise process with autocorrelation given by

$$E[e_s e_{s+r}] = \begin{cases} \sigma^2 & if \quad r = (0,0) \\ -\theta_r \sigma^2 & if \quad r \in \eta_s \\ 0 & \quad otherwise \end{cases} \cdot \tag{12}$$

where σ is a parameter that controls scale or variation in X.

It is possible, without loss of generality to halve the number of correlation parameters by making $\theta_r = \theta_{-r}$ $\forall r \in \eta_s$. In addition, the parameter set θ should satisfy $1 - \theta^T \phi_s > 0 \ \forall s \in \Omega$, and ϕ_s is a vector whose length is equal to the number of neighboring pixels η_s, which is defined by

$$\cos\left[\begin{pmatrix} \frac{2\pi s_1}{M} & \frac{2\pi s_2}{M} \end{pmatrix}\begin{pmatrix} r_1 \\ r_2 \end{pmatrix}\right] \quad r \in \eta_s \ . \tag{13}$$

This condition can guarantee the positive definiteness of covariance matrix of X. If the covariance matrix of e is $\sigma^2 B_\theta$, and that of X is $\Sigma = \sigma^2 B_\theta^{-1}$ [10]. Then the joint probability density functions of X is

$$p(X = x) = \frac{\sqrt{\det B_\theta}}{(2\pi\sigma^2)^{MN/2}} \exp\left\{-\frac{1}{2\sigma^2} x^T B_\theta x\right\} \ . \tag{14}$$

Also the conditional distribution for a single pixel may be written [10],

$$p(x_s \mid x_t, \forall t \neq s, t \in \Omega) = p(x_s \mid x_{s+r}, r \in \eta_s)$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{[x_s - \sum_{r \in \eta_s} \theta_r x_{s+r}]^2}{2\sigma^2}\right\} \cdot \tag{15}$$

This is a variation of the GMRF. Now we make an attempt to introduce GMRF into the MBNN. If we assume that the conditional distribution for every pixel belonging to some class *k* in the image satisfies the following forms:

$$p(x_s \mid x_t, k) = \frac{1}{\sqrt{2\pi\sigma_k{}^2}} \exp\left\{ -\frac{\left[(x_s - \mu_k) - \sum_{r\in\eta_s} \theta_r (x_{s+r} - \mu_k) \right]^2}{2\sigma_k{}^2} \right\} . \tag{16}$$

where μ_k is the mean parameter.

As we know, the weights W_{nk} in the MBNN are defined as follows (here we take account of the class *k*, not the type *m*):

$$W_{nk} = p(k \mid X_n) = \frac{p(X_n \mid k)}{\sum_{k'} p(X_n \mid k')} . \tag{17}$$

Thus when MRF is incorporated in the MBNN, the weights should be modified,

$$\begin{aligned} W_{nk} &= p(k \mid X_n, X_t) \\ &= \frac{p(X_n, X_t \mid k)}{\sum_k p(X_n, X_t \mid k)} = \frac{p(X_t \mid k) p(X_n \mid X_t, k)}{\sum_k p(X_t \mid k) p(X_n \mid X_t, k)} . \end{aligned} \tag{18}$$

$$p(X_n \mid k) = \sum_t p(X_n \mid X_t, k) p(X_t \mid k) . \tag{19}$$

these equations above are the basis for the design of the novel neural network named as MRF-MBNN.

To estimate the correlation coefficients θ, we adopt the modified version of the EM algorithm [11], which is an iterative procedure for approximating maximum-likelihood estimates. At each iteration, two steps are performed: the expectation step and the maximization step. Because iterations also exist in the MBNN, this modified version of EM algorithm can be applied to the estimation of MBNN parameters.

4 Results and Discussions

To decrease the computational burden, in all the experiments, the technique of preassigning a sufficiently large class number k_0 is used to adaptively determine the real class number *k*.

The Lena image is presented for segmentation (Fig. 1(a)). First of all, we employ the MBNN to segment the image. After 5 iterations, the MBNN convergences and the

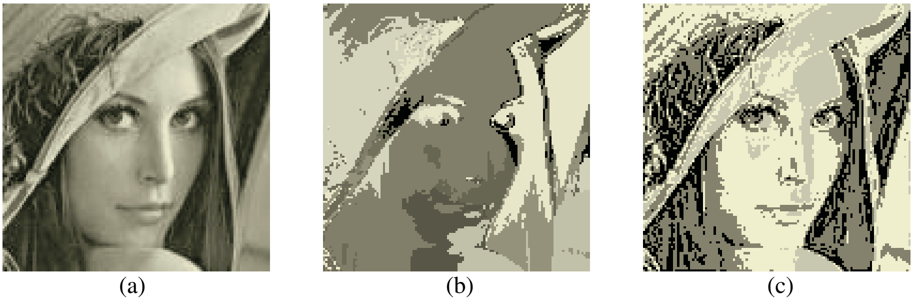


Fig. 1. (a) original image; (b) segmentation results by using the MBNN. The number of iterations is 5. k_0=40, k=20; (c) segmentation results by using the MRF-MBNN. The number of iterations is 3. k_0=40, k=4

segmentation results are shown in Fig. 1(b). Obviously the segmentation performance is poor and many details of the images are not well segmented. Does the selection of the preassigned class number k_0 or the number of iterations correspond to the poor performance? We repeated all the experiments after changing the preassigned class number and increasing the number of iterations. However, the results indicated our efforts are in vain and were almost the same as the previous ones (data not shown). Since neighborhood pixel values of the image are highly correlated, we consider that the fact that contextual information is not incorporated in the MBNN for segmentation maybe results in the poor performance.

Then we apply the MRF-MBNN proposed in this study, which well combines the MBNN and MRF and solves the problem of no contextual information existing in the MBNN, to segment the images. After three iterations, the network is in convergence and the results are shown in Fig. 1(c). Obviously, the results obtained by using the MRF-MBNN are superior to the previous ones obtained by using the MBNN. We consider that the proposed method takes account of the total interactions between all the pixels and their neighboring pixels, which results in better segmentation performance. The fact indicates that incorporating contextual information through MRF actually improves the performance of segmentation.

5 Conclusions

In this paper, we combine the MBNN with MRF and propose a new neural network termed MRF-MBNN. It well incorporates contextual information and a priori knowledge into neural networks, which exist in traditional neural networks. To test its performance, the segmentation experiments by using the MBNN are carried out as control experiments. The results indicate that the MRF-MBNN is actually superior to the MBNN and gives a good performance in segmentation. This study provides a novel approach to successfully incorporate contextual information and a prior knowledge into neural networks to segment images.

References

1. Egmont-Petersen, M., De Ridder, D., Handels, H.: Image Processing with Neural Networks – A Review. Pattern Recognition, **35** (2002) 2279-2301
2. Perlovsky, L.I., McManus, M.M.: Maximum Likelihood Neural Networks for Sensor Fusion and Adaptive Classification. Neural Networks, **4** (1991) 89-102
3. Perlovsky, L.I., Schoendorf, W.H., Burdick, B.J., Type, D.M.: Model-based Neural Network for Target Detection in SAR Image. IEEE Trans. Image Processing, **6** (1997) 203-215
4. Caelli, T.M., McG.Squire, D., Wild, T.P.J.: Model-based Neural Networks. Neural Networks, **6** (1993) 613-625
5. Cai, N., Hu, K.H., Xiong, H.T., Li, S.Y., Su, W.F., Zhu, F.S.: Image Segmentation of G Bands of Triticum Monococcum Chromosomes Based on the Model-based Neural Network. Pattern Recognition Letter, **25** (2004) 319-329
6. Besag, J.: Spatial Interaction and Statistical Analysis of Lattice Systems. Acad. R. Stat. Soc. B, **36** (1974) 721-741
7. Geman, S., Geman, D.: Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of Images. IEEE Trans. Pattern Anal. Machine Intell, PAMI-6 (1984) 721-741
8. Saquib, S.S., Bouman, C.A., Sauer, K.: ML Parameter Estimation for Markov Random Fields with Applications to Bayesian Tomography. IEEE Trans. Image Processing, **7** (1998) 1029-1044
9. Perlovsky, L.I.: Multiple Sensor Fusion and Neural Networks. DARPA Neural Network Study, MIT/Lincoln Lab, Lexington, MA (1987)
10. Kashyap, R.L.: Analysis and Synthesis of Image Patterns by Spatial Interaction Models. In: Kanal, L.N., Rosenfeld, A. (eds.): Progress in Pattern Recognition. North-Holland, Amsterdam (1981)
11. Comer, M.L., Delp, E.J.: Segmentation of Textured Images Using a Multiresolution Gaussian Autoregressive Model. IEEE Trans. Image Processing, **8** (1999) 408-420

Using LM Artificial Neural Networks and η-Closest-Pixels for Impulsive Noise Suppression from Highly Corrupted Images

Pınar Çivicioğlu

Erciyes University, Civil Aviation School
Department of Aircraft Electrics and Electronics, 38039, Kayseri, Turkey
civici@erciyes.edu.tr

Abstract. In this paper, a new filter, $\eta - LM$, which is based on Levenberg-Marquardt Artificial Neural Networks, is proposed for the impulsive noise suppression from highly distorted images. The $\eta - LM$ uses *Anderson-Darling goodness-of-fit test* in order to find corrupted pixels more accurately. The extensive simulation results show that the proposed filter achieves a superior performance to the other filters mentioned in this paper in the cases of being effective in *detail preservation* and *noise suppression*, especially when the noise density is very high.

1 Introduction

Images are often degraded by Impulsive Noise (IN) because of the errors caused by noisy sensors or transmission channels, thus suppression of IN is one of the most important issues in *digital TV broadcasting*, *image processing* and *real-time video restoration* systems. In image denoising, a compromise has to be achieved between noise reduction and preserving significant image details [1]-[9]. In the last years, nonlinear approaches based on artificial neural networks [3],[5],[10] have emerged as attractive alternatives to classical noise suppression techniques due to their advanced features and adaptive functionality. The images corrupted by IN typically have nonstationary statistical characteristics and are formed through a nonlinear system process. Therefore, *artificial neural networks* (ANN) based *nonlinear* approaches are very effective for noise cancellation because of their ability to approximate complex nonlinear functions. An important advantage of the ANN paradigm is knowledge representation and its ability of handling imprecise and inconsistent *real-world data*. If sufficient data can be obtained, an ANN learns from examples [10]. Therefore, ANN can play a *very important role* in information processing [6]-[8].

In this paper, a novel approach is presented for the restoration of IN-corrupted images by the application of *Levenberg-Marquardt*-ANN (LM-ANN) to nonlinear filtering. The task of impulse detection is a simple classification problem and in this paper, the *Anderson-Darling Goodness-of-Fit test value* (AD) [11] has been used to check each pixel in order to detect whether it is *distorted* or *not*. Then, the proposed nonlinear filtering scheme is achieved for the distorted

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 679–684, 2005.

pixels, while the noise-free pixels are left unaltered. In order to test the success of the proposed filter, η**-LM**, the performance of η**-LM** is compared with the performances obtained by using Histogram Based Fuzzy Filter (HBF) [7], Yüksel's Fuzzy Filter (YÜKSEL) [6], Russo's *if-then-else* fuzzy reasoning filter (RUSSO) [8], and Selective Median Filter (SEL) [9].

2 Statistical Analysis of Impulsive Behavior of Pixels

Statistical tests are beneficial in the analysis of impulsive behavior of pixels [1]-[5],[12], therefore, in this paper, impulsive behavior of the image pixels has been investigated with the use of statistical tests. In this paper *ten* statistical distribution models (i.e., *Binomial, Rayleigh, Gamma, Possion, Chi-squared, Normal, Integer uniform, Cumulative, Logaritmic normal, and Fisher-Tippett distribution* [10],[11]) have been examined and all of them have been implemented in Matlab [10], in order to find out the best-fitted statistical distribution model for each of the intensity levels. This statistical analysis has revealed that the well-known *Fisher-Tippett* distribution (FTD) [11] appears to be the best statistical distribution model for the sample of intensity data, which are derived from [32x32] pixels sized unoverlapping blocks (bins) [1]-[5]. In this paper, AD has been employed in order to statistically analyze the residuals of FTD. The FTD distribution has been used to find out the pixels, which are suspected to be *corrupted* intensity levels in distorted image. Since the corruption probability of each pixel is numerically equal in real images, noisy pixels *position uniform* [1]-[5] throughout the image surface. In this paper, AD [11] has been used in order to detect whether the intensity levels *position uniform* throughout the image surface or not. If one intensity level has been detected as *positioning uniform*, then the pixels possessing this intensity level are considered as *corrupted pixels* [1]-[5]. In order to statistically analyze impulsive behavior of the intensity levels, the image surface is divided into [32x32] pixels sized unoverlapping subimages. Extensive simulation results expose that [32x32] pixels sized bins are ideal for AD. For each intensity level, a *numerical set* denoting the number of the pixels, which possess this intensity level within the subimages, has been determined. This *numerical set* has been used for investigating the AD of an intensity level. It is observed empirically that the intensity levels, whose AD is smaller than the *threshold* $\mathbf{2.00 \pm 0.05}$ belong to the *corrupted pixels.*

3 Levenberg-Marquardt Training Algorithm

An ANN is a nonlinear information processing tool that is inspired by the way of biological nervous systems. ANNs have seen an explosion of interest over the last few years, and are being successfully applied across an extraordinary range of problem domains, in areas as diverse as *finance*, *medicine*, *engineering*, *geology*, *image processing*, *pattern recognition* and *optics*. There are various types of neural network applications available in the literature for image restoration [3],[5],[10]. **η-LM** updates weight and bias values according to *well-known* Levenberg-Marquardt (LM) optimization [10].

The LM algorithm is usually defined as a least-squares estimation algorithm that is based on the maximum neighborhood idea. Let E(w) be an objective error function which has m individual error terms $e_i(w)$. Then E(w) is defined as follows:

$$E(w) = \sum_{i=1}^{m} e_i^2\,(w) = \|f\,(w)\| \tag{1}$$

where $e_i^2(w) = (y_{di} - y_i)^2$, y_{di} is the desired value of output neuron i and y_i is the output of that neuron. The function $f(\cdot)$ and its Jacobian J are assumed to be known at point w. The aim of the LM algorithm is to compute the weight vector w so that the value of $E(w)$ is minimum. By using the previous weight vector w_k, a new weight vector w_{k+1} can be obtained as follows:

$$w_{k+1} = w_k + \delta w_k \tag{2}$$

where δw_k is defined as

$$\delta w_k = -(J_k^T f(w_k))(J_k^T J_k + \lambda I)^{-1} \tag{3}$$

where J_k, λ and I are the Jacobian of f evaluated at w_k, Marquardt parameter and the identity matrix, respectively [13]. The LM-ANN structure used in this paper is illustrated in Fig. 1, where x, y and g denote the spatial positions and the related gray value of the pixels, respectively.

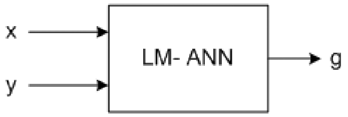


Fig. 1. The relation between the spatial positions of pixels, (x, y), LM-ANN and gray value, g at the image point (x, y)

4 Proposed Method

In this paper, the LM-ANN structures have been used as a Descriptive Surface Fitting (DSF) interpolant [1]-[6], in order to estimate acceptable intensity values for corrupted pixels.

Searching for 'η-Closest-Pixels (points)' is still an important problem in many fields of science and engineering [14],[15]. In this paper, the *Euclidean* distance has been used in order to find out the η-Closest-Uncorrupted-Pixels of a corrupted pixel. Readers interested in the details of 'η-Closest-Pixels (points) Search' may refer to some recently published excellent papers on this subject [14],[15]. The parameters of the LM-ANN structures, that are seen in Fig. 1, have been obtained by training the LM-ANN structures with 100 standard epochs. The training functions are sigmoidal and all the LM-ANN structures have 10 neurons in their *single* hidden layer.

The computational algorithm of the η-**LM** is given below step-by-step:

- The image coordinates, (x, y), of the corrupted pixels, which were determined by using AD as FTD are found.

For each corrupted pixel the following steps are performed:

- The spatial positions of the η-Closest-Uncorrupted-Pixels are found for the related corrupted pixel.
- The spatial coordinates and gray values of the η-Closest-Uncorrupted-Pixels are used in order to train an LM-ANN structure.
- The trained LM-ANN structure is used in order to make an estimation for the gray value of the corrupted pixel.
- The gray value of the corrupted pixel is updated with the value estimated in the previous step.

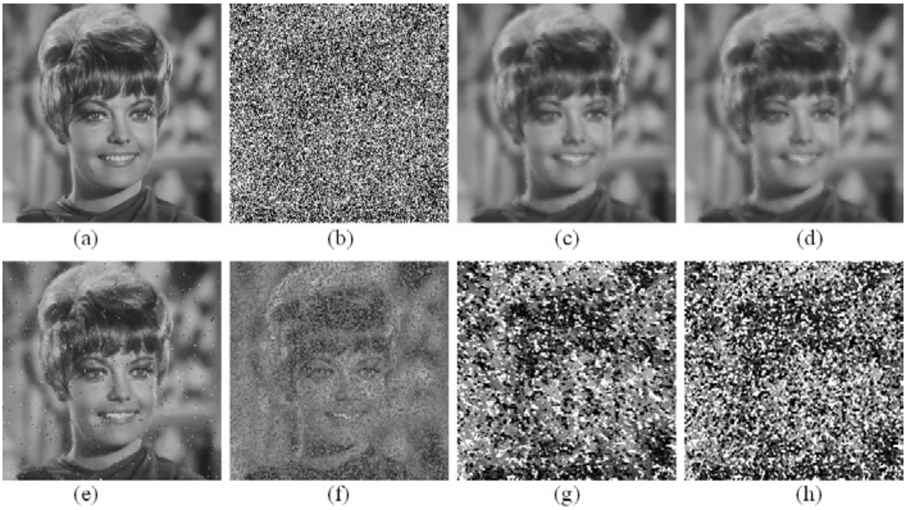


Fig. 2. The restored images of the Zelda Image for the Noise Density of 75%: (a) Original Image, (b) Corrupted Image at the noise density of 75%, (c) $\eta - LM$ (η=10) (proposed), (d) $\eta - LM$ (η=15) (proposed), (e) HBF, (f) YÜKSEL, (g) RUSSO, (h) SEL

5 Experiments

A number of experiments were realized in order to evaluate the performance of the proposed η-**LM** in comparison with the recently introduced and highly approved IN suppression filters [6],[7],[8],[9]. The experiments were carried out

Table 1. Comparison of the restoration performances of the mentioned methods in MSE for The Zelda test image

Method	Noise Density				
	15%	30%	45%	60%	75%
Corrupted Image	2706.60	5343.10	8052.90	10741.00	13442.00
η-LM (η=10) (proposed)	6.88	15.29	26.72	43.31	70.96
η-LM (η=15) (proposed)	8.51	19.12	32.67	51.81	85.46
HBF	6.03	13.40	25.40	73.10	133.23
YÜKSEL	33.73	75.61	179.61	380.78	719.09
RUSSO	28.70	151.87	665.67	2028.60	5084.00
SEL	44.65	261.17	1213.82	3655.36	7902.06

on the well-known test image; *The Zelda*, which is [256x256] pixels sized and 8 bits per pixel. The test image was corrupted by IN at various noise densities ranging from 15% to 75%. The restoration results of the test image for the noise density of 75% are illustrated in Fig. 2 where it is clearly seen that, the noise suppression and detail preservation are satisfactorily compromised by using the proposed η**-LM** even if the noise density is high (i.e. 75%). Restoration performances are quantitatively measured by the Mean-Squared-Error (MSE) [1]-[5]. The experiments have been conducted for both η=10 and η=15. Experimental studies realized on *real world* images revealed that $10 \leq \eta \leq 20$ supplies the best restoration results. It is obviously seen from Table 1 that η**-LM** provides a substantial improvement compared with the simulated filters, especially at the high noise densities. Robustness is one of the most important requirements of modern image enhancement filters and Table 1 indicates that the proposed η**-LM** provides robustness substantially across a wide variation of noise densities.

6 Conclusions

In this paper, a high performance filter is proposed for the suppression of IN, which yields satisfactory results with *detail preserving* while requiring a simple computational structure. The effectiveness of the proposed filter can easily be evaluated by appreciating Table 1 which demonstrates the restoration results of η**-LM** and the comparison filters for the test image degraded by IN, where noise density ranges from 15% to 75%. As can be seen from Table 1 and Fig. 2 that IN removal and detail preservation are best compromised by the proposed filter and η**-LM** provides visually satisfactory images, even if noise density is very high.

References

1. Çivicioğlu, P., Alçı, M: Impulsive Noise Ssuppression from Highly Distorted Images with Triangular Interpolants. AEU International Journal of Electronics and Communications, **58** (2004) 311–318

2. Çivicioğlu, P., Alçı, M: Edge Detection of Highly Distorted Images Suffering from Impulsive Noise. AEU International Journal of Electronics and Communications, **58** (2004) 413–419
3. Çivicioğlu, P., Alçı, M, Beşdok, E.: Using an Exact Radial Basis Function Artificial Neural Network for Impulsive Noise Ssuppression from Highly Distorted Image Databases. Lecture Notes in Artificial Intelligence, **3261** (2004) 383–391
4. Çivicioğlu, P., Alçı, M., Beşdok, E.: Impulsive Noise Suppression from Images with the Noise Exclusive Filter. EURASIP Journal on Applied Signal Processing, **2004** (16) (2004) 2434–2440
5. Beşdok, E., Çivicioğlu, P., Alçı, M.:Impulsive Noise Suppression from Highly Corrupted Images by Using Resilient Neural Networks. Lecture Notes in Artificial Intelligence, **3070** (2004) 670–675
6. Yüksel, M.E., Baştürk, A.: Efficient Removal of Impulse Noise from Highly Corrupted Digital Images by a Simple Neuro-fuzzy Operator. AEU International Journal of Electronics and Communications, **57** (2003) 214–219
7. Wang, J., Liu, W., Lin, L.: Histogram-based Fuzzy Filter for Image Restoration. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, **32** (2002) 230–238
8. Russo, F., Ramponi,G.: A Fuzzy Filter for Images Corrupted by Impulse Noise. IEEE Signal Processing Letters, **3** (1996) 168–170
9. Pok, G., Liu, J.C., Nair, A.: Selective Removal of Impulse Noise Based on Homogeneity Level Information. IEEE Transactions on Image Processing, **12** (2003) 85–92
10. MathWorks: Matlab v.6.2., Reference Guides, New York:The MathWorks, Inc., (2004)
11. Palisade: Bestfit, Probability Distribution Fitting for Microsoft Windows, Software, Version 4.5.2, (2002)
12. Brown, C.L., Zoubir, A.M.: Testing for Impulsive Behavior: A Bootstrap Approach. Digital Signal Processing, **11** (2001) 120–132
13. Hagan, M.T., Menhaj, M.B.: Training Feedforward Networks with the Marquardt Algorithm, IEEE Trans. Neural Networks, **5** (1994) 989–993
14. Agrell, E., Eriksson, T., Vardy, A., Zeger, K.: Closest Point Search in Lattices. IEEE Transactions On Information Theory, **48** (2002) 2201–2214
15. Henk, M.: Note on Shortest and Nearest Lattice Vectors. Information Processing Letters, **61** (1997) 183–188

Two Novel Image Filters Based on Canonical Piecewise Linear Networks

Xusheng Sun, Shuning Wang, and Yuehong Wang

Department of Automation, Tsinghua University, Beijing 100084, China
{sunxs03,wangyuehong03}@mails.tsinghua.edu.cn, swang@mail.tsinghua.edu.cn

Abstract. Although many filters have been proposed,image denoising is still worth further studying. In this paper, two novel image filters based on canonical piecewise linear networks are presented. They have the advantages of both linear filters and nonlinear filters. The former filter removes noises through the estimation of local structure, while the latter one accomplishes that by approximating the mapping from degraded images to clear images. They can remove noises effectively and preserve the details well. Finally, simulation results are shown to support their effectiveness.

1 Introduction

Image denoising is a basic approach in the field of image enhancement, and it is frequently used as an image preprocessing step. Many works have been done to deal with various noises. For instance, linear filters such as mean filters are effective against Gaussian noise, but they can not cope with impulse noise and will blur the edges. Median filters as typical nonlinear filters can remove salt and pepper noise easily, but they can not eliminate Gaussian noise and may destroy the details. Until now, the removal of mixed noises is still a difficult problem.

Neural networks play an important role in many aspects of image processing, mainly including image segmentation, classification and compression. Canonical piecewise linear networks[1],[2] are a special kind of neural networks that are simple and effective. They have been widely used in many fields, such as resistor networks synthesis[3]and signal filters design[4]. In fact, both mean filters and median filters are special forms of canonical piecewise linear networks [5].

In this paper, two novel filters based on these networks are proposed to cope with the mixed noises in images. Simulation results show that their effects of noise reduction are improved as compared with both mean filters and median filters.

2 Canonical Piecewise Linear Networks

Definition 1: The neural networks are called canonical piecewise linear networks[1]if they accord with the following form:

$$f(\mathbf{x}) = l(\mathbf{x}, \mathbf{a}_0) + \sum_{i=1}^{M} c_i |l(\mathbf{x}, \mathbf{a}_i)| \tag{1}$$

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 685–689, 2005.

where $\mathbf{x} = (x_1, \cdots, x_k, 1)$, $\mathbf{a}_i \in \mathbf{R}^{k+1}$, $c_i \in \{+1, -1\}$, $i = 0, 1, \cdots, M$ and $l(,)$ denotes the inner product of two vectors.

In fact, canonical piecewise linear networks can represent all piecewise linear functions in one dimension. They can also represent a large amount of piecewise linear functions in high dimensions and give a rough approximation to arbitrary continuous functions. These networks are actually a kind of one-hidden layer neural networks. As we know, for $\forall a, b \in \mathbf{R}$, $|a| = max\{0, 2a\} - a$, so we can rewrite the canonical piecewise linear networks in the following form:

$$f(\mathbf{x}) = l(\mathbf{x}, \mathbf{a}_0) + \sum_{i=1}^{M} c_i \; max\{0, l(\mathbf{x}, \mathbf{a}_i)\} \tag{2}$$

An effective algorithm called hinging hyperplanes algorithm is proposed in [6]to approximate continuous functions with the model in(2). So the same algorithm can be used to train canonical piecewise linear networks. In fact, it belongs to gradient based algorithms.

3 A Novel Image Filter Based on Canonical Piecewise Linear Networks

The two main objectives of image filters are preserving the edges and eliminating the noises. But there are always some constraints to achieve the above two goals simultaneously, as the noise point can not be distinguished easily from the edge point. For instance, if the window is larger, more noises are removed, meanwhile edges will be smoothed more heavily.

In this section, canonical piecewise linear networks are used to design a novel image filter. It is a special filter that has elements of both linear filters and nonlinear filters. Linear elements are located in linear functions, while nonlinear elements concentrate on maximal functions.

Piecewise linear structure is a familiar mode used in image approximation [7],[8]. Like mean filters, canonical piecewise linear networks are used to estimate local structure in the neighborhood of the current pixel.In some cases, the pixels in the neighborhood totally belong to the foreground or the background, while in other cases, one part is in the foreground and the other part is in the background.Considering that the edges in one window are always simple, we use only one neuron to construct the neural network here, namely $max\{l(\mathbf{x}, \mathbf{a}_1), l(\mathbf{x}, \mathbf{a}_2)\}$ or $min\{l(\mathbf{x}, \mathbf{a}_1), l(\mathbf{x}, \mathbf{a}_2)\}$.

Thus the neighborhood will be divided into two polygonal parts according to the intensities of the pixels. In each part, a linear function will be used to approximate the intensities through least mean square algorithm.In this way, the noises in each part will be removed separately and edges will also be more preserved.

This novel filter considers the position and intensity of every pixel at the same time. It may be more or less like the bilateral filter[10]. But the bilateral filter can not cope with the impulse noises effectively,since its nonlinear element is not apparent.

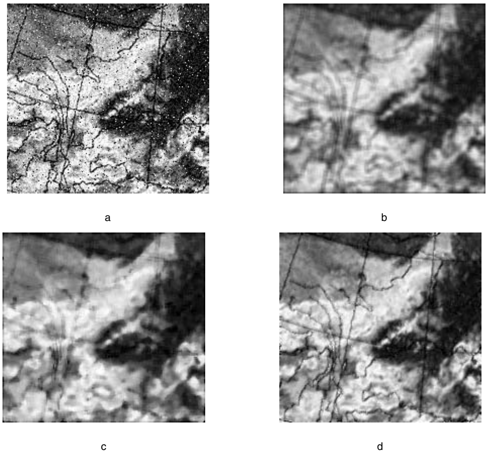


Fig. 1. The results of the first filter

4 A Novel Image Filter Combining Mean Filters and Median Filters

Since both mean filters and median filters are effective against some kind of noises but not to other kinds, so here we try to combine the advantages of these two filters to get better results. An algorithm[9] has been proposed to remove the mixed noises in images. Its main idea is to derive clear images through linear combination of the results of mean filters and median filters. However, the coefficients are not easy to compute, and it may be not reasonable to use simple linear combination sometimes.

In this section, we treat an image filter as a mapping from the noisy image to the clear image, and it can also be seen as the inverse mapping of the degraded process. Thus some appropriate approximators can be used to fit the mapping.Since the mapping is not assumed to be very complicated, canonical piecewise linear networks are available here.

In order to utilize the advantages of mean filters and median filters, here we construct the mapping from both the mean filtered image and the median filtered image to the clear image, which can be represented as:

$$f(\mathbf{x}) = l(\mathbf{x}, \mathbf{a}_0) + \sum_{i=1}^{M} c_i \; max\{0, l(\mathbf{x}, \mathbf{a}_i)\} \tag{3}$$

where $\mathbf{a}_i \in \mathbf{R}^3$, $i = 1, \cdots, M$, $\mathbf{x} = (x_1, x_2, 1)$, and x_1, x_2 are the results of the mean filter and the median filter respectively. In this way, the mapping from the degraded image to the original image is simplified.

Our filtering methodology will be stated as follows. First, we select a clear image similar to the given image in style as a reference, and the same noises as those in the given image are added to the reference image. Second, the mean filtered result and the median filtered result of the reference image are chosen as the inputs, meanwhile the reference image is chosen as the output. Then we can train canonical piecewise linear networks with them. Since both patterns and noises are similar between the given image and the reference image, the derived mapping can be used to estimate the unknown mapping from the degraded image to the original image in practice. Thus if we input the two filtered results of the given image, then the output of the network is the desired result. It is actually a combination of mean filters and median filters according to different proportions in different regions.

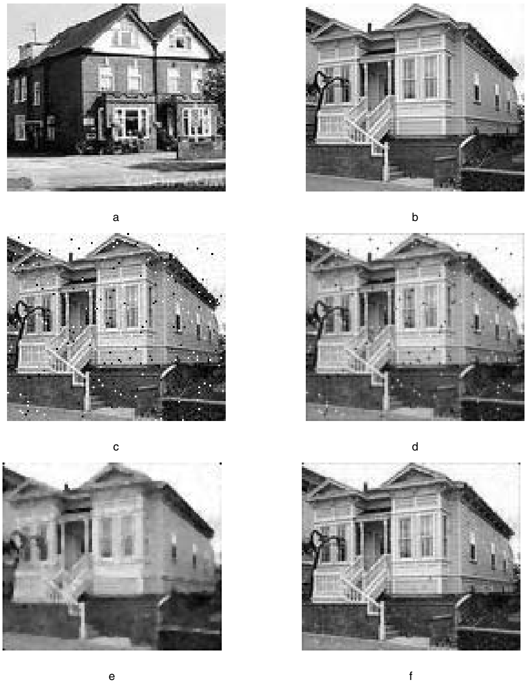


Fig. 2. The results of the second filter

5 Simulation Results

Simulation results are shown orderly to verify the effectiveness of the two novel filters. A part of climograph and two house images are used here orderly. Figure1 shows the results of the former filter. Figure1(a) is the degraded image derived from adding 0.004 Gaussian noise and 0.03 salt and pepper noise to the original image. The mean filtered result and the median filtered result of (a) are shown in (b) and (c), while the result of our filter is shown in (d). The widths of windows are all 5. The PSNR increases 6.55dB through our filter. Figure1(d) removes most noises and holds major details of (a).

Figure2 shows the results of the latter filter. Figure2(a) is the reference image, and (b) is the original image. Figure2(c) is the given image that is derived by adding 0.002 Gaussian noise and 0.02 salt and pepper noise to (b). The mean filtered result and the median filtered result of (c) are shown in (d) and (e) respectively, while the result of our filter is shown in (f). The widths of windows are all 5, and we use 3 neurons to fit the mapping here. The PSNR increases 7.15dB through our filter. It is obvious that (f) carries a better visual effect than (d) and (e).

6 Conclusions

In this paper, two novel image filters based on canonical piecewise linear networks have been proposed. They combine the advantages of both linear filters and nonlinear filters. Neural networks improve the effect of common filters, thus some tough noises can be suppressed effectively by the novel filters.

References

1. Chua, L.O., Kang, S.M.: Section-wise Piecewise-linear Functions: Canonical Representation, Properties, and Applications. IEEE Trans. Circuits Systems, **30** (1977) 125-140
2. Lin, J., Unbehauen, R.: Canonical Piece-wise Linear Networks. IEEE Trans. Neural Networks, **6** (1995) 43-50
3. Storace, M., Julian, P., Parodi, M.: Synthesis of Nonlinear Multiport Resistors: A Pwl Approach. IEEE Trans. Circuits Systems, **49** (2002) 1138-1149
4. Lin, J.N., Unbehauen, R.: Adaptive Nonlinear Digital Filter with Canonical Piecewise-linear Structure. IEEE Trans. Circuits Systems, **37** (1990) 347-353
5. Li, W., Lin, J.N., Unbehauen, R.: Unification of Order-statistics Based Filters to Piecewise-linear Filters. IEEE Trans. Circuits Systems, **46** (1999) 1397-1403
6. Breiman, L.: Hinging Hyperplanes for Regression, Classification, and Function Approximation. IEEE Trans. Information Theory, **39** (1993) 999-1013
7. Action, S.T., Bovik, A.C.: Nonlinear Image Estimation Using Piecewise and Local Image Models. IEEE Trans. Image Processing, **7** (1998) 979-991
8. Julian, P., Dogaru, R., Chua. L.O.: A Piecewise-linear Simplicial Coupling Cell For CNN Gray-level Image Processing. IEEE Trans. Circuits Systems, **49** (2002) 904-913
9. Hsmza, A.B., Krim, H.: Image Denoising: A Nonlinear Robust Statistical Approach. IEEE Trans. Signal Processing, **49** (2001) 3045-3053
10. Elad, M.: On the Origin of the Bilateral Filter and Ways to Improve It. IEEE Trans. Image processing, **11** (2002) 1141-1151

A New Effective and Powerful Image Segmentation Method*

Yalin Miao[1], Xianglin Miao[2], Zhengzhong Bian[1], Kai Chen[2], and Gang Yu[1]

[1] School of Life Science and Technology, Xi'an Jiaotong University
Xi'an, Shaanxi 710049, China
myl@mailst.xjtu.edu.cn

[2] School of Electronic and Information Engineering, Xi'an Jiaotong University
Xi'an, Shaanxi 710049, China

Abstract. A new algorithm for image segmentation named radar algorithm is proposed in this paper. The proposed algorithm is based on SOM network model, the mathematical morphology and physical simulation. This new algorithm is simple and easy to use while with powerful functions. The time complexity of the radar algorithm is 90*O(n^3). Further, the proposed algorithm has better robustness. The experimental results show that the Radar algorithm is an effective and powerful image segmentation method. It has wide applications in image segmentation especially in the processing of images with discontinuous edges.

1 Introduction

In the last decades, with the developments of machine vision and image understanding, higher requirements are expected for the image processing techniques especially in the edge detecting and image segmentation. So some new theories and algorithms are proposed. However, the new algorithms often become more complex with unsatisfied effectiveness [1].

The Canny operator [2] is an outstanding algorithm for the edge detection, and Hough transform [3] is an effective curve fit algorithm. These two algorithms result in the maturation of traditional algorithms of image segmentation and edge detection [4]. Other methods which are similar to these two algorithms basically use the analytical geometry, that is, they try to find out the analytic geometry formulas of the curves, and then search out or calculate the parameters for these formulas according to the properties of the pixels in the image. So these algorithms are mainly based on detailed computation, and the computation process is often very complex, but the results are often not good. These algorithms mainly aim at being used for the regular geometrical figures. The disadvantages of these algorithms are that they do not use topological features of the image; they only use local features of pixels but not con-

* Supported by the National Natural Science Foundation of China under Grant NO.60271022 and NO.60271025.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 690–697, 2005.

sider the global properties of the image like locations of pixels in the whole image and relationships between pixels. So these algorithms have unavoidable limitations. For example, when using the Canny operator or other traditional edge detection algorithms to extract edges of a region, we may find discontinuous rifts on edges more or less because of the complexity of real objects. If edges and rifts are very complex, the Curve Fit algorithm such as the Hough transform may not create satisfied results. And the detection of regions based on edges is becoming more difficult.

In this paper we propose a new algorithm for image segmentation–Radar Algorithm. The Radar algorithm gets its name from the similar operating principle as a radar searching its targets when the Radar algorithm is used to process images. We divide this algorithm into Narrow-sensed Radar Algorithm and Broad-sensed Radar Algorithm. Based on SOM network mode [5], physical simulation, and the mathematical morphology, it can extract single enclosed region from the image by using the global information and fuzzy statistical characteristics of pixels and discontinuous edges.

2 The Principle of Radar Algorithm

2.1 Narrow-Sensed Radar Algorithm

If there is only one closed region in the image, we process the image as follows: firstly, for each pixel in the image, we take it as a center and scan other pixels around the center from 0 to 2π at a certain angle step. For each angle during scanning, we draw a radial from the central pixel along the direction of the current angle. When the radial meets a boundary pixel, it stops and records this position and the distance from this position to the central pixel. If the boundary pixel isn't found in this direction, the distance is set to ∞ and regarded as -1 in the program. After the scanning, a distance is obtained in each direction for the current central pixel. When there is a sharp change of the distance between the adjacent angles, we regard there is a rupture in the boundary at that position. Finally, we count out the number of angles where the distance is -1 and the number of angles where there is a sharp change for distances. When these angles are very close in position or have a large numbers, this pixel is regarded not to belong to the closed region. On the contrary, when these angles are very separated in positions or have a small numbers, this pixel is regarded to belong to the closed region.

If boundaries are distributed from the inside to the outside in their gradients, we can make scanning and searching with multiple steps. In each step, we find the outside-most boundary and clear it from recordings. We continue the searching process from outside to inside and find all boundaries. We name this method as Narrow-sensed Multiple Steps Radar Algorithm.

2.2 Board-Sensed Radar Algorithm

If there are more than one closed regions in the image, we can combine the narrow-sensed radar algorithm and SOM network model together to form broad-sensed radar algorithm. In the SOM model, each weight coefficient sequence $W_j = (W_{1j}, W_{2j}, \cdots, LW_{nj})$ can be regarded as an internal expression of the Neural Network. It is also the relevant mapping of the ordered sequence of $X_j = (X_{1j}, X_{2j}, \cdots, X_{nj})$. The SOM model can carry out the self-organizing function. The aim of the function is to let the neural network converge to a kind of expressional state through modulating the weighting coefficient W_{ij}. One neural cell in the expressional state is only sensible or matching to some specifically input model. In other words, the aim of the SOM is that the expressional state of the weighting coefficient of the neural cells can indirectly imitate the model of the input signal.

From Ref. [5], we can find that SOM can intelligently classify each kind of animals according to the relative relationships. Meanwhile, we can find a very important phenomenon that the segmented regions are always connected if the segmentation is carried out according to some kind of resemblance between the animals no matter what kind of methods are used. In other words, the relative species are distributed to the associated parts. This phenomenon can be used to carry out the region segmentation in image processing.

3 Radar Algorithm

3.1 Narrow-Sensed Radar Algorithm

The Narrow-sensed Radar Algorithm is expressed as following steps:

(1) Assign one array to every pixel, as shown in Fig. 1, and then apply the following operations.

(2) Record the distance from the central pixel to the nearest boundary pixel for each angle from 0o to 359o. If we have not found this kind of pixel till the boundary of the image, we set the distance for this angle be -1. We consider one pixel has boundary when more than three pixels of eight neighboring pixels are set as boundary pixels.

(3) Scan and record each angle to see whether there is sharp change of the distance between angles. If the distance change is more than 1/3 of the current distance, we consider there is a sharp change of the distance.

(4) Record the affected range of every sharp changed distance.

(5) Get the weighted summation of all factors that the distance to the boundary is -1, there are sharp changes of distances, and the affected range of the sharp changed distance is large. If this summation is within a certain range, the pixel is regarded being surrounded by boundaries. That is, this pixel is inside the region and we set the value of this pixel be 1, otherwise the value of this pixel is not changed.

0^0	1^0	···	358^0	359^0

Fig. 1. The array assigned to angles

3.2 Broad-Sensed Radar Algorithm

The main principle of Broad-sensed Radar Algorithm is to make use of the uncompleted boundary information based on statistics method to segment regions of interest in the image. In this algorithm, we firstly record a group of values for each pixel in the image. The content of values recorded for each pixel is shown in Fig. 2.

R	G	B	X	Y	coordinate of center of mass x	coordinate of center of mass y	coordinate sequences of boundary pixels
							…

Fig. 2. Content of the record for each pixel

In Fig. 2, the R, G and B are the color values of the pixel in the original image, x and y are the coordinates of the pixel. The coordinate of center of mass can be gotten using following steps. Firstly we take the current point as the center point and scan the pixels in the boundary image with a radial from the central point with a certain angle step from 0 to 2π . For every angle, we can find a pixel, which lies on the boundary. If this pixel does not exist, we set the image boundary pixel as the pixel on the region boundary. Afterward, all the boundary pixels are filtered through the low pass filter according to the distance between the center pixel and boundary pixels. Finally the complete boundary is formed, and this complete boundary is regarded as a curve object. Then we compute and record the coordinate of center of mass of this curve object. We use the center of mass as the center and with even angles from the center to divide the boundary. We record the coordinates of every divided boundary point and get the coordinate sequences of boundary pixels [6]. This process is presented in Fig. 3.

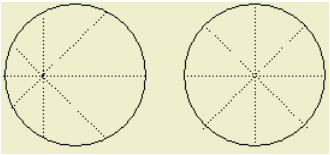

(a) (b)

Fig. 3. The boundary sequences: (a) The boundary sequence of the original point. (b) The boundary sequence of center of mass

Thus we finish the complete record for each pixel. We input these values into SOM and classify pixels. Through mapping the relative relationships of each pixel to SOM network of each region, and then we can segment regions based on the extracted features from regions.

4 Experiments

The image to be processed is a random shape. Fig. 4(a) is a calabash shape. The number of pixels is about 5200, which is changed to 3600 for the created circle after Hough Transform. The error rate of the Hough Transform is about 30.76%. Fig. 4(d) is a heart shape, the number of its pixels is about 4200, and the error rate of Hough Transform for the created result is about 35.48%. It is obvious that Hough Transform cannot work well for random shapes. On the other hand, Radar Algorithm does not reduce the efficiency because of the complexity of shapes, as show in Fig.4(c) and (f). So the remarkable advantages of Radar Algorithm can be gotten from these experiments.

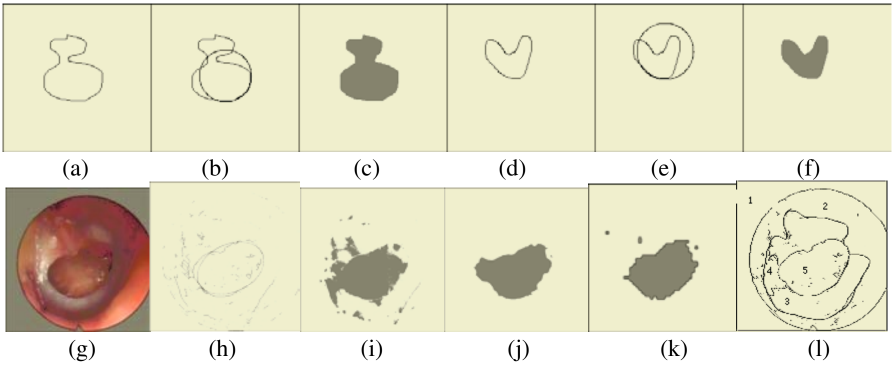


Fig. 4. The random calabash shape and the created results: (a) the original calabash image. (b) the result of Hough Transform. (c) the result of Radar Algorithm. (d) the original heart image. (e) the result of Hough Transform. (f) the result of Radar Algorithm. (g) the original eardrum image. (h) the result of Hough Transform. (i) the result of Narrow-sensed Radar Algorithm. (j) the result of Multi-Step Narrow-sensed Radar Algorithm. (k) the result of Vision Model method. (l) the result using Broad-sensed Radar Algorithm to extract regions

Another image to be processed is an image of eardrum with a hole inside as the test image, as show in Fig.4 (g). The purpose of this experiment is to find the location and size of the hole in the eardrum.

In this experiment, we firstly perform the circular curve fit operation and get the analytic equation of the outside circle. Then we remove the part of the inside circle.

In the second step, we perform elliptical curve fit operation. We use two-step Hough Transform in this operation. We should firstly find the center of the ellipse and then get other parameters. Because the shape of the object in the image is not regular as well as other influencing factors, it is often difficult to find reasonable parameters in most cases. Fig 4 (h) shows the processing result in the case of getting the parameters for ellipse. The error rate of the area of the region is not large. The number of pixels in the region is 27000, and the number that our method finds is 26000. The error rate is about 5%. But the matching of the region is not very good. The number of pixels of the regions where the computed result and the real image do not intersect (i.e. the region that the ellipse dose not cover over the object region), as well as the region where the ellipse covers over but there is object existed is about 7000. It is about 25% of the total pixels. The number of pixels of the intersected regions is about 24000, and is 90% of the total pixels.

Fig 4 (i) shows the experimental result using Narrow-sensed Radar Algorithm to extract regions. There are about 37900 pixels in the extracted region. The error rate is about 40%. The number of pixels of regions where the computed result and the real image do not intersect is about 12000. It is 45% of the total pixels.

Fig 4 (j) shows the experimental result using Multi-Step Narrow-sensed Radar Algorithm to extract regions. There are about 29400 pixels in the extracted region. The error rate is about 8%. The number of pixels of where the computed result and the real image do not intersect is about 4000. It is 14% of the total pixels.

The experimental result using Vision Model method [7] to extract regions is shown in Fig 4(k). There are about 25500 pixels in the extracted region. The error rate is about 5%. The number of pixels of where the computed result and the real image do not intersect is about 2700. It is 10% of the total pixels.

Fig 4(l) shows the experimental result using Broad-sensed Radar Algorithm to extract regions. There are about 27500 pixels in the extracted region. The error rate is about 7%. The number of pixels of where the computed result and the real image do not intersect is about 3000. It is 11% of the total pixels.

5 Analyses and Evaluation of the Algorithm

In the first place, we compare different algorithms of Hough Transform, Narrow-sensed Radar Algorithm, Multi-Step Radar Algorithm, and Vision Model method based on error rates. We divide error rates into two categories: general error rate and matching error rate. The general error rate refers to the ratio between computed result from different algorithms and the actual value. The matching error rate refers to the ratio between the number of pixels computed from algorithms, which do not intersect with actual regions, and the number of pixels in the actual regions. The matching error rate represents the degree of the matching between two regions of the extracted region and actual region. The different error rates are shown in Table 1.

Table 1. The results of error rates of different algorithms

Error rates	Hough Trans-form	Narrow-sensed Radar Algorism	Multi-Step Narrow-sensed Radar Algorism	Vision Model Algorithm	Broad-sensed Rader Algorithm
General error rate	5%	40%	8%	5%	7%
Matching error rate	25%	45%	14%	10%	11%

In the next place, it is analyses of complexity. For the sake of simplicity, we assume that the image to be processed is with size of $n \times n$, and so there are n^2 pixels in the image. We scan sounding pixels of each pixel in the image. About n/4 pixels are scanned for each angle. So the time complexity of the algorithm is $(n \times n) \times 360 \times$ n/4=90× n^3. The algorithm proposed in this paper only needs one buffer to store the segmented result. So we do not need other spaces expect for an n^2 buffer for the segmented result.

Hough Transform has to design specific algorithms for different kinds of shapes according to their analytic equations. What's more, the spatial complexity and the time complexity are exponentially increased with the increase of the complexity of the equation and number of parameters. For example, Hough Transform of line has two parameters, the parameter space is two dimensions, and the spatial complexity is only O(n^2). On the other hand, three parameters are used for Hough Transform of more complex shape of circles, and the parameter space is three dimensions. The spatial complexity is increased to O(n^3). We may reduce the dimensions of the parameter space, but it is still very complicated. Further, the parameter space of Hough Transform of ellipse is increased to five dimensions, and spatial complexity is increased to O(n^5) . This is very difficult to realize by computer. It is more difficult to extract regions for other more complex shapes. When there are several types of shapes in the image, it is very complex to use Hough Transform. We have considered all types of shapes to be extracted so that we can extract regions of interest. The comparison of complexity of different algorithms is shown in Table 2.

Table 2. Comparison of complexity of different algorithms

	Hough Transform	Narrow-sensed Radar Algorithm	Narrow-sensed Multiple Steps Radar Algorithm	Vision Model Algorithm	Broad sensed Radar Algorithm
complexity	O(n^3)~ ∞	O(n^3)	O(n^3)	>O(n^3)	O(n^3)

6 Conclusions

A new algorithm named Radar Algorithm was presented in this paper. This algorithm is mainly used to process images with discontinuous boundaries. The proposed algorithm segments regions of interest based on SOM network model, physical simulation, and the mathematical morphology, and statistically using the information of partial boundary lines of images. The experiments show that Radar Algorithm has more advantages in spatial complexity and robustness than Hough Transform. Furthermore, the proposed algorithm is simple and easy to implement. It is powerful and efficient for image segmentations.

References

1. Demigny, D., Tawfik-Kamle: A Discrete Expression of Canny's Criteria for Step Edge Detector Performance Evaluation. IEEE Trans. Pattern Analysis and Machine Intelligence, **19** (1997) 1199-1210

2. Demigny, D.: On Optimal Linear Filtering for Edge Detection. IEEE Trans. Image Processing, **11** (2002) 728-737
3. Palmer, J.M., Kittler, M., Petrou, B.D.: An Optimizing Line Finder Using a Hough Transform Algorithm. Computer Science, 77 (1999) 13-23
4. Sudeep-Sarkar, Kim, L. Boyer: On Optimal Infinite Impulse Response Edge Detection Filter. IEEE Trans. Pattern Analysis and Machine Intelligence, **13** (1991) 1154-1170
5. Kohonen, T.: Self-Organization Formation of Topologically Correct Feature Maps. Biological Cybernetics, **43 (**1988) 59-69
6. Pal, S.K., King, R.A.: On Edge Detection of X-Ray Images Using Fuzzy Sets. IEEE Trans Pattern Analysis and Machine Intelligence, PAMI-**5** (1983) 69-97
7. Freeman, H.: On the Encoding of Arbitrary Geometric Configurations. IEEE Trans. Elec. Computers, EC-**10** (1961) 260-268

A Novel Image Interpolator Based on Probabilistic Neural Network with Shapeness/Smoothness Adaptation

Chinghan Chen and Shenghsien Hsieh

Department of Electrical Engineering, I-Shou University
1, Section 1, Hsueh-Cheng Rd. Ta-Hsu Hsiang, Kaohsiung County, Taiwan, 840, China
pierre@isu.edu.tw, shiehmen@ms41.hinet.net

Abstract. In this paper, we propose a novel image interpolator based on Probabilistic Neural Network(PNN) that adjusts automatically the smoothing parameters of interpolative model for varied smooth/edge image region. This method takes into consideration both smoothness(flat region) and sharpness(edge region) characteristics at the same model. A single neuron, combined with PSO training, is used for sharpeness/smoothness adaptation. The experimental results demonstrate that this interpolator possesses better performance than bicubic polynomial interpolation either at flat region or at edge region of images.

1 Introduction

Image interpolation is a popular technique for image manipulation and processing. Its most common application is to provide better visual effect after resizing a digital image for display or printing. In recent years, due to the consumer multimedia products being in vogue, imaging and display device become ubiquitous, and image interpolation is becoming more and more important.

The size and resolution of image display device, like LCD, increase constantly. The display systems require full screen and high quality visual effect. The image interpolation becomes an important process. Besides, digital zooming of DSC (Digital Still Camera) also relies on image interpolation technique.

Divers interpolation methods have been proposed. The simplest and fastest one is Nearest Neighbor (NN) method in which the value of the pixel point the closest is given as the value at the interpolating position [1]. This method is efficient, but its disadvantage is the blocking effect. Another generic method is bilinear interpolation. The interpolative value is given by the weighting average of neighboring pixels. This method considers that the variation value between the neighbor pixels of image is always smooth, despite the shape features like edges in images. It conduce blurry effect on image visualization.

Better interpolative image quality can be produced by bicubic interpolation [2]. Its principle is similar as bi-linear interpolation, but the linear spatial model is replaced by cubic-spline model. Consequently, it provides higher precision. Inevitably, it still has obvious artifact and blurry interpolative effect.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 698–706, 2005.

Aiming at above problems, some researchers proposed various methods to reduce blurry interpolative effect at edge region [3], [4] [5], [6]. These methods can be classified into two categories: one is to carry out edge detection or sharpness estimation before interpolation. This kind of interpolations consumes a great deal of computational resources. Besides, it may produce discontinuities at edge transition region and degrade therefore the visual quality of images. The category resides in the execution of edge enhancement after normal interpolation procedure. In addition to the consumption of extra computing time, it also has the drawback of introducing more noises as a result of twofold image processing stages.

An innovative approach of image interpolation is introduced in this paper. We propose an interpolative PNN (Probabilistic Neural Network) model which is consisted of five layers of neurons: Euclidian layer, Gaussian layer, weighting layer, summation layer and division layer. During the interpolation, the interpolative PNN is adjusted by a sharpness-adaptation single neuron. This approach not only provides smooth interpolation but also preserves sharpness at edge region.

2 Intelligent Image Interpolator

2.1 Probabilistic Neural Network

Specht [7],[8] has introduced a neural classifier architecture, named Probabilistic Neural Network (PNN) that is well adapted to manipulate pattern recognition and classification problem. PNN is constructed on the basis of Bayes theorem and Parzen probability density function estimation.

Specht's PNN model is consisted of three layers of neurons. The middle layer is kernel function layer which uses Gaussian kernel. Fig.1 shows the simplified kernel neuron. With input feature $X=\{x_1,x_2,\ldots,x_N\}$, the estimation of probability density function f for given category C is

$$\text{f=p(X|C)} = e^{-\frac{E}{\sigma^2}} . \tag{1}$$

where σ is smoothing factor.

2.2 Interpolative PNN Model

The interpolative PNN model that we propose is shown in figure 2. It is consisted of five layers neurons: Euclidian layer, Gaussian layer, weighting layer, summation layer and division layer.

In Fig. 2, the Euclidian distance

$$d_i = \left\| U_i - X_i \right\| . \tag{2}$$

U is interpolation position, $U = (u, v)$ for two dimensional image spaces.

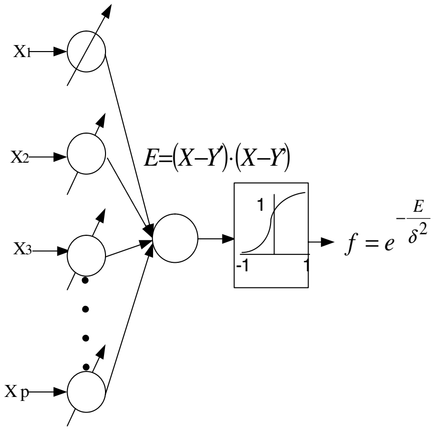


Fig. 1. The simplified PNN model

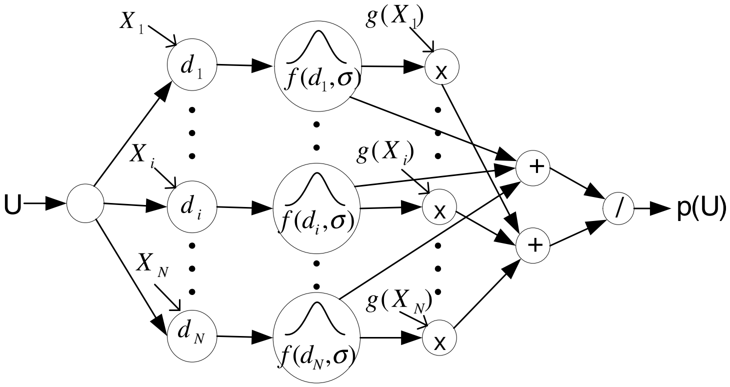


Fig. 2. Proposed interpolative PNN

$X_i = \{(x_i, y_i) | i \in N_U\}$, N_U is neighboring region of interpolation position U

$g(X_i)$ denotes gray level at X_i

$p(U)$ is interpolated value

For an interpolative position U=(u,v), the positions *Xi* , i=0,...,N, of N neighbor pixels in N_U are adopted as connection weights of Euclidian layer. The output is then feedforward to Gaussian neuron for obtaining the probability density f(d, σ). The gray level of neighbor pixel is used as weighting coefficient in weighting layer. Finally the resulting interpolated value is obtained by rule of gravity center:

$$P(U)=\frac{\sum_{i} g(U)*f(d,\sigma)}{\sum_{i} f(d,\sigma)},\ i\in N_U . \quad (3)$$

Where

$$f_i(d,\sigma) = \exp(\frac{-d^2}{\sigma^2}). \tag{4}$$

Fig.3 shows a case of interpolative position and its 16 neighbor pixels.

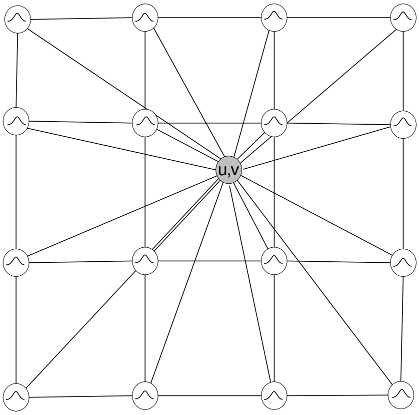


Fig. 3. Interpolative PNN with 16 neighbor pixels

3 Sharpness Adaptation in Interpolative PNN

3.1 Estimation of Smoothing Parameter by Single Neuron

Due to the adoption of Gaussian function as kernel function of interpolative PNN, the interpolation will produce good smoothing result in flat image region. But at edge region, it may cause undesired blurring effect.

To overcome this blurring effect, we have elaborated a mechanism to make interpolative PNN adapt the region smoothing/sharpness. We use a single neuron [9] to adjust the parameter σ of interpolative PNN. The edge feature at each pixel is measured as input vector of single neuron, and the output of neuron is smoothing parameter σ.

Figure 4 shows a single neuron model. When *n* incoming stimulating signals reach the neuron, they will be multiplied by *n* connection weights w_i, i = 1, 2,...*n*. The summation of the multiplication will be added up by a bias and the result is obtained after the transfer function *f( )*.

We define *f( )* to be a parameterized transfer function:

$$o = f(net) = \frac{a}{1+e^{-bnet}}. \tag{5}$$

and

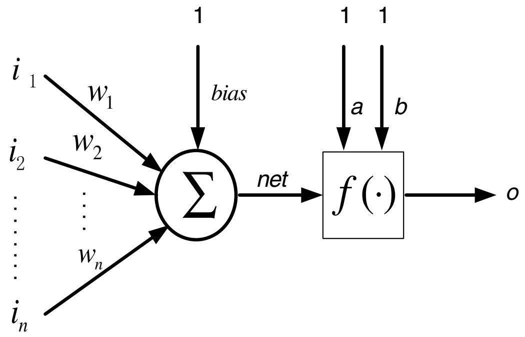


Fig. 4. Single neuron model

$$net = \sum_{j=1}^{n} w_j i_j + bias. \tag{6}$$

where *b* is used to modify the form of the function and *a* its scale. *i* and o are respectively the input and the output.

3.2 Training of Single Neuron Using PSO

PSO is a new bio-inspired optimization developed by Kenney and Eberhart in 1995[10],[11]. PSO exploits cooperative and social behavior's heuristics, such as shoal of fishes, flock of birds and swarm of insects. The basic algorithm involves the starts from a population of distributed individuals, named particles, which tend to move toward the best solution in the search space. The particles will remember the best solution encountered and the best solution of swarm. At each interaction, every particle adjusts its velocity vector, based on its momentum and the influence of both its best solution and the best solution of swarm.

At time unit t, the position of *i*th particle x_i, $i = 1,2,\ldots,M$, (M is the number of particles) moves by addition of a velocity vector v_i, which is a function of the best position (the best fitness) found by that particle, (p_i , for individual best) and of the best position found so far among all particles of swarm(*g*, for global best). The movement can be formulated as:

$$\begin{cases} v_i = w(t)v_i(t-1) + \varphi_1 \quad \mu_1(p - x_i(t-1)) + \varphi_2 \quad \mu_2(g - x_i(t-1)) \\ x_i(t-1) = x_i(t) + v_i(t) \end{cases}. \tag{7}$$

where *w(t)* is the *inertial weight*, φ the *acceleration constants*, and $\mu \in (0,1)$ the uniformly distributed random numbers.

The PSO is used for training the single neuron to optimize smoothing parameter estimation model. We encode the single neuron by (N+3) parameters, including N weights, 1 bias and 2 adjustable parameters of transfer function. PSO, in offline mode, searches the best set of parameters in the (N+3) dimensional space.

3.3 Fitness Function for PSO

Since the conventional PSNR does not precisely reflect human's visual perception of a quality deviation between two images, we use the percentage edge error (PEE), proposed in[12] as the fitness function to evaluate the quality of PSO's solution:

$$PEE = \frac{E_{ORG} - E_{INT}}{E_{ORG}} \times 100\% . \tag{8}$$

where E_{ORG} is the edge strength of the original image and E_{INT} is that of the interpolated image. The *PEE* measures how close the interpolated image sharpness is from the original image. The smaller *PEE* signifies a better edge preserving during interpolation process.

The complete system architecture of the proposed intelligent image interpolator is shown in figure 5.

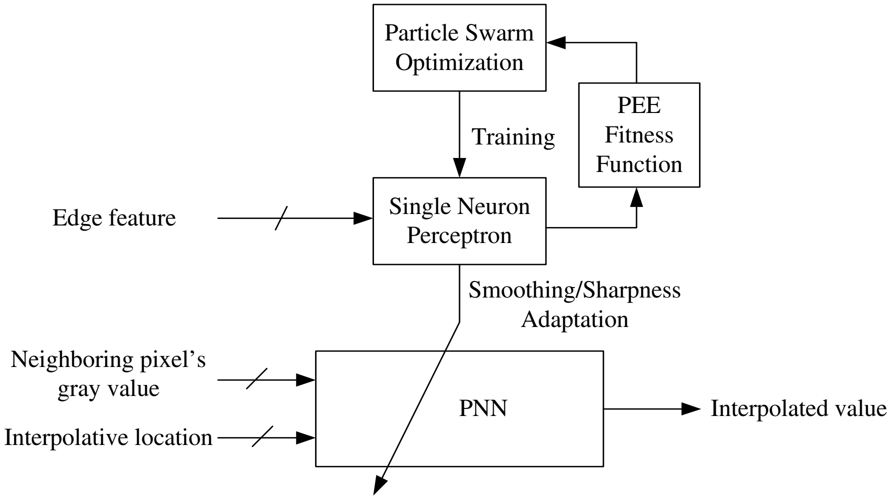
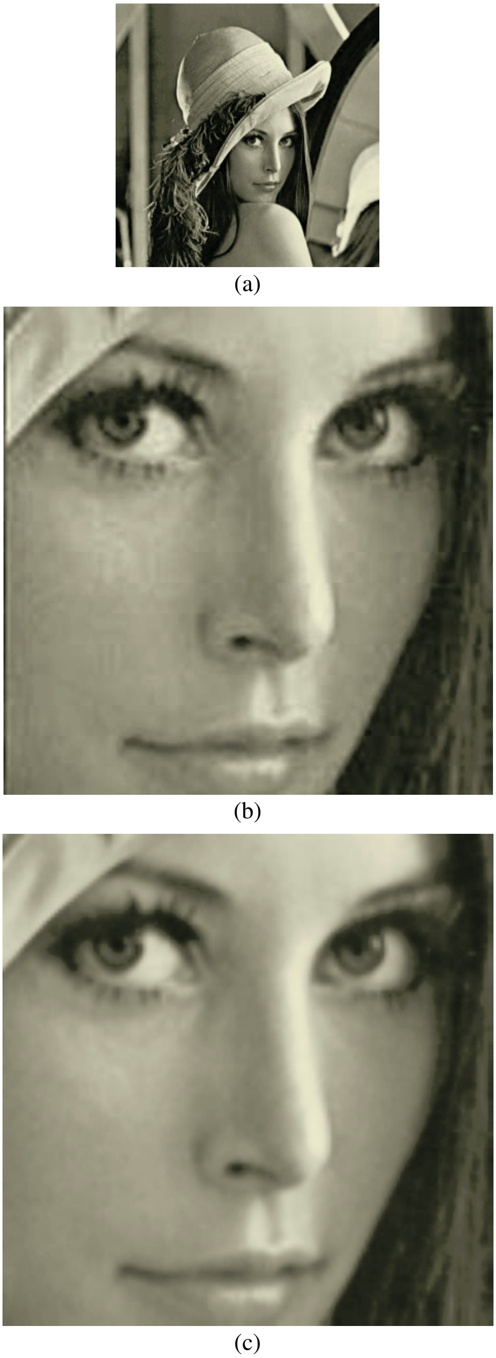


Fig. 5. System architecture of intelligent image interpolator

4 Experiments

We use the Lenna image(Fig. 6a) to test the proposed interpolator, and compare with bicubic interpolation. From the original image, we select a block of 128×128 pixels, and enlarge this block image by a factor of 8 using respectively bicubic interpolation and interpolative PNN. The results are shown in Fig. 6b-c.

We can see in fig. 6(b) uneven effect at skin region, which is a consequence of using bicubic polynomial interpolation. This effect can be more clearly observed in Fig.7. We extract a profile cross eye's region as shown in fig.7, then plot the original



(a)

(b)

(c)

Fig. 6. The 8X image enlargement.(a)original Lenna image; (b)result obtained by bicubic interpolation; (c)result obtained by interpolative PNN

(control) points with two interpolated profiles for comparison. Bicubic interpolation exhibit an overshoot effect at edge region(marked by a red circle). With regard to iPNN interpolation, it reveals better smoothness than bicubic interpolation either at flat region or at edge region of image, even though the contrast between bright side and dark side diminish slightly.

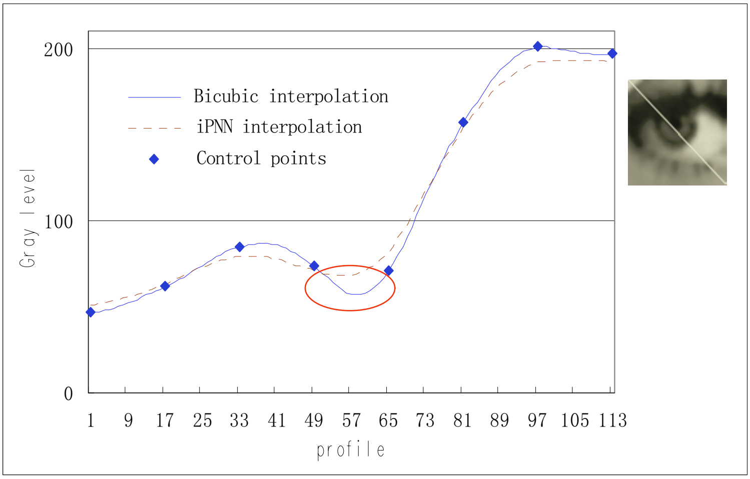


Fig. 7. Comparison of bicubic and iPNN interpolation by cross profile at edge region

5 Conclusions

We have proposed in this paper an interpolative Probabilistic Neural Network (iPNN) as intelligent image interpolator. This interpolator takes into consideration both smoothness(flat region) and sharpness (edge region) at the same model. Owing to the single neuron model combined with PSO training, the interpolator can adapt automatically the smoothing parameters of interpolative model for varied smooth/edge image region. The experimental results demonstrate that this interpolator possesses better performance than bicubic polynomial interpolation either at flat region or at edge region of image.

Acknowledgements

Authors would like to thank the National Science Council for their financial support on this research by the grant NSC-92-2213-E-214-015.

References

1. Thevenaz, P., Blu, T., Unser, M.: Interpolation Revisited IEEE Trans. Medical Imaging, **19** (2000) 739-758
2. Key, R.s: Cubic Convolution Interpolation for Digital Image Processing. IEEE Trans. Acoustics, Speech, Signal Processing, **29** (1981) 1153-1160

3. Thurnhofer, S., Mitra, S.: Edge-Enhanced Image Zooming. Optical Engineering, **35** (1996) 1862-1870
4. Battiato, S., Gallo, G., Stanco, F.: A locally Adaptive Zooming Algorithm for Digital Images. Image and Vision Computing, **20** (2002) 805-812
5. Dai, D., Shih, T., Chau, F.: Polynomial Preserving Algorithm for Digital Image Interpolation. Signal Processing, **67** (1998) 109-121
6. Arandiga, F., Donat, R., Mulet, P.: Adaptive Interpolation of Images. Signal Processing, **83** (2003) 459-464
7. Specht, D.F.: Probabilistic Neural Networks for Classification, Mapping, or Associative Memory. IEEE International Conference on Neural Networks, **1** (1988) 525 -532
8. Specht, D.F.: Enhancements to Probabilistic Neural Networks. International Joint Conference on Neural Networks , **1** (1992) 761-7687
9. Chen, C.: Automatic Design of Neural Networks Based on Genetic Algorithms. ICS'98 Workshop on Artificial Intelligence (1998) 8-13
10. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In Proc. IEEE Int. Conf. Neural Networks, Perth, Australia (1995) 1942-1948
11. Kennedy, J.: The Particle Swarm: Social Adaptation of Knowledge. In Proc. 1997 Int. Conf. Evolutionary Computation, Indianapolis (1997) 303-308
12. Al-Fahoum, A.S., Reza, A.M.: Combined Edge Crispiness and Statistical Differencing for Deblocking JPEG Compressed Images. IEEE Trans. Image Processing, **10** (2001) 1288-1298

Contrast Enhancement for Image with Simulated Annealing Algorithm and Wavelet Neural Network

Changjiang Zhang, Xiaodong Wang, and Haoran Zhang

College of Information Science and Engineering, Zhejiang Normal University
Jinhua, Zhejiang 321004, China
{zcj74922,wxd,hylt}@zjnu.cn

Abstract. A new contrast enhancement algorithm for image is proposed with simulated annealing algorithm (SA) and wavelet neural network (WNN). Incomplete Beta transform (IBT) is used to obtain non-linear gray transform curve. Transform parameters are determined by SA to obtain optimal gray transform parameters. In order to avoid the expensive time for traditional contrast enhancement algorithms, which search optimal gray transform parameters in the whole parameters space, a new criterion is proposed. Contrast type for original image is determined employing the new criterion. Parameters space is given respectively according to different contrast types, which shrinks parameters space greatly. Thus searching direction and selection of initial values of SA is guided by the new parameter space. In order to calculate IBT in the whole image, a kind of WNN is proposed to approximate the IBT. Experimental results show that the new algorithm is able to adaptively enhance the contrast for image well.

1 Introduction

Traditional image enhancement algorithms are as following: point operators, space operators, transform operators and pseu-color enhancement[1]. H.D. Cheng gave a kind of algorithm for contrast enhancement based on fuzzy operator[2]. However, the algorithm cannot be sure to be convergent. Lots of improved histogram equalization algorithms were proposed to enhance contrast for kinds of images[3]. The visual quality cannot be improved greatly with above algorithms. Tubbs gave a simple gray transform algorithm to enhance contrast for images[4]. However, the computation burden of the algorithm was large. Existing many enhancement algorithms' intelligence and adaptability are worse and much artificial interference is required.

To solve above problems, a new algorithm employing IBT, SA and WNN is proposed. To improve optimization speed and intelligence of algorithm, a new criterion is proposed based on gray level histogram. Contrast type for original image is determined employing the new criterion. Contrast for original images are classified into seven types: particular dark (PD), medium dark (MD), medium dark slightly (MDS), medium bright slightly (MBS), medium bright (MB), particular bright (PB) and good gray level distribution (GGLD). IBT operator transforms original image to a new space. A certain objective function is used to optimize non-linear transform parameters. SA, which was given by William, is used to determine the optimal non-linear transform parameters. In order to reduce the computation burden for calculating IBT, a new kind of WNN is proposed to approximate the IBT in the whole image.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 707–712, 2005.

2 IBT

The incomplete Beta function can be written as following:

$$F(u) = B^{-1}(\alpha,\beta)\times\int_0^u t^{\alpha-1}(1-t)^{\beta-1}dt, \quad 0<\alpha,\beta<10. \tag{1}$$

All the gray levels of original image have to be unitary before implementing IBT. All the gray levels of enhanced image have to be inverse-unitary after implementing IBT. Let x shows gray level of original image, g indicates unitary gray level. We have:

$$g = \frac{x-\min(x)}{\max(x)-\min(x)}. \tag{2}$$

Where $\min(x)$ and $\max(x)$ shows the minimum gray level and the maximum one in original image respectively. g is mapped to g':

$$g' = IB\left(a,b,g\right). \tag{3}$$

Let x' shows gray level of enhanced image, we have:

$$x' = \left[\max(x)-\min(x)\right]g' + \min(x). \tag{4}$$

3 Objective Function for Image Contrast Enhancement

The objective function can be written as following [1]:

$$f = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} g'^2\left(i,j\right) - \left[\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} g'\left(i,j\right)\right]^2. \tag{5}$$

Where M,N show width and height of original image. $g'\left(i,j\right)$ Shows gray level at $\left(i,j\right)$ in enhanced image. More f is, more well proportioned the distribution of image gray level is.

4 Contrast Classification for Image Based on Histogram

Based on gray level histogram, contrast classification criterion can be described in Fig.1:

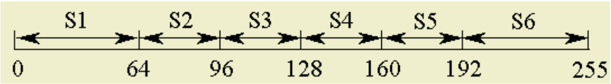


Fig. 1. Image classification sketch map based on gray level histogram

Given that original image has 255 gray levels, the whole gray level space is divided into six sub-spaces: S1, S2, S3 , S4 , S5 , S5. Where Si (i=1, 2, ···, 6) is the number of all pixels which distribute in the ith sub-space. Let,

$$S = \max_{i=1}^{6} S_i, \quad S_1 = \sum_{k=2}^{6} S\mathbf{k}, \quad S_2 = \sum_{k=2}^{5} S\mathbf{k}, \quad S_3 = \sum_{k=1}^{5} S\mathbf{k},$$

$$S_4 = S1 + S6, \quad S_5 = S2 + S3, \quad S_6 = S4 + S5,$$

Following classification criterion can be obtained:

```
    $if\ S = S1\ \&\ S1 > S_1$
    Image is PB;
  $elseif\ S_2 > S_4\ \&\ S_5 > S_6\ \&\ S_5 > S1\ \&\ S_5 > S6\ \&\ S2 > S3$
    Image is MD;
  $elseif\ S_2 > S_4\ \&\ S_5 > S_6\ \&\ S_5 > S1\ \&\ S_5 > S6\ \&\ S2 < S3$
    Image is MDS;
    $elseif\ S_2 > S_4\ \&\ S_5 < S_6\ \&\ S1 < S_6\ \&\ S6 < S_6\ \&\ S4 > S5$
    Image is MBS;
    $elseif\ S_2 > S_4\ \&\ S_5 < S_6\ \&\ S1 < S_6\ \&\ S6 < S_6\ \&\ S4 < S5$
    Image is MB;
    $elseif\ S = S6\ \&\ S6 > S_3$
    Image is PB;
    $else$
    Image is GGLD;
  $end$
```

Where symbol & represents logic "and" operator.

5 Transform Parameters Optimization with SA

We will employ the SA, which was given by William L. Goffe, to optimize transform parameters [5]. If the algorithm is used directly to enhance image contrast, it will result in large computation cost and worse robust to initial points. The range of α and β can be determined by Tab.1 so as to solve above problems.

Table 1. Range of α and β

Parameter	PD	MD	MDS	MBS	MB	PB
α	[0 , 2]	[0 , 2]	[0 , 2]	[1 , 3]	[1 , 4]	[7 , 9]
β	[7 , 9]	[1 , 4]	[1 , 3]	[0 , 2]	[0 , 2]	[0 , 2]

Let $\mathbf{x} = (\alpha, \beta)$, $F(\mathbf{x})$ is function to be minimized, corresponding to (4). Where $a_i < \alpha, \beta < b_i$ ($i = 1,2$), a_i and b_i ($i = 1,2$) can be determined by Tab.1.

Having made lots of experiments and tests, a satisfactory result will be obtained to all contrast types of images when parameters above are determined as follows: $N_S = 20$, $N_T = 100$, $c_i = 2$, $i = 1,2$, $T_0 = 5$, $r_T = 0.95$. Detail steps on simulated annealing algorithm can be obtained in Ref [5].

6 IBT Calculation with WNN

IBT is calculated pixel-to-pixel. Operation burden is very large when pixels in original image are large. Different IBT have to be calculated to different α and β. Different IBT need to be calculated one time in every iterative step during optimization. To improve operation speed during the whole optimization, a new kind of wavelet neural network is proposed.

6.1 Topology Structure of WNN

Let $f(x) \in L^2(R^n)$, WNN can be described approximately as follows:

$$Wf(x) = \sum_{i=1}^{N} w_i \psi[(a_i x - \tau_i)]. \tag{6}$$

where τ_i is translation factor, a_i is scale factor, $Wf(x)$ shows the output of WNN. The translation factor, scale factor and wavelet basis function, which are on the same line, is called wavelet unit. Fig.2 shows the topology structure for the above WNN.

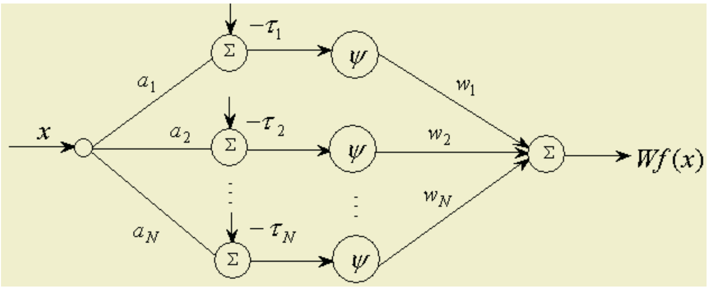


Fig. 2. Structure of wavelet neural network

6.2 Parameter Estimation for WNN

Parameters to be estimated are w_i, a_i, τ_i, $i = 1,2,\cdots,N$ (where N is the number of wavelet unit). "Forgetting factor" algorithm is used to train weight of WNN. Iterative prediction error algorithm is employed to train translation factors and scale factors. Weight, translation factors and scale factors are trained iteratively and mutually with above two algorithms [6].

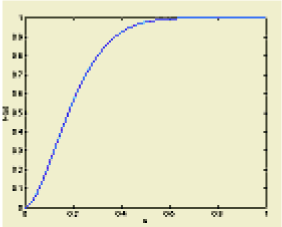


Fig. 3. Transform curve

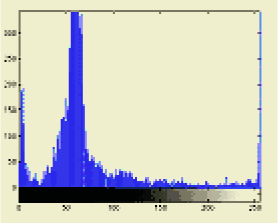


Fig. 4. Histogram of original image

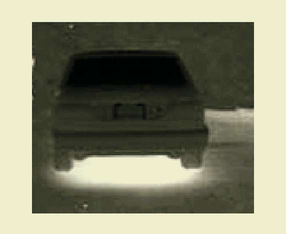


Fig. 5. Infrared car

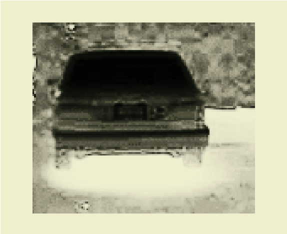


Fig. 6. Enhancing by HE

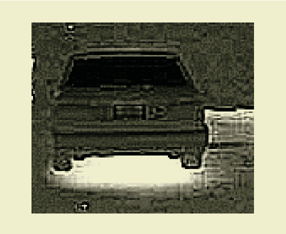


Fig. 7. Enhancing by US

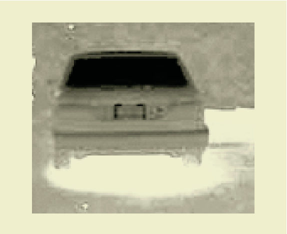


Fig. 8. Enhancing by new algorithm

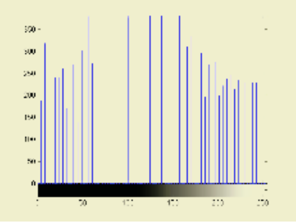


Fig. 9. Histogram by HE

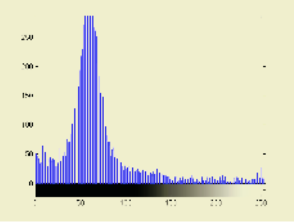


Fig. 10. Histogram by USM

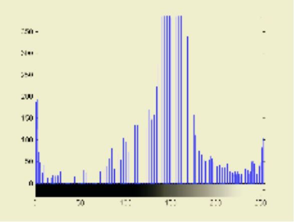


Fig. 11. Histogram by new algorithm

6.3 IBT Approximation Using WNN

IBT can be calculated by the above WNN. Parameters α, β, g are input to trained WNN and output g' for IBT is obtained directly. 100000 points are selected as sample sets. Parameter α and β, which are between 1 and 10, are divided into 10 parts at the same interval. Parameter x, which is between 0 and 1, is divided into 1000 parts at the same interval. 25 wavelet units are selected. The dimension number of input layer and output layer are determined according to the dimension number of input samples and output samples. Mexican hat wavelet is selected as mother wavelet:

$$\psi(x) = (1 - x^2)e^{-x^2/2}. \tag{7}$$

The "forgetting factor" $\alpha = 0.97$ in the WNN. Mean square error is selected as error index and set as 0.00001.

7 Experimental Results

Fig.3 shows transform curve, where $\alpha = 1.9224, \beta = 7.6299$. Fig.4 indicates histogram of original image. Fig.5 shows infrared car image. Two traditional contrast enhancement algorithms are compared with the new algorithm. They are histogram equalization (HE) and unsharpened mask algorithm (USM) respectively. Fig.6-Fig.8 are enhanced images using HE, USM and the new algorithm respectively. Fig.9-Fig.11 show histograms of enhanced images using above three algorithms respectively. It is very obvious that the new algorithm is more excellent in visual quality than HE and USM.

8 Conclusion

Experimental results show that the new algorithm can enhance adaptively image contrast effectively while keep detail information well in original image. The new algorithm is more excellent than HE and USM in visual quality. The computation complexity is O (MN), where M and N are width and height in the original image.

References

1. Rosenfield, A., Avinash, C K: Digital Picture Processing. New York: Academic Press (1982)
2. Cheng, H.D., Xu, H.: A Novel Fuzzy Logic Approach to Mammogram Contrast Enhancement. Information Sciences, **148** (2002) 167-184
3. Stark, J.A: Adaptive Image Contrast Enhancement Using Generalizations of Histogram Equalization. IEEE Transactions on Image Processing, **9** (2000) 889-896
4. Tubbs, J D: A Note On Parametric Image Enhancement. Pattern Recognition, **30** (1997) 616-621
5. Goffe, W.L., Ferrier, G.D., Rogers, J.: Global Optimization of Statistical Functions with Simulated Annealing. Journal of Econometrics, **60** (1994) 65-99
6. Wang, M., Zhang, C., Fu, M.: Simulation Study of a Kind of Wavelet neural Network Algorithm Used in Approaching Non-Linear Functions. Journal of Beijing Institute of Technology, **22** (2002) 274-278

Adaptive Constructive Neural Networks Using Hermite Polynomials for Image Compression

Liying Ma and Khashayar Khorasani

Department of Electrical and Computer Engineering, Concordia University
Montreal, Quebec, Canada H3G 1M8
kash@ece.concordia.ca

Abstract. Compression of digital images has been a very important subject of research for several decades, and a vast number of techniques have been proposed. In particular, the possibility of image compression using Neural Networks (NNs) has been considered by many researchers in recent years, and several Feed-forward Neural Networks (FNNs) have been proposed with reported promising experimental results. Constructive One-Hidden-Layer Feedforward Neural Network (OHL-FNN) is one such architecture. We have previously proposed a new constructive OHL-FNN using Hermite polynomials for regression and recognition problems, and good experimental results were demonstrated. In this paper, we first modify and then apply our proposed OHL-FNN to compress still images and investigate its performance in terms of both training and generalization capabilities. Extensive experimental results for still images (Lena, Lake, and Girl) are presented. It is revealed that the performance of the constructive OHL-FNN using Hermite polynomials is quite good.

1 Introduction

In the past several decades numerous techniques have been proposed to compress digital images to ease their storage and transmission over communication channels [1].In particular, in the past decade, a large number of image compression algorithms using Neural Networks (NNs) have been proposed [2]-[4], such as auto-associative neural networks [5], Self-Organizing Kohonen Map (SOM) [6], cellular neural networks [7], counter-propagation [8], among others.

In conventional NN-based image compression techniques, Multi-Layer Perceptron (MLP) type feedforward neural networks were first considered [9]-[12]. This is due to their structural elegance, abundance of training algorithms and good generalization capabilities. In these FNN-based algorithms, an image is divided into small square blocks of pixels, which are then framed into patterns for NN training. The size of the block generally depends mainly on the nature of the image being compressed and the training algorithm used. The hidden layer in the middle of the network has fewer number of nodes than the input and output layers, and the hidden neuron output values associated to all the blocks

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 713–722, 2005.

are considered as the compressed image or transmitted image to the receiver. If the number of all the output values for the middle layer units plus their connections on its right hand side are less than the total number of pixels of the entire image, a compression is then achieved. The fewer is the number of units in the middle layer, the higher will be the degree of compression. The FNN is symmetric in structure, with the network from input layer to the middle hidden layer acting as data compressor, while the network from the middle layer to the output layer playing the role of a reconstructor. This structure works quite well as already shown by many experimental results in the literature [9]-[12]. However, this technique suffers from two major limitations: first, a large network size that leads to long training cycle and a slow convergence; and second, the difficulty in determining *a priori* a proper network size for a given problem.

To overcome the above two difficulties with the MLP-based techniques, the idea of using only OHL-FNN was proposed in Benbenisti et al [13] (and the references therein). This idea can address the first limitation at the expense of a reduced quality of the reconstructed image for the same compression ratio, however it is still unable to address the second problem. A constructive OHL-FNN was proposed for image compression by Setiono and Lu [14], which is a close variant of the dynamic node creation algorithm by Ash [15]. Their approach retrains the whole network by a second-order training algorithm each time a new hidden unit is added to the network, and is therefore not as efficient as the constructive OHL-FNN that is developed by Kwok et al. [16], where the input-side weights of the hidden unit added to the existing network are frozen and are not retrained in the subsequent network training. We have modified this efficient OHL-FNN by introducing some new pruning techniques and have applied it to regression problems and still image compression [17]. Application of our technique to two "still" images, namely the Girl and the Lena, yielded quite promising results.

In the above-mentioned constructive OHL-FNN [17], all the hidden units have identical sigmoidal activation functions. Although this OHL-FNN is simple in both structure and training efficiency, yet it has a wide applicability due to its "universal approximation property", that is, a OHL-FNN can approximate a continuous function to any desired accuracy as long as enough hidden units with sigmoidal activation functions are included. However, it has not been proven that the use of the same activation functions for all the hidden units is the best or optimum choice for performance and generalization considerations. Consequently, there are a lot of opportunities for attempting to possibly improve the performance of the OHL-FNN by using more appropriate activation functions for the hidden units rather than the identical sigmoidal functions. Motivated from this observation, we have proposed a new constructive OHL-FNN using Hermite polynomials for regression and recognition problems, and good experimental results were demonstrated [18], [19].

In this work, our objective is to apply the OHL-FNN with Hermite polynomial activation functions to compress still images and to investigate its performance in both training and generalization. Extensive experimental results for still images (Lena, Lake, and Girl) are presented. The organization of the paper is as follows. Section 2 gives the outline of our constructive OHL-FNN incorporated with Hermite polynomial activation functions. Experimental results for still images are presented in Section 3. Conclusions are included in Section 4.

2 Constructive Polynomial OHL-FNN for Image Compression

The constructive OHL-FNN with Hermite polynomials [18] is modified for image compression and is given in Fig.1. All the network output units are linear. The first hidden unit is selected as a linear activation function so that the network initiates with the simplest activation function. From the second hidden unit onwards the activation function of a unit will be a Hermite polynomial of order $n - 2$ (where n is the number of hidden units). Each time a new hidden unit is added to the net the order of the Hermite polynomial will be raised by one, giving the hidden layer a hierarchical nonlinearity that may be suitable for the compression task. The formal description of the polynomials are explained next.

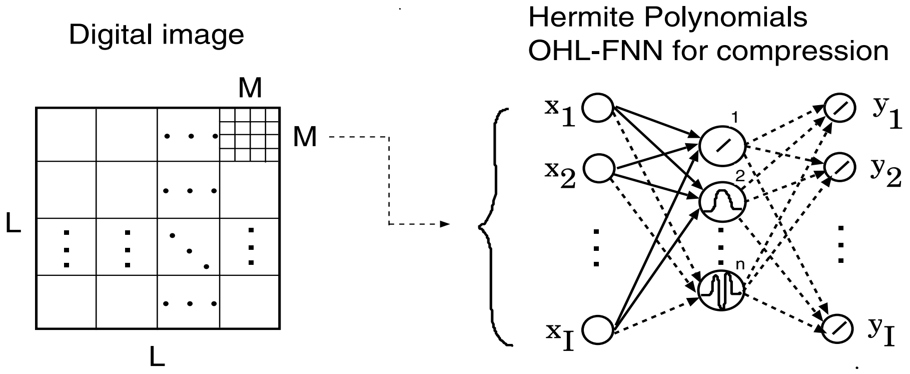


Fig. 1. Constructive OHL-FNN with Hermite polynomials for image compression

2.1 Hermite Polynomials

The formal mathematical description of the orthogonal Hermite polynomials [20] (see also the references therein) are specified as follows:

$$H_0(x) = 1, \tag{1}$$

$$H_1(x) = 2x, \tag{2}$$

$$\cdots$$

$$H_n(x) = 2xH_{n-1}(x) - 2(n-1)H_{n-2}(x), \ n \geq 2 \tag{3}$$

where x is the input variable to the polynomial defined over $(-\infty, \infty)$. The definition of $H_n(x)$, in general, may be given by

$$H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n}\left(e^{-x^2}\right), \quad n > 0, \quad H_0(x) = 1. \tag{4}$$

The polynomials given in (1)–(3) are orthogonal to each other, but not orthonormal. The orthonormal Hermite polynomials may be defined by

$$h_n(x) = \alpha_n H_n(x)\phi(x), \tag{5}$$

$$\alpha_n = (n!)^{-1/2}\pi^{1/4}2^{-(n-1)/2}, \tag{6}$$

$$\phi(x) = \frac{1}{\sqrt{2\pi}}e^{-x^2/2}. \tag{7}$$

For $h_n(x)$, the following orthonormal relationship holds

$$\int_{-\infty}^{\infty} h_i(x)h_j(x)dx = \begin{cases} 1 & i = j, \\ 0 & i \neq j. \end{cases} \tag{8}$$

The first-order derivative of $h_n(x)$ can be easily obtained by the virtue of the recursive relationship (3), as follows:

$$\frac{dh_n(x)}{dx} = (2n)^{1/2}h_{n-1}(x) - xh_n(x), \quad n \geq 1, \tag{9}$$

$$\frac{dh_0(x)}{dx} = \alpha_0 \frac{d\phi(x)}{dx} = -xh_0(x). \tag{10}$$

In [20] the orthonormal Hermite polynomials were used as basis functions to model 1-D signals in biomedical field for the purposes of signal analysis and detection. In [21] a selected group of the orthonormal Hermite polynomials in a weighted-sum form is used as the activation functions of all the hidden units.

2.2 Image Compression Using Constructive Polynomial OHL-FNN

A digital image of size $L \times L$ is to be compressed by a constructive OHL-FNN that utilizes Hermite polynomials as activation functions. The image is first divided into square blocks of equal size $M \times M$ (see Fig.1). Each square block is then arranged into a vector ($I \times 1$, $I = M^2$). All these vectors are put together as a network training matrix $\mathbf{X}$ of size $I \times P$, where P is the number of square blocks and $P = L^2/M^2$. The target matrix $\mathbf{T}$ of the neural network is the same as the input matrix $\mathbf{X}$. As part of pre-processing step the input patterns applied to the neural network are normalized by

$$x_i^j = U_i^j / U_{max} \tag{11}$$

where x_i^j is the normalized i-th element of the j-th input vector, U_i^j is the gray-level value of the corresponding original pixel, and U_{max} is the largest possible gray-level value of the image being compressed. The output of the k-th hidden unit for the j-th input pattern is given by

$$z_k^j = f_k(s_k^j), \quad with \quad s_k^j = \sum_{i=1}^{I} \left(w_{k,i} x_i^j + b_{z,k} \right) \tag{12}$$

where $w_{k,i}$ and $b_{z,k}$ are the input-side weights and bias, $f_k(\cdot)$ is the activation function of the k-th hidden unit with $f_1(x) = x$, $f_2(x) = h_0(x)$, $f_3(x) = h_1(x)$, etc. Output of the o-th output node will be defined according to

$$y_o^j = g\left(\sum_{k=1}^{n-1} v_{o,k} z_k^j + b_o \right) \tag{13}$$

where $v_{o,k}$, b_o and $g(\cdot)$ are the input-side weights, bias and linear activation function of the o-th output node, respectively, and $n-1$ is the number of hidden units of the present (active) network. The function $g(\cdot)$ can also be nonlinear, but linear type output activation functions will simplify both network training and network implementation and are therefore selected in this work.

The initial network with a single linear hidden unit is trained by the BP-type algorithm. The constructive training starts from the 2nd hidden unit with activation function $h_0(\cdot)$. The constructive network training is divided into two phases: Input-side training and output-side training. A general correlation-based objective function [22] for input-side training of the n-th hidden unit is given as

$$J_{input} = \sum_{o=1}^{I} \left| \sum_{j=1}^{P} (e_{n-1,o}^j - \bar{e}_{n-1,o})(f_n(s_n^j) - \bar{f}_n) \right| \tag{14}$$

where $e_{n-1,o}^j = y_o^j - T_o^j$, $\bar{e}_{n-1,o} = \frac{1}{P}\sum_{j=1}^{P} e_{n-1,o}^j$, and $\bar{f}_n = \frac{1}{P}\sum_{j=1}^{P} f_n(s_n^j)$. The "quickprop" algorithm [22] is used to maximize the above objective function. The output-side training is performed according to the following summed square error criterion

$$J_{output} = \frac{1}{2} \sum_{o=1}^{I} \sum_{j=1}^{P} \left(y_o^j - T_o^j \right)^2 . \tag{15}$$

In our experiments, the output-side training is performed in an LS sense by computing the pseudo-inverse exactly, since the output layer activation function $g(\cdot)$ is linear.

The Peak Signal-to-Noise Ratio (PSNR) used to measure the quality of the reconstructed and the generalized/compressed images (8 bits/pixel) is given by

$$PSNR \ = \ 10 \ log_{10} \left(\frac{P \times I \times (255)^2}{\sum_{o=1}^{I} \sum_{j=1}^{P} (y_o^j - T_o^j)^2} \right) . \tag{16}$$

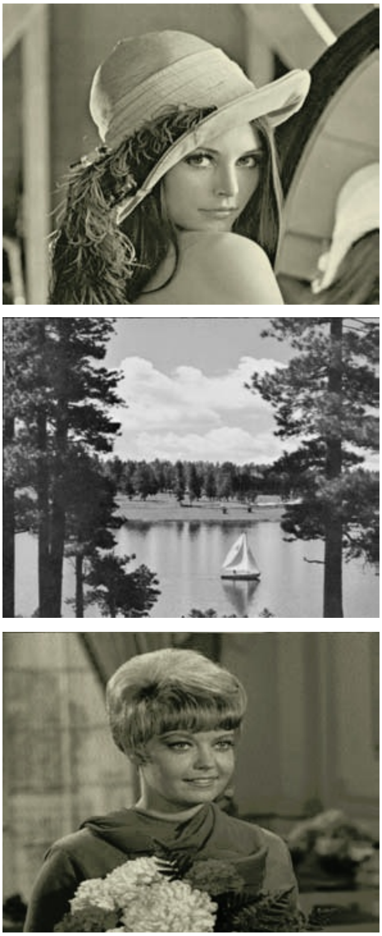


Fig. 2. Original images of the Lena, the Lake, and the Girl (512 × 512 pixels, bite rate R=8 bits/pixel

The compression ratio is computed using the expression (each hidden unit output and output-side weights are supposed to be quantized using 8 bits)

$$\rho_1 = \frac{L \times L}{n \times P + n \times I + I + 1}. \tag{17}$$

Two types of sensitivity measures may be introduced to prune the redundant input-side weights as the constructive learning proceeds in order to produce a smaller network without degenerating the generalization capability of the resulting network [17]. In our previous work [17] the pruning method B was found to be more effective than the pruning method A, and is therefore used in this work. Details of these two pruning methods are available from our previous paper [17].

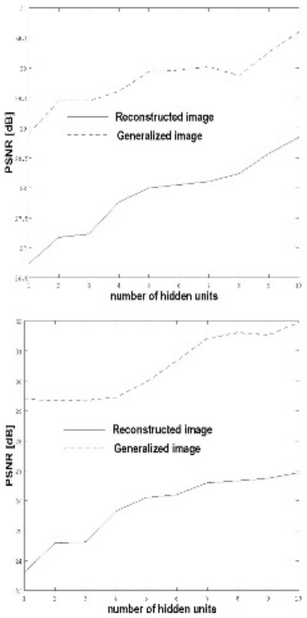


Fig. 3. PSNRs of the reconstructed Lena image and the generalized Girl image, and reconstructed Lake image and the generalized Girl image

3 Experimental Results

In this section, we present the experimental results obtained when applying the modified polynomial OHL-FNN to the compression of still images.

In this section the results for the compression of still images Lena, Lake and Girl are presented. Fig.2 shows the original three still images. The Lena and the Lake are used to train two constructive networks, and subsequently the two networks so trained are used to generalize the Girl image. The PSNRs for the reconstructed Lena and the generalized Girl are given in Fig.3 with respect to the number of hidden units and the PSNRs for the reconstructed Lake and the generalized Girl. The block size in all these simulations is 4-by-4. We can observe from Figs. 3 and 4 that the polynomial constructive OHL-FNNs are capable of compressing the still images, and the trained networks may be used to generalize quite satisfactorily. Furthermore, it is interesting to note that the generalized Girl in both cases has even higher PSNR when compared with the reconstructed Lena and Lake. The Lena and the Lake images are more “complicated” than the Girl and have been found in our experiments more difficult to train. This implies that the proposed constructive OHL-FNN trained by a difficult image may generalize better with simpler images.

Pruning method B described in details in [17] is applied in the network training in order to reduce the number of input-side weights. Pruning level is set to 10%. The input-side weights of the network trained by the Lena image

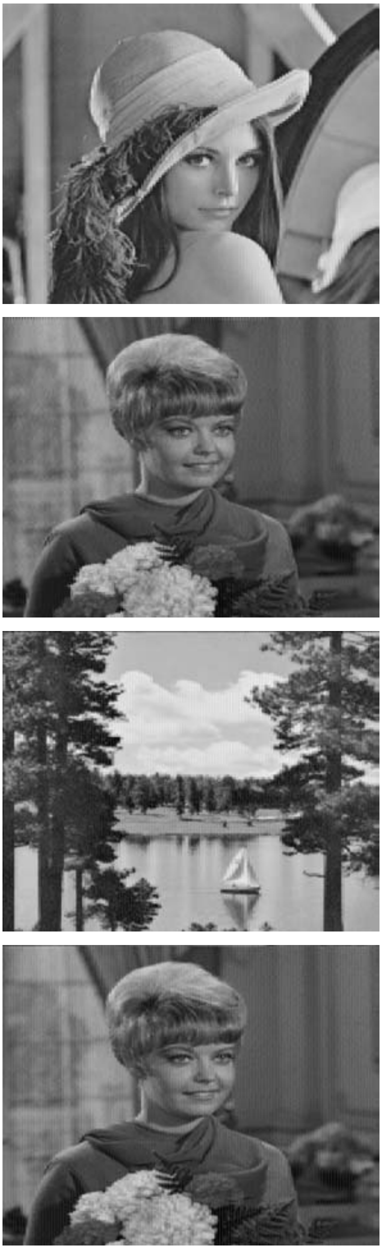


Fig. 4. Reconstructed and generalized images: (a) & (b) training with Lena (c) & (d) training with Lake; (512 × 512 pixels, 5 hidden units, bite rate R=8 bits/pixel)

is reduced by 12.5%, whereas a weight reduction of 13.3% is obtained for the network trained by theLake image.

4 Conclusions

In this paper we have presented a modified constructive polynomial OHL-FNN for the purpose of image compression. Experimental results for three still images were presented. From these results, it follows that the polynomial OHL-FNN can perform successfully image compression. This we believe is due to the rich nonlinearity of the Hermite activation functions for the purpose of image compression. Further experimental and theoretical considerations are needed to establish advantages and disadvantages of our proposed architecture. Comparisons with the other constructive OHL-FNN having identical sigmoidal hidden activation function is a topic of future work.

References

1. Netravali, A. N., Haskell, B. G.: Digital Pictures, Representation, Compression, and Standards. Applications of Communications Theory, Series Editor: R. W. Lucky, Second Edition, Plenum Publishing Corporation, New York (1995)
2. Dony, R. D., Haykin, S.: Neural Network Approaches to Image Compression. Proceedings of the IEEE, **83** (1995) 288-303
3. Cramer, C.: Neural Networks for Image and Video Compression: a Review. European Journal of Operational Research, **108** (1998) 266-282
4. Jiang, J.: Image Compression with Neural Networks — a Survey. Signal Processing:Image Communication, **14** (1999) 737-760
5. Basso, A., Kunt, M.: Autoassociative Neural Networks for Image Compression. European Trans. on Telecommunications and Related Technologies, **3** (1992) 593-598
6. Amerijckx, C., Verleysen, M., Thissen, P., Legat, J.-D.: Image Compression by Self-Organized Kohonen map. IEEE Trans. on Neural Networks, **9** (1998) 503-507
7. Venetianter, P. L., Roska, T.: Image Compression by Cellular Neural Networks. IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications, **45** (1998) 205-215
8. Sygnowski, W., Macukow, B.: Counter-propagation Neural Network for Image Compression. Optical Engineering, **35** (1996) 2214-2217
9. Cottrell, G.W., Munro, P., Zipser, D.: Learning Internal Representations from Gray-level Images: An Example of Extensional Programming. Proc. 9th Annual Conf. Cognitive Science Society, (1987) 461-473
10. Namphol, A., Chin, S. H., Arozuliah, M.: Image Compression with a Hierarchical Neural Network. IEEE Trans. on Aerospace and Electronic Sys., **32** (1996) 326-337
11. Chang, Y., Kumar, D., Mahalingam, N.: Data Compression for Image Recognition. Proc. IEEE TENCON - Speech and Image Technologies for Computing and Telecommunications, (1997) 399-402
12. Panda, G., Singh, P.: A Novel Scheme of Data Compression and Reconstruction Using Multilayer Artificial Neural Network. Journal of the IETE, **41** (1995) 329-334
13. Benbenisti, Y., Kornreich, D., Mitchell, H. B., Schaefer, P. A.: New Simple Three-layer Neural Network for Image Compression. Opt. Eng., **36** (1997) 1814-1817
14. Setiono, R., Lu, G.: Image Compression Using a Feedforward Neural Network. Proc. IEEE Int. Conf. Neural Networks, (1994) 4761-4765

15. Ash, T.: Dynamic Node Creation in Backpropagation Networks. Connection Sci., **1** (1989) 365-375
16. Kwok, T. Y., Yeung, D. Y.: Objective Functions for Training New Hidden Units in Constructive Neural Networks. IEEE Trans. on Neural Networks, **8** (1997) 1131-1148
17. Ma, L., Khorasani, K., New Pruning Techniques for Constructive Neural Networks with Application to Image Compression. Proc. of SPIE, Vol.4052, pp.298-308(2000)
18. Ma, L., Khorasani, K.: Adaptive Structure Feed-forward Neural Networks Using Polynomial Activation Functions. Proc. of SPIE, Vol.4055, pp.120-129(2000).
19. Ma, L., Khorasani, K.: Constructive Hermite Polynomial Feedforward Neural Networks with Application to Facial Expression Recognition. Proc. of SPIE (2001)
20. Rasiah, A. I., Togneri, R., Attikiouzel, Y.: Modeling 1-D Signals Using Hermite Basis Functions. IEE Proc.-Vis. Image Signal Process., **144** (1997) 345-354
21. Hwang, J. N., Lay, S. R., Maechler, M., Martin, D., Schimert, J.: Regression Modeling in Back-propagation and Projection Pursuit Learning. IEEE Trans. Neural Networks, **5** (May 1994) 342-353
22. Fahlman, S. E., Lebiere, C.: The Cascade-correlation Learning Architecture. Tech. Rep., CMU-CS-90-100, Carnegie Mellon University (1991)

Compression of Remote Sensing Images Based on Ridgelet and Neural Network

Shuyuan Yang, Min Wang, and Licheng Jiao

Institute of Intelligence Information Processing, National Lab of Radar Signal Processing
Xi'an, Shaanxi 710071, China
syyang@xidian.edu.cn

Abstract. To get a high-ratio compression of remote sensing images, we advanced a new compression method using neural network (NN) and a geometrical multiscale analysis (GMA) tool-ridgelet. Ridgelet is powerful in dealing with linear singularity (or curvilinear singularity with a localized version), so it can represent the edges of images more efficiently. Thus a network for remote sensing image compression is constructed by taking ridgelet as the activation function of hidden layer in a standard three-layer feed-forward NN. Using the characteristics of self-learning, parallel processing, and distributed storage of NN, we get high-ratio compression with satisfying result. Experiment results indicate that the proposed network not only outperforms the classical multilayer perceptron, but also is quite competitive on training of time.

1 Introduction

The remote sensing images have extremely intensive data, so they have to be compressed before they are stored and transmitted [1]. By using data compression techniques such as prediction coding, vector quantization and wavelet coding, it is possible to remove some redundant information contained in images while maintaining a good level of visual quality [2]. Wavelet is considered as one of the best static compression tools available for its good property. It can fulfill good result for an 8:1 compression of remote sensing images. However, the result degrades dramatically when the compression ratio is larger than 8:1. Up to now, there are still not very efficient algorithms for the image compression bigger than 8:1.

On the other hand, apart from some traditional methods, new technologies such as neural networks (NNs) and genetic algorithms (GAs) are being developed to explore the future of image compression [3],[4]. NN is considered as possible solutions to problems where high computation rates are required[5]. Using its capability of parallel processing, nonlinear mapping and distributed storage, NN can realize a rapid and adaptive image coding and decoding system. Successful applications of image compression based on back-propagation (BP)[6] and Hebbian learning[7] have now become well established, including self-organizing feather mapping (SOFM) for vector quantization (VQ)[8]and predictive coding[9]. Recently abundant work about NN-based compression methods have been done[10],[11],[12],[13]. The first compression

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 723–729, 2005.

model based on MLP and BP algorithm was investigated by Cottrell[14], whose hidden neurons are far less than input and output neurons. The image is compressed in the hidden layer, and hence we can adjust the number of hidden neurons to get desired high-ratio compression for remote sensing images. However, it uses *Sigmoid* function that has infinite support as the activation function, which leads to slow convergence and existence of local minimum in training. Replacing *Sigmoid* function with other local basis function can achieve better outputs, such as wavelet neural network.

Edges of objects in an image can be looked as the positions where the singularities exist. Paper [15] combined wavelet with BP to obtain a lossless compression scheme. Although wavelet can deal with 1-D singularity, it cannot identify singularities in high dimension, so it performs badly in compression the edges. Therefore a new basis function in high dimension is required to represent singularities more efficiently. Ridgelet is such a function developed from geometrical multiscale analysis (GMA). It is an extension of wavelet to high dimension, and it can give an optimal representation of linear singularity in high dimension, as well as curvilinear singularity by a localized version. By combining ridgelet with MLP, an effective adaptive coding technique is proposed in this paper to get a high-ratio compression of remote sensing images. By using ridgelet as the activation function of a three-layer feed-forward NN, this new model possesses all the advantages of MLP-based compression methods, and the network can compress the linear or curvilinear edges more efficiently, which results in simpler structure and faster convergence of the network.

2 Image Compression Using Neural Network and Ridgelet

2. 1 Ridgelet

As an extension of wavelet to higher dimension, ridgelet is an efficient tool in dealing with directional information. It was first proposed by Candés in 1996 with such definition[16]: If $\psi : R^d \to R$ satisfies the condition of $K_\psi = \int (|\hat{\psi}(\xi)|^2 / |\xi|^d) d\xi < \infty$, then we call the functions $\psi_\tau(x) = a^{-1/2}\psi((u \cdot x - b)/a)$ as ridgelet.

Parameter τ=(a,u,b) belongs to the neuron space $\Gamma = \{\tau = (a,u,b), a,b \in R, a > 0, \|u\| = 1\}$, where a,u,b represent the scale, direction and localization of ridgelet. Denote σ_d as the surface area of the unit sphere S^{d-1}, any function $f \in L^1 \cap L^2(R^d)$ can be expanded as a superposition of ridgelets: $f = c_\psi \int < f, \psi_\tau > \psi_\tau \mu(d\tau) \sigma_d da / a^{d+1} dudb$. Because ridgelet is localized in the narrow band $\{x{:}|u.x\text{-}b|<a\}$, the cross section of ridgelet is a curve like wavelet. There is a line along the ridge, and just this kind of geometry structure makes ridgelet deal with hyperplane singularity efficiently. It has been proved that for a group of functions with singularity, a representation by ridgelet is optimal, and the approximation is faster than Fourier transform and wavelet transform[16].

2.2 Neural Network Based Image Compression with Ridgelet

Using ridgelet as the activation function of a three-layer feed-forward NN, we get the network for compression shown in Fig.1. If there are hyperplane singularities in images, because of the sensitivity of ridgelet to directions, the network can adjust the directions of ridgelets adaptively to locate the singular region.

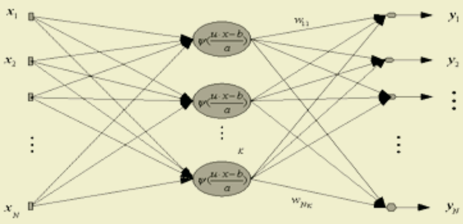

x_1
x_2
x_N
$\psi(\frac{u\cdot x-b}{a})$
$\psi(\frac{u\cdot x-b}{a})$
$\psi(\frac{u\cdot x-b}{a})$
K
w_{11}
w_{NK}
y_1
y_2
y_N

Fig. 1. A *N-K-N* feed-forward network(*K*<*N*) for compression with ridgelet being its activation function in the hidden layer. The input layer takes the original images, and the output layer outputs a same reconstructed image as the input. The output of hidden layer is the extracted feather of the image, which has far less neurons than that of input and output layer. Thus the network can represent the original image with less data, and a high-ratio compression of remote sensing images is achieved by assigning a small *K*, i.e., the number of neurons in hidden layer.

If the size of the image to be compressed is 512×512, the network needs N=512×512 input neurons. To get a compression ratio 16:1, the hidden layer should have K=512×32 neurons. Obviously it is too difficult to train so big a network. So we often divide the image into several sub-blocks, each of which is of size 8×8 or 16×16. Then *N* is equal to the number of pixels in each block. If the image is divided into blocks of size 8×8, for every block we only need to construct a network of 64×4×64, as shown in Fig .2.

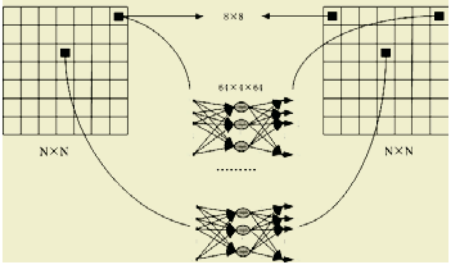


Fig. 2. This is the network for image compression. The compression is divided into two phases-training (coding) and testing (decoding). In the training, a group of images is used to train the network. This is equivalent to compressing the input into a narrow channel represented by the hidden layer, then we can reconstruct the input by the weight between hidden and output layer. The second phase simply involves the entropy coding of the output of the hidden layer.

All the coupling weights connected to each neuron in the hidden layer can be represented by w_{ji}, (j = 1,2,..K; i = 1,2,..N), which is a matrix of K×N. The connections between the hidden and output layer are represented by w_{ij}, which is another weight matrix of N×K. With respect to the fact that NN can operate more efficiently when both their inputs and outputs are limited in[0,1], the pixel value of the image is generalized to [0,1]. Compression is achieved by training the network in such a way that the coupling weights w_{ji} scale the N-dimensional input into a narrow channel of K dimension (K<N) in the hidden layer. Finally we get a network which minimizes the error between input and output. Decoding is a reverse process of coding; the output of hidden layer and the weights in right are needed to recover the image.

2.3 Learning Algorithm of the Network

Assume there is a training set S={X,Y} with P pairs of samples: X=[$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_P$],Y=[$\boldsymbol{y}_1$, ..$\boldsymbol{y}_P$], where $\boldsymbol{x}_i$=[$x_{1i}, \ldots, x_{Ni}$] and $\boldsymbol{y}_i$=[$y_{1i}, \ldots, y_{Ni}$](i=1,..,P) represent the i-th input sample and its corresponding output. Denote Z_i=[$z_{i1}, \ldots, z_{iP}$](i=1,..,K) as the output of the hidden layer. For the p-th sample, the output of hidden layer is:

$$z_{ip} = \psi(\Sigma_{ip}) = \psi[\sum_{j=1}^{N} ((u_{ij} \cdot x_{jp} - b_i) / a_i)] \quad (\sum_{j=1}^{N} \|u_{ij}\|^2 = 1, i = 1,..,K; p = 1,..,P) . \tag{1}$$

In the output layer, w_{ji} represents the weight between the j-th hidden node and i-th output node, and w_{i0} is the threshold of the i-th output node. Denote z_{0p}=1, a linear output layer follows: $y_{jp} = w_{j0} + \sum_{i=1}^{K} w_{ji} z_{ip} (j = 1,....,N)$.Define the squared error of training: $J = \frac{1}{2}\sum_{i=1}^{P} e_i^2 = \frac{1}{2}\sum_{i=1}^{P}\sum_{j=1}^{N}(y_{ji} - d_{ji})^2$.The gradient descent algorithm is adopted to train the network, and.(0<η <1)is the learning step. So the update equation of the parameters of the network is: $\beta_i(k+1) = \beta_i(k) + \eta\partial J(k) / \partial\beta_i (i \in [1, N])$, where . means w,α ,b,u. The gradients of the network are given in equation (2):

$$\frac{\partial J}{\partial w_{ij}} = \sum_{p=1}^{P}(y_{ip} - d_{ip}) z_{jp} \quad (j \in [0, K]), \qquad \frac{\partial J}{\partial u_{ij}} = \sum_{p=1}^{P} \frac{(y_{ip} - d_{ip}) w_{ji} x_{jp}}{\alpha_i} \frac{\partial \psi(\Sigma_{ip})}{\partial \Sigma_{ip}} \quad (j \in [1, K])$$
$$\frac{\partial J}{\partial b_i} = \sum_{j=1}^{N}\sum_{p=1}^{P} \frac{(d_{ip} - y_{ip}) w_{ji}}{\alpha_i} \frac{\partial \psi(\Sigma_{ip})}{\partial(\Sigma_{ip})}, \quad \frac{\partial J}{\partial \alpha_i} = \sum_{j=1}^{N}\sum_{p=1}^{P} \frac{(u_{ij} x_{jp} - b_i)^2 (y_{ip} - d_{ip}) w_{ji}}{a_i} \frac{\partial \psi(\Sigma_{ip})}{\partial(\Sigma_{ip})} . \tag{2}$$

3 Simulation Experiments

In this section, the proposed network for image compression was tested by some experiments. Three remote sensing images of 512×512 pixels are shown in Fig.3. The images are divided into several sub-blocks. In the test, the images are compressed by our proposed network and a classical network in paper[14]. The image was normalized before training and the output of network is quantized by different bit rates. Comparisons are focused on the complexity of encoding and decoding, as well as the measure of peak signal-to-noise ratio (PSNR) which is defined as:

$$PSNR = 10\times\log(255^2 / MSE)(dB),\ (MSE = \sum_i \sum_j (\hat{f}_{ij} - f_{ij})^2 / (m\times n))$$

Here f and $\hat{f}$ are the original and restored image and *m,n* are the dimensions of image. Allowing a maximum 500 iterations, our network is trained using equation (2), η is fixed to 0.1. The PSNR of the compressed image with different bit rates, different size of sub-block, and different compression ratio are shown in Table 1.

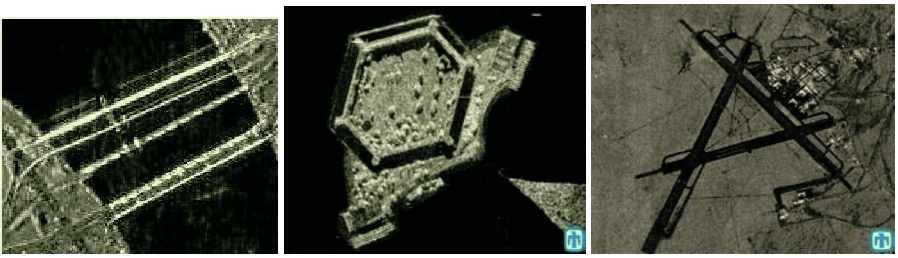


Fig. 3. Remote sensing image for compression.(came from USC-SIPI Image Database, Univ. of Southern California Signal and Image Processing Institute).

Table 1. The compression results of MLP and our method. Here I1-I3 are three images shown in Fig.3. "S" means the classical MLP using *Sigmoid* function and "R" means our method.

PSNR (dB)		Compression ratio 16:1						Compression ratio 32:1					
		Size 8×8			Size 16×16			Size 8×8			Size 16×16		
		b=8	b=6	b=4	b=8	b=6	b=4	b=8	b=6	b=4	b=8	b=6	b=4
I1	S	26.39	25.83	23.40	24.28	23.72	20.43	19.60	19.28	17.85	17.99	16.84	14.61
	R	29.71	28.94	30.35	27.03	26.80	24.55	22.36	21.77	19.03	19.81	18.90	17.33
I2	S	27.36	26.44	24.51	25.38	24.85	23.60	17.81	16.94	14.95	15.19	14.28	12.93
	R	28.68	27.02	26.18	26.99	26.08	25.32	20.05	19.47	17.26	17.45	16.81	14.95
I3	S	24.37	23.46	21.37	23.61	22.35	20.86	16.58	15.45	13.11	14.28	13.55	11.44
	R	26.09	25.88	26.18	25.84	25.17	23.40	19.63	18.70	16.28	16.86	15.94	13.38

Although the results are not same for each execution, the differences are little ranging from *0dB* to 0.1*dB*. The times consumed for b=8 are given in Table 2.

Table 2. The time consumed of two methods.

Time(s)		Compression ratio 16:1		Compression ratio 32:1	
		Size 8×8	Size 16×16	Size 8×8	Size 16×16
I1	S	34.75	44.87	29.73	49.90
	R	29.94	37.58	23.26	40.55
I2	S	37.49	45.10	20.33	44.26
	R	30.24	33.48	15.25	36.38
I3	S	45.50	54.26	29.86	60.92
	R	38.17	45.88	21.61	51.50

From the result we can see that for the high ratio compression, our method has relatively faster learning and better results than classical MLP, especially for a group of images with many linear or curvilinear edges. On the other hand, the full potential of our proposed approach relies on a true parallel form of implementation. Most of the implementations have been based on simulations on serial computers. With the rapid development of VLSI implementations for NN, the speed for training will be increased remarkably.

4 Conclusions

Image compression methods based on feed-forward neural network are a new class of approaches with many good characteristics. They show much promise in the high ratio compression of remote sensing images. To get better compression of the edges of objects in an image, which can be regarded as the linear or curvilinear singularities in high dimension, a new network is proposed by combing ridgelet with feed-forward NN in this paper. Ridgelet is a new developed GMA tool which can preserve the edges well in image compression, and NN can achieve a high-ratio compression for remote sensing images in a parallel way. By combination of ridgelet with NN, this new network is characteristics of a fast learning and better outputs than classical NN models. The simulation results also prove its superiority.

References

1. Manual of Aerial Survey: Primary Data Acquisition. Roger Read (2004)
2. Netravali, A.N., Limb, J.O.: Picture Coding: A Review, Proc. IEEE, **68** (1980) 366-406
3. Robert, D. D. Neural Network Approaches to Image Compression, Processing of IEEE, **83** (1995) 288-303
4. Feiel, H.: A Genetic Approach to Color Image Compression, Symposium on Applied Computing, Proceedings of the ACM symposium on Applied computing (1997) 252-256
5. Pao, Y.H.: Adaptive Pattern Recognition and Neural Networks. Reading, MA: Addison-Wesley (1989)
6. Kohno, R., Arai, M., Imai, H.: Image Compression Using a Neural Network with Learning Capability of Variable Function of a Neural unit, SPIE Visual Communications and Image Processing '90 , **1360** (1990) 69-75
7. Kung, S.Y.: Adaptive Principal Component Extraction (APEX) and Applications, IEEE Transactions on Signal Processing, **42** (1994) 1202-1217
8. Krishnamurthy, A.K., Ahalt, S.C., Melton, D.E., Chen, P.: Neural Networks for Vector Quantization of Speech and Images. IEEE J. on Selected Areas in Communications, **8** (1990) 1449-1457
9. Dianat, S.A., Nasrabadi, N.M., Venkataraman, S.: A Non-linear Predictor for Differential Pulse-code Encoder (DPCM) Using Artificial Neural Networks. Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing , Toronto, Canada (1991) 2793–2796
10. Hatami, S., Yazdanpanah, M.J., Forozandeh, B., Fatemi, O.A: Modified Method for code-book Design with Neural Network in VQ Based Image Compression Circuits and Systems. ISCAS, **2** (2003) 612-615
11. De Almeida, Filho, W.T.: A Neural and Morphological Method for Wavelet-based Image-Compression Neural Networks. IJCNN, **3** (2002) 2168-2173

12. Park, D.: Weighted Centroid Neural Network for Edge preserving Image Compression Neural Networks. IEEE Transactions neural network, (2001) 1134-1146
13. Centroid.: Neural Network for Unsupervised Competitive Learning Dong-Chul Park. IEEE Transactions on Neural Networks (2000) 1045-9227
14. Cottrell., G.W.: Principal Components Analysis of Images via Back propagation. In SPIE Visual Communications and Image Processing, **1001** (1988) 1070-1077
15. Yang, G., Tu, X.: A high Efficiency Image Data Compression Scheme based on Wavelet and Neural Network. Opto-Electronic Engineering, **31** (2004)
16. Candes, E. J., Ridgelet: Theory and Applications. Ph.D. dissertation. Stanford Univ. (1998)

The SAR Image Compression with Projection Pursuit Neural Networks

Jian Ji[1], Zheng Tian[1,2,3], Wei Lin[2], and Yanwei Ju[2]

[1] Department of Computer Science & Technology, Northwestern Polytechnical University
Xi'an, Shaanxi 710072, China
jijiangao@126.com
[2] Department of Applied Mathematics, Northwestern Polytechnical University
Xi'an, Shaanxi 710072, China
[3] Key Laboratory of Education Ministry for Image Processing and Intelligent Control
Huazhong University of Science & Technology, Wuhan, Hubei 430074, China

Abstract. Synthetic Aperture Radar (SAR) image compression is important in image transmission and archiving. We present a new algorithm for SAR image compression based on projection pursuit neural networks. At first, we segment an SAR image into regions of different sizes based on mean value in each region and then constructing a distinct code for each block by using the projection pursuit neural networks. The process is stopped when the desired error threshold is achieved. The experimental results show that excellent performance can be achieved for typical SAR images with no significant distortion introduced by image compression.

1 Introduction

Efficient and reliable compression techniques for remote sensing imagery become more and more necessary as the number and size of images to be archived and transmitted over general purpose networks grow constantly. Even the most recent standards, such as JPEG2000, cannot always guarantee a good compression performance as they neglect the peculiar characteristics of the data. This is certainly true for SAR images, which are characterized by a wide dynamic range and are affected by a strong speckle that destroys the statistical regularities on which compression techniques rely.

In this paper, we propose a new compression algorithm of SAR images that fits different artificial neural networks (ANNs) models to different segmented image blocks by utilizing the theory of projection pursuit neural networks (PPNNs) [1],[2],[3]. Our approach is based upon the statistical ideas of projection pursuit regression (PPR) [4] [5]. Projection pursuit regression introduces a regression family that can be modeled as a sum of subnets of a single hidden layer neural network. The parameters of the subnets can be determined sequentially following the strategy of a so called greedy algorithm.

The organization of this paper is as follows. Section 2 is devoted to the image segmentation. Section 3 presents the theory of the projection pursuit neural networks,

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 730–734, 2005.

and the quantization and coding of the PPNN parameters. Section 4 presents the results of experiments and conclusions.

2 SAR Images Segmentation

In the first stage, an initial quadtree segmentation of the input SAR image is performed to locate large regions having homogeneous mean values. The concept of a quadtree is quite simple. It involves the construction of a quadtree structure (each node has either no children or four children) in which each leaf represents a homogeneous region [6],[7]. The definition of "homogeneous" in this context depends upon the type of compression which is desired. In the lossless compression of a binary image, for instance, a homogeneous region would be defined as a region that is either all back or all white.

There are two ways in which quadtree construction can be done top-down or bottom-up. The more efficient of the two, bottom-up, considers a "complete" quadtree structure for the image (leaves represent individual pixels). If any four "neighbor" pixels meet an appropriate merging criterion, their corresponding nodes in the quadtree are eliminated, and a resulting average value is stored in the parent node which becomes a leaf. A typical segmentation map of the test SAR image in Fig. 3(a) is shown in Fig. 1.

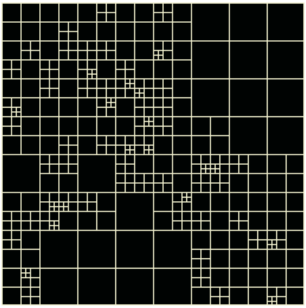


Fig. 1. Quadtree segmentation map of the test SAR image (Figure 3(a))

3 Projection Pursuit Neural Networks Image Coding

Once the image has been segmented into various size regions, each block can be coded using the PPR image approximation technique. PPR image approximation technique was developed in 1981 by Friedman and Stuezle [4]. It is a nonlinear regression approach which can be represented by the following equation:

$$t = \sum_{i=1}^{K} s_i(w_i \cdot \underline{x}) + \varepsilon \ . \tag{1}$$

Where ε is a noise random variable independent of the input variables in $\underline{x}$. Functions $\{s_i\}$ can be approximated by small single hidden layer networks

$$s_i(z) \approx \sum_{m=1+j_i}^{j_{i+1}} a_m g_m(z-\tau_m), j_1 = 0, j_{K+1} = J. \tag{2}$$

where $\{g_i\}$ are the ridge functions that are any arbitrary smooth function to be learned from the data. In this way the PPR approximation becomes a single hidden layer network

$$y = \sum_1^J a_j g_j(w_j \cdot \underline{x} - r_j) \ . \tag{3}$$

By using ideas from PPR that have us determine, in greedy algorithm fashion, each block individually, we are able to quickly estimate a moderately sized single hidden layer network that can well approximate the block image.

We need to determine iteratively the subnet functions $\{s_i\}$ and the "ridge" directions $\{w_i\}$. Each subnet s_i is implemented by a small single-hidden layer net. We rewrite the error $t-y$ in terms of successive residuals

$$z_0(\underline{x},t) = t, z_i(\underline{x},t) = z_{i-1} - s_i(w_i \cdot \underline{x}) \qquad for \quad i = 1 \cdots K \ . \tag{4}$$

$$z_K(\underline{x},t) = t - y \ . \tag{5}$$

Fig. 2 shows a network of this structure with three subnets that illustrates (2)–(5) for $K = 3$. The first subnet s_1 has only one node in its hidden layer while the other two subnets have two nodes each. The point of this rewrite is that we are adopting the view of sequentially selecting the subnets s_i with reducing the residual error z_i. For example, approximating the block image using only the single simple subnet $s_1(w_1 \cdot \underline{x})$, is likely to yield a large residual error $z_1 = t - s_1$. We then iterate for K steps to reduce this residual error. This is an example of a so-called greedy algorithm; at the ith step we choose the best subnet s_i to reduce that residual error z_i without regard to whether this choice is optimal with respect to the collection of all K choices. The process was terminated when the overall error dropped below the desired threshold or if the desired bit rate was achieved.

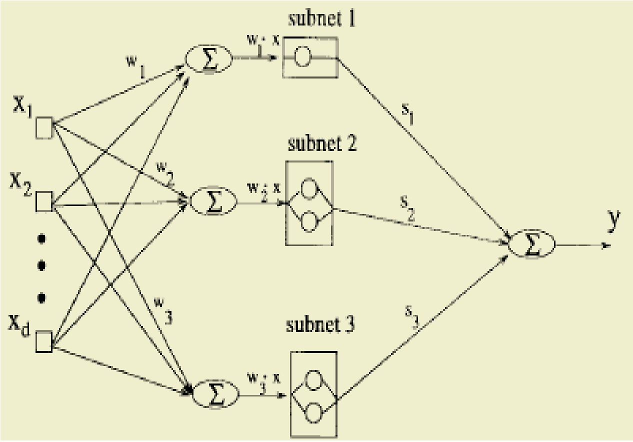
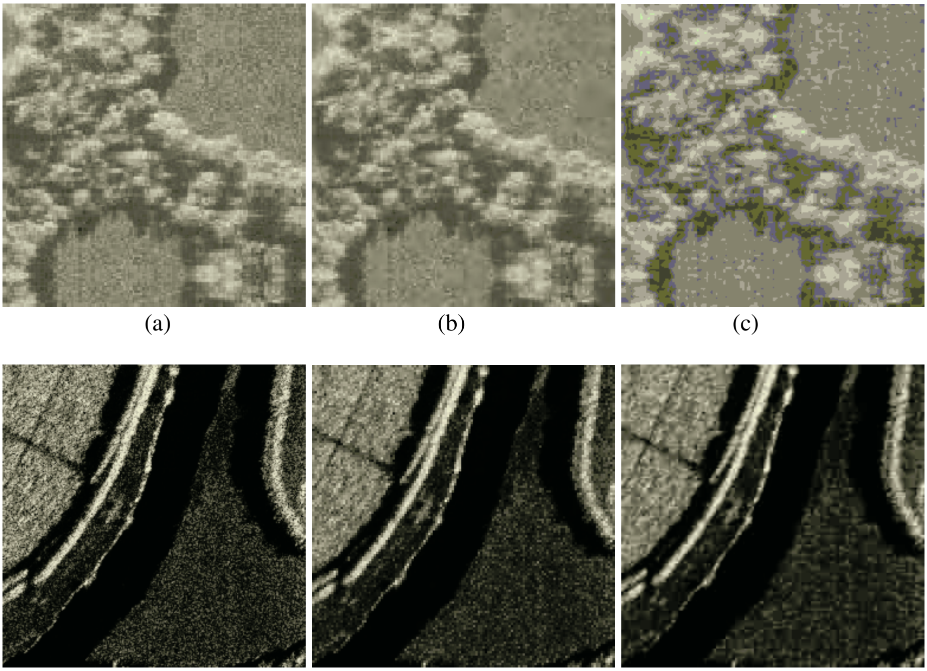


Fig. 2. A five hidden node networks consisting of three subnets

The optimum parameters for each block must be quantized and encoding. We quantized parameters at each iteration of the algorithm. By quantizing the coefficients at each iteration we force the algorithm to correct the quantization errors at each iteration and improving the resulting PSNR.

4 Experimental Results and Conclusions

Simulations were carried out to demonstrate the effectiveness of the proposed algorithm. The test SAR images (256x256) are shown in Fig. 3(a) and in Fig. 3(d). The segmentation map, which can be represented as a quadtree, was generated by first dividing the test SAR image into blocks of 32x32. These blocks were further subdivided into blocks of 16x16, 8x8 and 4x4. The segmentation map for our first SAR image in Fig. 3(a) is shown in Fig. 1.



(a) (b) (c)

(d) (e) (f)

Fig. 3. The performance of our coding algorithm on test SAR images: (a) The original SAR test image (256x256), (b) Decoded SAR image (a); our method, at 0.073 bit/pixel, PSNR =29.3 dB, (c) Decoded SAR image (a); JPEG, at 0.073 bit/pixel, PSNR = 28.5 dB, (d) The original SAR test image (256x256), (e) Decoded SAR image (d); our method, at 0.1 bit/pixel, PSNR= 21.8 dB, (f) Decoded SAR image (d); JPEG, at 0.1 bit/pixel, PSNR = 21.5 dB

Once the SAR image was segmented, the PPNN model for each block was constructed by the following procedure. Every function in a set of fixed pre-determined basis functions called the Universal Basic Function Set (UBFS) was used to obtain the best PPNN model for each block by optimizing the parameters of the function in the mean square error sense. The optimal network parameters were selected by performing a greedy search over all the functions in the UBFS. This provided the first level approximation for each block. The next levels of approximations were obtained by repeating the above process on the residual errors. The process was terminated when the overall error dropped below the desired threshold. In every iteration of this algorithm, from UBFS, a function is selected that best approximates the current image in the given block. In the first step of the iteration, the current image is the original image, and in the step k the current image is the residual image that results from subtraction of the original image and linear combination of all the previous approximations. The UBFS considered in our experiments included the B-splines of up to order three. Finally, the quantized parameters for each block were separately encoded to form the compressed bit stream. The decoded test SAR image in Fig. 3(a) at the bit rate of 0.073 bit/pixel with PSNR of 29.3 dB is shown in Figure 3(b). For comparison, the JPEG algorithm was used to encode SAR image at the same bit rate of 0.073 bit/pixel. The PSNR of the corresponding JPEG decoded image was found to be 28.5 dB. The results for the test SAR image in Fig. 3(d) with the same UBFS are shown in Figures 3(e) and 3(f). Our experimental results showed that at low rates, our method is superior to JPEG both in terms of PSNR and the visual image quality.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (60375003), Aeronautics and Astronautics Basal Science Foundation of China (03I53059) and the Doctorate Foundation of Northwestern Polytechnical University (CX200327).

References

1. Safavian, S.R., Rabiee, H.R., Fardanesh, M.: Projection Pursuit Image Compression with Variable Block Size Segmentation. Signal Processing Letters, IEEE , **4** (1997) 117-120
2. Dotan, Y., Intrator, N.: Multimodality Exploration by an Unsupervised Projection Pursuit Neural Network. IEEE Transactions on Neural Networks, **9** (1998) 464–472
3. Jen-Lun Yuan, Fine, T.L.: Neural-network Design for Small Training Sets of High Dimension. IEEE Transactions on Neural Networks, **9** (1998) 266–280
4. Friedman, J.H., Stuetzle, W.: Projection Pursuit Regression. JASA, **76** (1981) 817-823
5. P. Huber: Projection Pursuit. Ann. Stat., **13** (1985) 435–525.
6. Vaisey J., Gersho A.: Image Compression with Variable Block Size Segmentation. IEEE Trans. Signal Processing, **40** (1992) 2040–2060
7. Burlet, J., Aycard, O., Fraichard, T.: Robust Motion Planning Using Markov Decision Processes and Quadtree Decomposition. Proceedings of IEEE International Conference on Robotics and Automation 2004, **3** (2004) 2820-2825

Image Restoration Using Hopfield Neural Network Based on Total Variational Model

Hongying Zhang[1], Yadong Wu[2], and Qicong Peng[1]

[1] School of Communication and Information Engineering
University of Electronic Science and Technology of China
Chengdu, Sichuan 610054, China
zhy0838@163.com
[2] School of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, Sichuan 610054, China
wyd028@163.com

Abstract. The relations between Total Variational (TV) image restoration model and Hopfield neural network are deduced by using energy function. Then a novel algorithm realizing TV image restoration using Hopfield neural network is given. Because of the advantages of neural network techniques such as the abilities of parallel computing and error-tolerance, it can improve the quality of restored image efficiently. Experimental result shows that the performance of the proposed numerical method is perfect.

1 Introduction

Image restoration is one of the very essential and important topics in digital image processing. So far, many methods have been proposed. Recently, a novel image processing technology based on variation and partial differential equation (PDE) is motivating many mathematicians and image scholars to research on this challenging topic. Among the variational image restoration models, Total Variational (TV) model proposed by Rudin, Osher and Faterni [1] is very important and has been applied widely. It is very efficient when the noise and blur models are known. It recently won a benchmark in satellite image deblurring organized by the French Space Agency (CNES) [2].

Generally, digital implementation schemes of restoration models may affect the quality of restored image and the efficiency of the algorithm. In current literatures [1], [3], [4], [5], [6], most of the digital implementation schemes about TV image restoration are translating TV model into corresponding two order partial differential equation (PDE), then using the steepest descent algorithm or direct difference method to solve the PDEs. With these methods, we can restore the image efficiently, but the precision of restored image is greatly affected by practical numerical methods.

Neural network (NN) research has gone a long way in the past decades. The specialists in the image processing field have fully understood the advantages of neural network techniques such as the abilities of parallel computing, nonlinear mapping and self-adaptiveness, and applied a variety of neural network models into the image proc-

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 735–740, 2005.

essing field. Among the various neural networks, Hopfield's model is suitable for solving optimization problems, such as the restoration problem [7], [8]. So, this paper will deduce the relations between TV image restoration model and Hopfield neural network by using energy function, and give an algorithm to realize TV image restoration using Hopfield neural network. At last, Experimental results show that the performance of the proposed numerical method is perfect.

2 Relations Between TV Restoration Model and Hopfield Neural Network

One popular and useful image degraded model widely used in astronomic and medical imaging processes is blurring followed by noising. The image-degraded model is given by [1]:

$$u_0(x,y)=h(x,y)*u(x,y)+n(x,y)\ . \tag{1}$$

where $u(x,y)$is the original image, $u_0(x,y)$is the observed image, i.e. blurred image, $h(x,y)$is the point spread function(PSF), and$n(x,y)$ is the additive noise due to the imaging system. The problem of image restoration is to recover an unknown original image$u(x,y)$ from a given blurred image $u_0(x,y)$and the PSF $h(x,y)$. The Bayesian restoration energy function proposed by Rudin, Osher, and Fatemi [1] is

$$E_{TV}[u]=\iint\limits_{\Omega}|\nabla u|\,dxdy+\frac{\lambda}{2}\iint\limits_{\Omega}(h*u-u_0)^2dxdy\ . \tag{2}$$

This is the famous TV image restoration model. In the following, we will deduce the relations between TV restoration model and Hopfield neural network.

Discretizing TV restoration model (2), we have

$$E_{TV}=\sum_x\sum_y\sqrt{\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial u}{\partial y}\right)^2}+\frac{\lambda}{2}\sum_x\sum_y\left(h*u-u^0\right)^2\ . \tag{3}$$

Usually, we can replace $\sqrt{\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial u}{\partial y}\right)^2}$ with $\left|\frac{\partial u}{\partial x}\right|+\left|\frac{\partial u}{\partial y}\right|$ for numerical implementation. Because the Sobel operator performs a 2-D spatial gradient measurement on an image and emphasizes regions of high spatial gradient that correspond to edges. Typically, the Sobel operator is applied to find the approximate absolute gradient magnitude at each point in an input image. So we can see the gradient magnitude as the Sobel operator. It is well known that the masks of the Sobel operator are:

$$d_x=\begin{bmatrix}-1&0&1\\-2&0&2\\-1&0&1\end{bmatrix},\quad d_y=\begin{bmatrix}1&2&1\\0&0&0\\-1&-2&-1\end{bmatrix}.$$

Based on the above reasons, we represent the Toeplitz matrix generated by $h(x,y)$ as H , a $L\times L$ matrix, $L=MN$,the image size is N by M , the Toeplitz matrix generated by d_x, d_y as D_x, D_y , $L\times L$ matrix too. Then, rewrite the equation(3) by the form of column vector, we can get:

$$E_{TV}=\sum_{p=1}^{L}\left(\sum_{i=1}^{L}D_{x_{pi}}u_i+\sum_{i=1}^{L}D_{y_{pi}}u_i\right)+\frac{\lambda}{2}\sum_{p=1}^{L}\left(u_{0p}-\sum_{i=1}^{L}H_{pi}u_i\right)^2$$

$$=\frac{\lambda}{2}\sum_{i=1}^{L}\sum_{j=1}^{L}\left(\sum_{p=1}^{L}H_{pi}H_{pj}\right)u_iu_j+\sum_{i=1}^{L}\left(\sum_{p=1}^{L}D_{x_{pi}}+\sum_{p=1}^{L}D_{y_{pi}}-\lambda\sum_{p=1}^{L}u_{0p}H_{pi}\right)u_i+\frac{\lambda}{2}\sum_{p=1}^{L}\left(u_{0p}\right)^2 \tag{4}$$

On the other hand, the neural model parameters, the interconnection strengths, and bias inputs can be determined in terms of the energy function of the neural network. As defined in [8], the energy function of the Hopfield neural network can be written

$$E_{hs}(t)=-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m}w_{ij}v_i(t)v_j(t)-\sum_{i=1}^{n}b_iv_i(t)\ . \tag{5}$$

By comparing the terms in (4) to the corresponding terms in (5) and ignoring the constant term $\frac{\lambda}{2}\sum_{p=1}^{L}\left(u_{0p}\right)^2$, we can determine the interconnection strengths and bias inputs as

$$w_{ij}=-\lambda\sum_{p=1}^{L}H_{pi}H_{pj}\ . \tag{6}$$

and

$$b_i=\lambda\sum_{p=1}^{L}u_{0p}H_{pi}-\left(\sum_{p=1}^{L}D_{x_{pi}}+\sum_{p=1}^{L}D_{y_{pi}}\right). \tag{7}$$

where H_{ij} , $D_{x_{ij}}$ and $D_{y_{ij}}$ are the elements of the matrices H , D_x and D_y ,respectively. So the equation (4) can be written by

$$E_{TV}=-\frac{1}{2}\sum_{i=1}^{L}\sum_{j=1}^{L}w_{ij}u_iu_j-\sum_{i=1}^{L}b_iu_i\ . \tag{8}$$

Through the above deduce, we can get the conclusion that TV restoration model and Hopfield neural network are interrelated. In addition, from (6), one can see that the interconnection strengths are only determined by the shift-invariant blur function and constant λ . Hence, w_{ij} can be computed without error provided the blur function is known. However, the bias inputs are functions of the observed degraded image u_0 . If the image is degraded by a shift-invariant blur function only, then b_i can be estimated perfectly. Otherwise, b_i is affected by noise.

3 Numerical Algorithm and Simulation Results

According to the above analysis, we can use Hopfield neural network to realize the image restoration based on TV model. In this algorithm, we suppose that the PSF is given, and noise is absent. The network update rules used in our algorithm are referred to [8]. So our algorithm has the following form:

(a) Generating the Toeplitz matrix H , D_x, D_y using the PSF and the Sobel operator, respectively;
(b) Computing the interconnection strengths w_{ij} and bias inputs b_i using the equations (6) and (7), respectively;

```
(c) While iter<Number defined and $u^{\mathrm{n+1}} \neq u^{\mathrm{n}}$
        for $i = 1 : L$
            $State_i = b_i + \sum_{\mathrm{j=1}}^{L} w_{ij} u_j^{\mathrm{n}}$
            $\Delta u_i = \mathrm{TVNNR}G(State_i)$
            where $\mathrm{TVNNR}G(State_i) = \begin{cases} -1, & State_i < -\theta_i \\ 0, & -\theta_i \le State_i \le \theta_i \ , \theta_i = -\frac{1}{2} w_{ii} > 0 \\ 1, & State_i > \theta_i \end{cases}$
            $u_i^{\mathrm{n+1}} = \mathrm{TVNNR}K\left(u_i^{\mathrm{n}} + \Delta u_i\right)$
            where $\mathrm{TVNNR}K(v) = \begin{cases} 0, & v<0 \\ v, & 0 \le v \le S \ , S \text{ is maximum grayvalue} \\ S, & v > S \end{cases}$
        end
        $u^{\mathrm{n}} = u^{\mathrm{n+1}}$
    end
```

3.1 Convergence Analysis

According to (8), the energy function of neural model, based on total variation image restoration model, is given by:

$$E_{TV} = -\frac{1}{2} \sum_{i=1}^{L} \sum_{j=1}^{L} w_{ij} u_i u_j - \sum_{i=1}^{L} b_i u_i \ . \tag{9}$$

In the above algorithm, the energy of network is updated when each neuron state changing. In order to measure the change of (9), we consider the change of k th neuron, and rewrite (9) as followings:

$$\begin{aligned}
E_{TV} &= -\frac{1}{2}\sum_{i=1}^{L} u_i \left(\sum_{\substack{j=1 \\ j\neq k}}^{L} w_{ij} u_j \right) - \frac{1}{2}\sum_{i=1}^{L} u_i w_{ik} u_k - \sum_{\substack{i=1 \\ i\neq k}}^{L} b_i u_i - b_k u_k \\
&= -\frac{1}{2}\sum_{\substack{i=1 \\ i\neq k}}^{L} u_i \left(\sum_{\substack{j=1 \\ j\neq k}}^{L} w_{ij} u_j \right) - \frac{1}{2} u_k \sum_{\substack{j=1 \\ j\neq k}}^{L} w_{ij} u_j - \frac{1}{2}\sum_{i=1}^{L} u_i w_{ik} u_k - \sum_{\substack{i=1 \\ i\neq k}}^{L} b_i u_i - b_k u_k \qquad (10) \\
&= -\frac{1}{2}\sum_{\substack{i=1 \\ i\neq k}}^{L} \sum_{\substack{j=1 \\ j\neq k}}^{L} u_i w_{ij} u_j - \sum_{\substack{i=1 \\ i\neq k}}^{L} b_i u_i - \frac{1}{2} u_k \sum_{\substack{i=1 \\ i\neq k}}^{L} u_i w_{ik} - \frac{1}{2} u_k \sum_{\substack{j=1 \\ j\neq k}}^{L} w_{kj} u_j - \frac{1}{2} w_{kk} u_k^{\ 2} - b_k u_k
\end{aligned}$$

According to update rule in the algorithm above, new state of k th neuron is $u_k^{\mathrm{n}+1} = u_k^{\mathrm{n}} + \Delta u_k$. So the change of energy of network is given by:

$$\begin{aligned}
\Delta E_{TV} &= E_{TV}(n+1) - E_{TV}(n) \\
&= -\frac{1}{2}\Delta u_k \sum_{\substack{i=1 \\ i\neq k}}^{L} u_i w_{ik} - \frac{1}{2}\Delta u_k \sum_{\substack{j=1 \\ j\neq k}}^{L} w_{kj} u_j - \frac{1}{2} w_{kk} (2u_k \Delta u_k + \Delta u_k^{\ 2}) - b_k \Delta u_k \quad . \qquad (11) \\
&= -\frac{1}{2}\Delta u_k \sum_{\substack{i=1 \\ i\neq k}}^{L} u_i w_{ik} - \frac{1}{2}\Delta u_k \sum_{\substack{j=1 \\ j\neq k}}^{L} w_{kj} u_j - u_k w_{kk} \Delta u_k - \frac{1}{2} w_{kk} \Delta u_k^{\ 2} - b_k \Delta u_k
\end{aligned}$$

Because $W = W(w_{ij})$ is a symmetry matrix, and $State_k = b_k + \sum_{\mathrm{j}=1}^{L} w_{kj} u_j^{\ \mathrm{n}}$, so we get

$$\Delta E_{TV} = -\frac{1}{2} w_{kk} (\Delta u_k)^2 - State_k \Delta u_k \quad . \qquad (12)$$

Because w_{kk} is negative, the above formula (12) is the same with the Paik's sequential algorithm, which is detailed analyzed in [8]. According to the update rule of our algorithm, the energy will descend to a fixed point (local minimum) after a finite number of iterations.

3.2 Simulation Results

In this simulation, we choose the numbers of iteration are 200 and $\lambda = 1$. The original image and the distorted image without noise are shown in Fig. 1(a), (d) and (b), (e), respectively. The corresponding restored images are shown in Fig. 1 (c) and (f). From these figures, one can see that the proposed numerical method is very efficient to restore the blurred images.

4 Conclusions

This paper has introduced a new approach for realizing the TV image restoration of gray level images degraded by a shift-invariant blur function using Hopfield neural

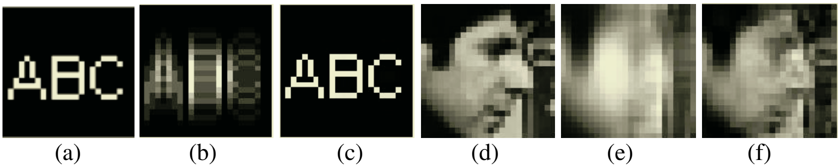


Fig. 1. Image restoration for 3×3 motion blur without noise. (a) and (d) Original image; (b) and (e) Image distorted by 3×3 motion without noise; (c) and (f) Restored image

network. First, with energy function we deduce the relations between TV image restoration model and Hopfield neural network. Then a novel algorithm realizing TV image restoration using Hopfield neural network is given. At last, we explain the convergence of the proposed numerical algorithm and give the simulation results to show the efficiency of this new method.

References

1. Rudin, L., Osher, S., Faterni, E.: Nonlinear Total Variation Based Noise Removal Algorithms. Physica D, **60** (1992) 259-268
2. Guichard, F., Moisan, L., Morel, J. M.: A Review of PDE Models in Image Processing and Image Analysis. Journal de Physique IV, France, **12** (2002) 1-18
3. Chan, T.F., Shen, J.H.: A Good Image Model Eases Restoration. Available at http://www.ima.umn.edu/preprints/feb02/1829.pdf
4. Vogel, C.R., Oman, M.E.: Iterative Methods For Total Variation Denoising. SIAM Journal on Scientific Computing, **17** (1996) 227 - 238
5. Vogel, C.R., Oman, M.E.: Fast, Robust Total Variation Based Reconstruction of Noisy, Blurred Images. IEEE Trans. Image Processing , **7** (1998) 813 - 824
6. Bing Song: Topics in Variational PDE Image Segmentation, Inpainting and Denoising. Ph.D. Thesis, University of California Los Angeles (2003)
7. Zho, Y.T.u, Chellappa, R.: Image Restoration Using a Neural Network. IEEE Trans. Acoust., Speech, Signal Processing, **36** (1988) 1141-1151
8. Paik, J.K., Katsaggelos, A.K.: Image Restoration Using a Modified Hopfield Network. IEEE Trans. Image Processing , **1** (1992) 49-63

Pulse Coupled Neural Network Based Image Fusion

Min Li[1,2], Wei Cai[2], and Zheng Tan[1]

[1] School of Electronics and Information Engineering, Xi'an Jiaotong University
Xi'an, Shaanxi 710049, China
limin@mailst.xjtu.edu.cn
[2] Xi'an Research Inst. of Hi-Tech Hongqing Town, Shaanxi 710025, China
clwn@163.com

Abstract. For the pulse-coupled neural network (PCNN) has an inherent ability to segment images, we present a multisensor image fusion scheme based on PCNN in this paper. The algorithm adopts salience and visibility as two extracted features for each segmented region to determine the fusion weight. Extensive experimental results have demonstrated that the proposed method has extensive application scope and it outperforms the discrete wavelet transform approach, both in visual effect and objective evaluation criteria, particularly when there is movement in the objects or mis-registration of the source images.

1 Introduction

Being an efficient method of information fusion, image fusion has been used in many fields such as machine vision, medical diagnosis, military applications and remote sensing [1].

The simplest image fusion method just takes the pixel-by-pixel average of the source images. In recent years, a various alternatives based on multiscale transforms have been proposed [2]. Examples of this approach include the Laplacian pyramid, the gradient pyramid, the ratio-of-low-pass pyramid and the morphological pyramid. More recently, the discrete wavelet transform (DWT) has also been used. In general, DWT is superior to the previous pyramid-based methods [2].

While these methods often perform satisfactorily, their multiresolution decompositions and the consequent fusion results are shift-variant because of an underlying down-sampling process. When there is slight movement or mis-registration of the source images, their performance will thus deteriorate quickly. One possible remedy is to use the shift-invariant discrete wavelet frame transform. However, the implementation is more complicated and the algorithm is also more demanding in terms of both memory and time.

For PCNN has great advantage on image segmentation [3], we propose an image fusion scheme based on PCNN image segmentation. The implementation of the algorithm is computationally simple and can be realized in real-time. Experimental results show that the proposed method outperforms the discrete wavelet transform approach, particularly when there is movement in the objects or mis-registration of the source images.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 741–746, 2005.

2 PCNN Image Segmentation Based Multi-sensor Image Fusion

Fig.1 shows a schematic diagram of the proposed scheme.

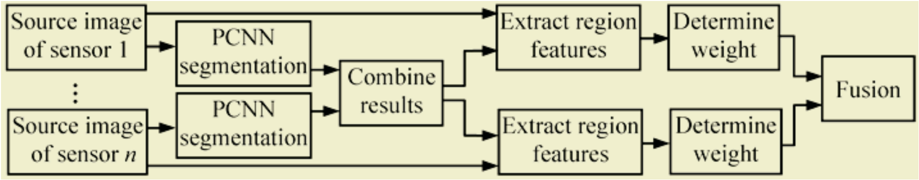


Fig. 1. Schematic diagram of the proposed fusion method

In detail, the algorithm consists of the following steps:

Step1. Segment the registered source images into different regions by PCNN (details in Section 2.1).

Step2. Combine the segmentation results with source images to determine the region each pixel belongs to. Suppose the multisensor source images are $S_1, S_2, \ldots, S_n$, denote the *i*th region of image S_m (m=1, 2, …, n) by $DB_i(S_m)$.

Step3. From each image region $DB_i(S_m)$, extract two features salience(*SA*) and visibility(*VI*) that reflect its clarity. Denote the feature vectors for $DB_i(S_m)$ by ($SA_{DBi(Sm)}$, $VI_{DBi(Sm)}$) (details in Section 2.2).

Step4. Determine the fusion weight of $DB_i(S_m)$ according to ($SA_{DBi(Sm)}$, $VI_{DBi(Sm)}$). Denote the fusion weight for $DB_i(S_m)$ by $W_{DBi}(S_m)$ (details in Section 2.3).

Step5. Get the fusion image *F* by $DB_i(S_m)$ and $W_{DBi}(S_m)$.

2.1 PCNN Based Image Segmentation

The PCNN model is a system composed of closely interacting nodes, with spiking neural behavior. It finds many applications in image processing, including segmentation, edge extraction and so on [3].

Each PCNN neuron is divided into three compartments with characteristics of the receptive field, the modulation field, and the pulse generator (see Fig. 2).

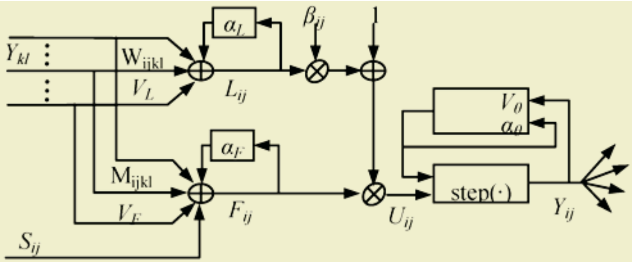


Fig. 2. PCNN neuron model

The receptive field is comprised of feeding field and linking field. In the feeding field and linking field, there are six parameters, i.e., three time decay constants (α_F, α_L, α_θ) and three amplification factors (V_F, V_L, V_θ).

The following five equations are satisfied.

$$F_{ij}(n) = \exp(-\alpha_F) \cdot F_{ij}(n-1) + S_{ij} + V_F \cdot \sum M_{ijkl} Y_{kl}(n-1) \tag{1}$$

$$L_{ij}(n) = \exp(-\alpha_L) \cdot L_{ij}(n-1) + V_L \sum W_{ijkl} Y_{kl}(n-1) \tag{2}$$

$$U_{ij}(n) = F_{ij}(n)(1 + \beta \cdot L_{ij}(n)) \tag{3}$$

$$\theta_{ij}(n) = \exp(-\alpha_\theta)\theta_{ij}(n-1) + V_\theta Y_{ij}(n-1) \tag{4}$$

$$Y_{ij}(n) = step(U_{ij}(n) - \theta_{ij}(n)) \tag{5}$$

Where, *M* and *W* are the linking matrix, and normally *W=M*, β is the linking coefficient, *step*(•) is the unit step function.

In the application of image segmentation, each pixel corresponds to a single PCNN neuron. That is, a two dimensional intensity image (M×N) can be thought as a PCNN neuromime with M×N neurons, and the gray level of pixels can be thought as S_{ij}, the input of the neuron. The neurons are organized in a single layer network to perform the segmentation task. When there are pixels whose gray levels are approximate in the neighborhood of *M* and *W*, one pixel's pulsating output can activate other corresponding pixels having the approximate gray level in the neighborhood and let them generate pulsating output sequence *Y(n)*. Obviously *Y* contains some information about this image such as regional information, edge, and texture features. Then the binary image constructed by *Y(n)*, the output of PCNN, is the segmented image. This is why the PCNN achieved the image segmentation.

To satisfy the demand of the following fusion process, all segmentation information in different source images is combined together. A simple way is to draw the contours for all segmented images, and then overlap all contour images together; thus each image can be divided into different regions.

Considering the lonely points have little contributions to image quality and probably belong to noise, we should get rid of them after the segmentation by PCNN.

2.2 Feature Extraction

In this paper, we extract two features from each image region to represent its clarity, the salience and visibility.

Salience(SA). The salience proposed in this paper is mainly with a view to the difference between target and its neighboring regions. Assume $DB_i(S_m)$ is *i*th region of the image S_m, how to determine the salience of DB_i is given as following.

The edge of DB_i is determined at first. Assume p is any point of DB_i's edge and N_8 (p) is its eight neighboring pixels, we expand the edge by N_8 (p) to get an edge band about three pixels width. Take the edge band into account, we can work out the mean gray value of those pixels belong to DB_i and those not. The absolute difference is defined as the salience of DB_i.

$$SA(\mathrm{DB_i}) = abs(\underset{q\in \mathrm{DB_i}}{AVER(q)} - \underset{q\notin \mathrm{DB_i}}{AVER(q)}) \tag{6}$$

where q is point in edge band, $\underset{q\in \mathrm{DB_i}}{AVER(q)}$ is the mean gray value of pixels which belong to DB_i, $\underset{q\notin \mathrm{DB_i}}{AVER(q)}$ is the mean gray value of pixels not belong to DB_i.

Visibility(VI). This feature is inspired from the human visual system, and is defined as [4]. We rectify the formula as following.

$$VI(DB_i) = \frac{1}{sum(DB_i)} \sum_{(x,y)\in DB_i} \left(\frac{1}{m_k}\right)^{\alpha} \cdot \frac{|f(x,y) - m_k|}{m_k} \tag{7}$$

where $sum(DB_i)$ is the total number of pixels in DB_i, m_k is the mean gray value of the image region, and α is a visual constant ranging from 0.6 to 0.7.

2.3 Fusion Weight of the Region

Considering the different contributions to the fusion result between various source images, we use region fusion weight to denote these. Assume $W_{DBi}(S_m)$ is the fusion weight of region $DB_i(S_m)$, we use visibility and salience of the region as the two main factors to determine it. Based on extensive experiments, we define it as

$$W(DB_i) = \exp(VI(DB_i) \times W_{VI}) + \exp(SA(DB_i)) \tag{8}$$

where W_{VI} is a visibility constant.

3 Experiments

To evaluate the performance of the proposed fusion method, extensive experiments with multi-focus images fusion and different sensor images fusion have been made. Here, we give a typical experiment.

Three objective evaluative criteria are used. They are the root mean squared error (RMSE), cross entropy (CEN) and spatial frequency (SF) [5]. Assume R is reference image and F is the fused image, the definitions are given as follows

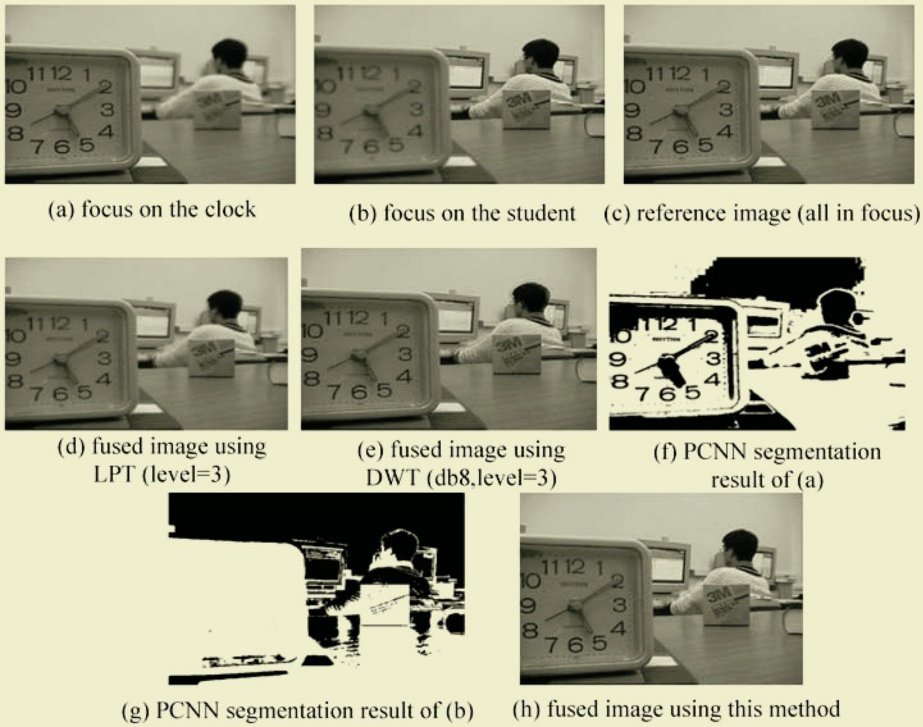
$$RMSE = \sqrt{\frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} [R(i,j) - F(i,j)]^2} \tag{9}$$

$$SF = \sqrt{\frac{1}{MN}\left[\sum_{i=0}^{M-1}\sum_{j=1}^{N-1}[F(i,j)-F(i,j-1)]^2 + \sum_{j=0}^{N-1}\sum_{i=1}^{M-1}[F(i,j)-F(i-1,j)]^2\right]} \tag{10}$$

$$CEN(P:Q) = \sum_{g=0}^{L} p(g)\log_2 \frac{p(g)}{q(g)} \tag{11}$$

For comparison purposes, besides the fusion scheme proposed in this paper, another two fusion algorithms, Laplacian pyramid based (LPT) and wavelet decomposition based (DWT)[2], are applied to fuse the same images.

The typical experiment is performed on two sets of multifocus source images in Fig.3 (there is slight movement of the student's head in Fig.3(a) and Fig.3(b)). Fig.3(c) is all in focus reference images. Quantitative comparison of their performance is shown in Table 1. In comparison, we can see the fusion method presented in this paper show significant improvement over the DWT-based method. In particular, notice that Fig.3(a) and Fig.3(b) are mis-registration, this method outperforms the DWT-based approach at this situation very well. By the way, for perfectly registered images, it shows more significant improvement over the DWT-based method.



(a) focus on the clock (b) focus on the student (c) reference image (all in focus)

(d) fused image using LPT (level=3) (e) fused image using DWT (db8,level=3) (f) PCNN segmentation result of (a)

(g) PCNN segmentation result of (b) (h) fused image using this method

Fig. 3. The multifocus source images (256-level, size of 640×480) and fusion results. Parameters in PCNN as: CT: β=0.3,α_F=1, α_L=1, α_θ=2.5(1), V_F=10, V_L=10, V_θ=100,r=1,N=2

Table 1. Performance of the different fusion methods on processing Fig.3

Fusion Method	LPT	DWT	This method
RMSE	7.1126	4.6622	2.9719
SF	7.7312	12.8760	12.9161
CEN	0.3642	0.2273	0.0112

4 Conclusions

In this paper, we combine the idea of artificial neural networks with image fusion, and propose an image fusion scheme based on PCNN. Lots of experiments on studying the fusion performance have been made and the results show that the proposed method has extensive application scope and it outperforms the discrete wavelet transform approach, both in visual effect and objective evaluation criteria, particularly when there is movement in the objects or mis-registration of the source images. Future work will deal with finding a self-adaptive algorithm on adjusting the parameters of the PCNN.

References

1. Hall, D. L., Llinas, J.: An Introduction to Multisensor Data Fusion. Proceedings of IEEE, **85** (1997) 6-23
2. Zhang, Z., Blum, R.S.: A Categorization of Multiscale-Decomposition-Based Image Fusion Schemes with a Performance Study for a Digital Camera Application. Proceedings of IEEE, **87** (1999) 1315-1326
3. Johnson, J.L., Padgett, M.L.: PCNN Model and Applications. IEEE Trans. on Neural Networks, **10** (1999)480-498
4. Huang, J.W. Shi, Y.Q., Dai, X.H.: A Segmentation-based Image Coding Algorithm Using the Features of Human Vision System. Journal of Image and Graphics, 4 (1999) 400-404
5. Xydaes, C., Petrovi, V.: Objective Image Fusion Performance Measure. Electronic Letters, **36** (2000) 308-309

A Novel Image Fusion Method Based on SGNN

Zheng Qin[1], Fumin Bao[1], and Aiguo Li[2]

[1] Department of Computer Science, Xi'an Jiaotong University
Xi'an, Shaanxi, 710049, China
zhqin@mail.xjtu.edu.cn, bob0916@263.net

[2] Department of Computer Science, Xi'an University of Science and Technology
Xi'an, Shaanxi, 710054, China
liag@xust.edu.cn

Abstract. Multi-sensor image fusion is a challenging research field, which is a issue to be further investigated and studied. Self-Generating Neural Networks (SGNNs) are self-organization neural network, whose network structures and parameters need not to be set by users, and its learning process needs no iteration. An approach of image fusion using a SGNN is proposed in this paper. The approach consists of pre-processing of the images, clustering pixels using SGNN and fusing images using fussy logic algorithms. The approach has advantages of being wieldy to be used by users and having high computing efficiency, The experimental results demonstrate that the MSE (mean square error) of this approach decreases 30%-60% than those by Laplacian pyramid and discrete wavelet transform approaches.

1 Introduction

Recently, multi-sensor image fusion has been a hot spot in the research area of computer vision, automatic target recognition, robots and military applications[1],[2],[3].

Because neural network is especially applicable where there is no any suitable theoretical model or there is noise or nonlinearity, the neural network can simulate flexibly various nonlinear characteristics whose knowledge we need not learn in advance. In the past years, the neural network has been used in the classification of remote multi-sensor image information. Yiyao *et al*[4] proposed a method that fuses edge maps of original images using a knowledge-based neural network. They use a BP network with two implied layers for fusion of an edge map whose edge information is more complete and reliable than that from any single original image. Zhang *et al*[5] presented an one-dimensional self-organizing neural network (SOM) based method to fuse image information. The method uses one-dimensional pixel feature quantity to perform directly two-dimension image signal processing. It completes fusion of multiple noisy maps and is effective to recognize objects in those maps. Researches mentioned above show that it is feasible to fuse image with the neural network techniques, but as the same as common neural network applications, the image fusion approaches based on the above neural networks request a human expert

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 747–752, 2005.

to set up their structure and select their parameters. It is quite tricky to choose the right network structure suitable for a particular application at hand. Concerning the design of the network structure, the following must be decided: (1)the number of the network layers; (2)the number of neurons of each layer and (3)the weights on connection between consequent layers. During the repeated learning, to find centrally the valuable parameters of the statistic network structure, the assigned connecting weight is continuously refreshed. Therefore it is the most important and difficult to get the correct network structure for every network, not only in the learning process, but also in the design process of the neural network.

Like SOM, the self-generating neural network (SGNN) is also a self-organizing neural network of distinct features which needs neither special network structure and network parameters and nor iterative learning. This paper attempts to apply the SGNN to image fusion processing to eliminate the deficiencies of exiting methods. We offer a new image fusion approach based on the SGNN, called SIF (SGNN Image Fusion), which features highly self-learning property and high computing speed and is applicable to the fusion of multiple frequency spectrum/multi-sensor images.

2 SGNN

The SGNN was first proposed by Wen *et al.* [6],[7],[8]. Then Inoue *et al*[9] made profound research in its applications. The SGNN is an extension of the self-organizing neural network (SONN). It features simple network design and short learning time. The traditional neural network is a human-designed network structure and adjusts the connecting weight through learning the sample. It is a skillful job to select a good network structure for the current learning task, because only at a special application, can a certain network structure give full play of its capacity. But the SGNN generates a self-generating neural tree (SGNT) in learning the sample and its whole structure including its neurons, connection between the neurons and weights is generated on its own in the non-supervisory learning method during learning, so it has good adaptation and is applicable to classification and cluster. For more detailed information of optimization, pruning and performances of the SGNT, see [6],[7],[8],[9]. In this paper the SGNN is used for clustering images at pixel level.

The following definitions are made before describing the generating algorithm of the SGNT.

1) The input sample vector e_i is the vector composed of its attributes, $e_i = (a_{i1}, a_{i2}, \cdots, a_{iq})$;

2) The neuron n_j is an ordered pair (w_j, c_j). w_j is weight vector of neuron n_j, $w_j = (w_{j1}, w_{j2}, \cdots, w_{jq})$; c_j is the aggregation of leaves n_j ;

3) The SGNT is a tree <{ n_j }, { l_k }> generated automatically from the training samples in the following algorithm, where { n_j } the aggregation of neurons, { l_k } is the connecting aggregation of the tree. And only when at $n_j \in c_i$, n_i and n_j has direct connection.

4) For an input sample e_i , if $\forall j$, $d(n_k, e_i) \leq d(n_j, e_i)$, then n_k is called winner. $d(n_j, e_i)$ is the Euclidean distance between the neuron n_j and the sample e_j .

3 Image Fusion Approach Based on SGNN

3.1 Pre-processing of Image

There may be differences between two images of the same scene captured by two sensors in contrasts and grey levels, and there may be noises. So it is possible to improve the final results by filtering out the noises before using the original images. The main difficulty is how to trade off the reducing of noise and the maintaining of the image details. It is possible to use the wavelet transform theory to reduce noises of an image while maintaining the details of image. In practice, a class of wavelet and starting sizes are selected to threshold the detail factors and decide the overall threshold. Then noise reduction of images can be performed.

3.2 Image Clustering

For an image of $n_1 \times n_2$ pixels, its pixels are used as the input sample set and its grey level as the attribute values. Thus, there is only one attribute value, that is, the grey level.

For the input sample set $e_i = (a_{i1}, a_{i2}, \cdots, a_{iq})$, where $q = 1$; i equals the amount of pixels of input image, namely, $i = n_1 n_2$. The pixels of image are input one by one into the SGNT generated algorithm, finally generating a SGNT which clusters automatically cluster pixels in their grey levels. The cluster is generated automatically and needs not any manually assigned amount.

In a SGNT, every sub-node of the root node represents a cluster center and its weight is the weight of the cluster center and its amount represents the amount of cluster. And the leaf nodes of the sub-tree rooted on the sub-nodes are the elements of relevant cluster. Thus every pixel must belong to a special cluster.

Use $p(m,n)$ to represent the grey level of the pixel whose position is (m,n). Assuming the pixels of the image are divided into k classes and the centre grey level of the j th ($j = 1,2 \cdots, k$) class is w_j , then the pixels belonging to the j class satisfy:

$$d(w_j, p(m,n)) = \min_{1 \leq j \leq k} \{d(w_j, p(m,n))\} \qquad m = 1,2,\cdots,n_1; n = 1,2,\cdots,n_2 \tag{1}$$

3.3 Fuzzification of Clustered Image

We introduce the fuzzed membership degrees to change the value range of μ from $\{0,1\}$ to [0,1], that is to say, the pixels will no longer belong to a special class but belong to a different class according to their degrees. We use triangular membership functions. Every pixel in the image clustered by the SGNN belongs to a special class. $\mu_j(p(m,n))$ represents the pixel whose position is (m,n), The subscript j of μ represents that this pixel belongs C_j class. The definitions of the triangular membership functions are Eq. (2) and Eq. (3) respectively.

$$\mu_j(p(m,n)) = \begin{cases} 0, & p(m,n) \notin [w_j, w_{j+1}] \\ -(p(m,n) - w_{j+1})/(w_{j+1} - w_j), & p(m,n) \in [w_j, w_{j+1}] \end{cases} \quad (2)$$
$$j = 1,2,\cdots,k; m = 1,2,\cdots,n_1; n = 1,2,\cdots,n_2$$

Where, w_j and W_{j+1} are the central weights (grey levels) of j th class and $j+1$ th class respectively.

$$\mu_{j+1}(p(m,n)) = 1 + (p(m,n) - w_{j+1})/(w_{j+1} - w_j) = 1 - \mu_j(p(m,n)) \quad (3)$$
$$m = 1,2,\cdots,n_1; n = 1,2,\cdots,n_2$$

Namely, when the grey level $p(m,n)$ of the pixel at (m,n) is between $[w_j, w_{j+1}]$, the memberships of this pixel to w_j and w_{j+1} can be got by substituting $p(m,n)$ into Eq. (2) and Eq. (3) respectively. Thus we get $\sum_{j=1}^{k} \mu_j(p(m,n)) = 1$.

3.4 Fusion

The fusion process of images is the process to make the fuzzy quantity, i.e. the membership degree more precise. Assuming M images of $n_1 \times n_2$ pixels are to be fused, $p_l(m,n)$ represents the grey level of l th image at (m,n), $\mu_j(p_l(m,n))$ is the membership degree of the pixel of l th image at (m,n) to the j th class, the membership degree of the pixel of the finally fused image F at the same position to the j class is got according to its maximum membership degree:

$$\mu_j(p_f(m,n)) = \max_{1 \le i \le M} \{\mu_j(p_i(m,n))\} \quad (4)$$
$$j = 1,\cdots,k; m = 1,2,\cdots,n_1; n = 1,2,\cdots,n_2$$

The grey level $p_f(m,n)$ of the fused image F at (m,n) is got in the weighed mean method:

$$p_f(m,n) = (\sum_{j=1}^{k} \mu_j(p_f(m,n)) w_j) / (\sum_{j=1}^{k} \mu_j(p_f(m,n))) \quad (5)$$
$$m = 1,2,\cdots,n_1; n = 1,2,\cdots,n_2$$

Value the grey level of every pixel one by one in Eq. (5) and finally get a fused image.

4 Experiments

In our experiment, three fusion approaches are used for comparison: the SIF approach of this paper, the Laplacian pyramid algorithm (four-level Laplacian pyramid with maximum high pass and mean low pass, called LAP in short) and the wavelet transform approach (four discrete wavelet transform with maximum high pass and mean low pass, called DWT in short). A grey-level image of 256 x 192 is used as the original image as shown in Fig. 1(a). The original image is polluted with Gaussian noise and multiplicative noise respectively, resulting in source images A (Fig. 1(b)) and B (Fig. 1(c)). Then the images A and B are fused by SIF, LAP and DWT, getting Figs.1 (d), (e) and (f). And the images A and B are noise eliminated with the wavelet transform, resulting in source images C (Fig. 1 (g)) and D (Fig. 1(h)). The image C and D are fused by SIF, LAP and DWT, resulting in Figs. 1 (i), (j) and (k). Table 1 lists the error analyses of fused images of images A and B, and images C and D by SIF, LAP and DWT respectively. In the table, MSE is defined as:

$$MSE = \frac{\sum_{m=1}^{n_1} \sum_{n=1}^{n_2} (z(m,n) - o(m,n))^2}{n_1 \times n_2} \tag{6}$$

Where, n_1, n_2 are the length and the width of image respectively; $z(m,n)$ is the grey level of the pixel of the image to be compared; and $o(m,n)$ is the grey level of the original image Fig. 1(a).

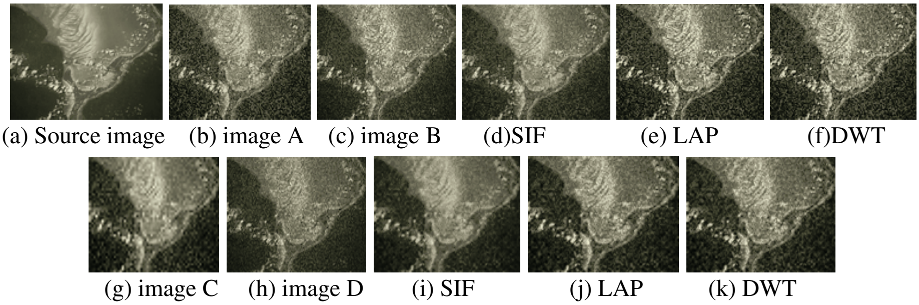
Table 1. MSE of fused images

Images	SIF	LAP	DWT
A, B	0.00899	0.02661	0.02798
C, D	0.00355	0.00556	0.00583

The fused images of source images A and B show that before pre-processing by wavelet transform, the SIF approach has 66% and 68% lower MSE than that of the LAP and DWT approaches. After pre-processing by wavelet transform, the SIF, LAP and DWT approaches have better fusion results than that not by pre-processing. At the same time, the SIF approach has 36% and 39% lower MSE than that of the LAP and DWT approaches. Therefore, the SIF approach can get better fusion result (Fig. 1(i)) than that in the other two approaches.

5 Conclusions

SGNNs are self-organizing neural network of distinct features which needs no any network structure and network parameters specification by the user and every sample is used only once. They are neural networks of prominent features. The new image fusion SIF approach based on SGNN features highly self-learning property and high computing speed. The SIF approach effectively maintains the characteristics of the



(a) Source image (b) image A (c) image B (d)SIF (e) LAP (f)DWT

(g) image C (h) image D (i) SIF (j) LAP (k) DWT

Fig. 1. Fused images of remote sensing images containing noise in the three approaches

original images from different image fusion and improves the quality of the polluted images by suppressing their noises as seen in the noisy fusion error analyses. The experiment shows that SIF approach is better than the other two approaches and achieves better fusion results by pre-processing to remove noises.

References

1. Hall, D.L., Llinas, J.: An Introduction to Multisensor Data Fusion. Proceedings IEEE, **85** (1997) 6-23
2. Varshney, P.K.: Multisensor Data Fusion. Electronics and Communication Engineering Journal, **9** (1997) 245-253
3. Phol, C.: Multisensor Image Fusion in Remote Sensing: Concepts. Methods and Application. International Journal of Remote sensing, **9** (1998) 823-854
4. Yiyao, L., Venkatesh, Y.V., Ko, C.C.: A Knowledge-based Neural Network for Fusing Edge Maps of Multi-Sensor Images. Information Fusion, **2** (2001) 121-133
5. Zhang, Z.L., Sun, S.H., Zheng, F.C.: Image Fusion Based on Median Filters and SOFM Neural Networks: a Three-step Scheme. Signal Processing, **81** (2001) 1325-1330
6. Wen, W.X., Jennings, A., and Liu, H.: Learning a Neural Tree. In: Proc of International Joint Conference on Neural Networks, Beijing, China (1992)
7. Wen, W.X., Jennings, A., Liu, H., Pang, V.: Some Performance Comparisons for Self-Generation Neural Tree. In: Proc of International Joint Conference on Neural Networks, Beijing China (1992)
8. Wen, W.X., Jennings, A., Pang, V.: A Comparative Study between SGNN and SONN. In: Proc of International Joint Conference on Neural Networks, Beijing China (1992)
9. Inoue, H.: A Study on Ensemble Self-Generating Neural Networks. (Ph.D. dissertation), Okayama University of Science, Japan (2002)

Multifocus Image Fusion Using Spatial Features and Support Vector Machine

Shutao Li[1,2] and Yaonan Wang[1]

[1] College of Electrical and Information Engineering, Hunan University
Changsha, Hunan 410082, China
[2] National Laboratory on Machine Perception, Peking University, Beijing 100871, China
shutao_li@yahoo.com.cn

Abstract. This paper describes an application of support vector machine to pixel-level multifocus image fusion problem based on the use of spatial features of image blocks. The algorithm first decomposes the source images into blocks. Given two of these blocks (one from each source image), a SVM is trained to determine which one is clearer. Fusion then proceeds by selecting the clearer block in constructing the final image. Experimental results show that the proposed method outperforms the discrete wavelet transform based approach, particularly when there is movement in the objects or misegistration of the source images.

1 Introduction

Optical lenses often suffer from the problem of limited depth of field. Consequently, the image obtained will not be in focus everywhere. A possible way to alleviate this problem is by image fusion [1], in which several pictures with different focus parts are combined to form a single image. This fused image will then hopefully contain all relevant objects in focus.

In recent years, various methods based on multiscale transforms have been proposed, including the Laplacian pyramid [2], the gradient pyramid [1], the ratio of Low pass pyramid [3] and the morphological pyramid [4]. More recently, the discrete wavelet transform (DWT) [5], [6] has also been used. In general, DWT is superior to the previous pyramid based methods [6]. While these methods often perform satisfactorily, their multiresolution decompositions and consequently the fusion results are not shift invariant because of an underlying down sampling process. When there is slight camera/object movement or when there is misregistration of the source images, their performance will thus quickly deteriorate.

In this paper, we propose a pixel level multifocus image fusion method based on the use of spatial features of image blocks and support vector machines (SVM). The implementation is computationally simple and is robust to shift problem. Experimental results show that it outperforms the DWT based method. The rest of this paper is organized as follows. The proposed fusion scheme will be described in Section 2. Experiments will be presented in Section 3, and the last section gives some concluding remarks.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 753–758, 2005.

2 SVM Based Multifocus Image Fusion

2.1 Feature Extraction

In this paper, we extract two measures from each image block to represent its clarity. These are described in detail as follows.

2.1.1 Spatial Frequency (SF)

Spatial frequency is used to measure the overall activity level of an image [7]. For an $M \times N$ image *F*, with the gray value at pixel position (*m*,*n*) denoted by *F*(*m*,*n*), its spatial frequency is defined as

$$SF = \sqrt{RF^2 + CF^2}\ . \tag{1}$$

where *RF* and *CF* are the row frequency

$$RF = \sqrt{\frac{1}{MN}\sum_{m=1}^{M}\sum_{n=2}^{N}(F(m,n) - F(m,n-1))^2}\ ,$$

and column frequency

$$CF = \sqrt{\frac{1}{MN}\sum_{n=1}^{N}\sum_{m=2}^{M}(F(m,n) - F(m-1,n))^2}\ ,$$

respectively.

2.1.2 Absolute Central Moment (ACM) [8]

$$ACM = \sum_{i=0}^{I-1}|i-u|p(i)\ . \tag{2}$$

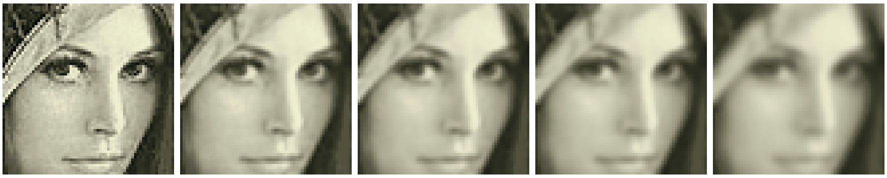
where μ is the mean intensity value of the image, and *i* is the gray level.

2.1.3 Demonstration of the Effectiveness of the Measures

In this section, we experimentally demonstrate the effectiveness of the two focus features. An image block of size 64×64 (Fig. 2(a)) is extracted from the “Lena” image. Fig. 2(b) to Fig. 2(e) show the degraded versions by blurring with a Gaussian filter of radius 0.5, 0.8, 1.0 and 1.5 respectively. As can be seen from Table 1, when the image becomes more blurred, the two features are monotonic accordingly. These results suggest that both two features can be used to reflect image clarity.

2.2 The Fusion Algorithm

Fig.2 shows a schematic diagram of the proposed multifocus image fusion method. Here, we consider the processing of just two source images, though the algorithm can be extended straightforwardly to handle more than two.



(a) original region (b) radius=0.5 (c) radius=0.8 (d) radius=1.0 (e) radius=1.5

Fig. 1. Original and blurred regions of an image block extracted from "Lena"

Table 1. Feature values for the image regions in Fig.1.

	Fig. 1(a)	Fig. 1(b)	Fig. 1(c)	Fig. 1(d)	Fig. 1(e)
SF	40.88	20.61	16.65	14.54	11.70
ACM	51.86	48.17	47.10	46.35	44.96

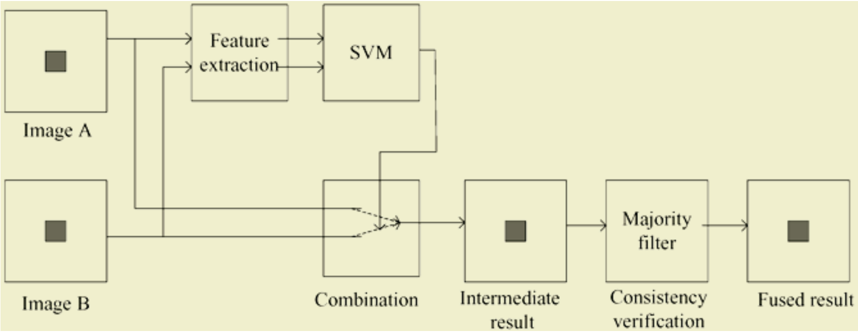


Fig. 2. Schematic diagram of the proposed fusion method.

In detail, the algorithm consists of the following steps:

1. Decompose the two source images A and B into blocks with size of M×N. Denote the ith image block pair by A_i and B_i respectively.

2. From each image block, extract two features above described that reflect its clarity. Denote the feature vectors for A_i and B_i by (SF_{A_i}, ACM_{A_i}) and (SF_{B_i}, ACM_{B_i})respectively.

3. Train a SVM to determine whether A_i or B_i is clearer. The difference vector $(SF_{A_i} - SF_{B_i}, ACM_{A_i} - ACM_{B_i})$ is used as input, and the output is labeled according to

$$\text{target}_i = \begin{cases} 1 & \text{if } A_i \text{ is clearer than } B_i \\ 0 & \text{otherwise} \end{cases} . \tag{3}$$

4. Perform testing of the trained SVM on all image block pairs obtained in Step 1. The ith block, Z_i, of the fused image is then constructed as

$$Z_i = \begin{cases} A_i & \text{if } out_i > 0.5 \\ B_i & \text{otherwise} \end{cases} . \tag{4}$$

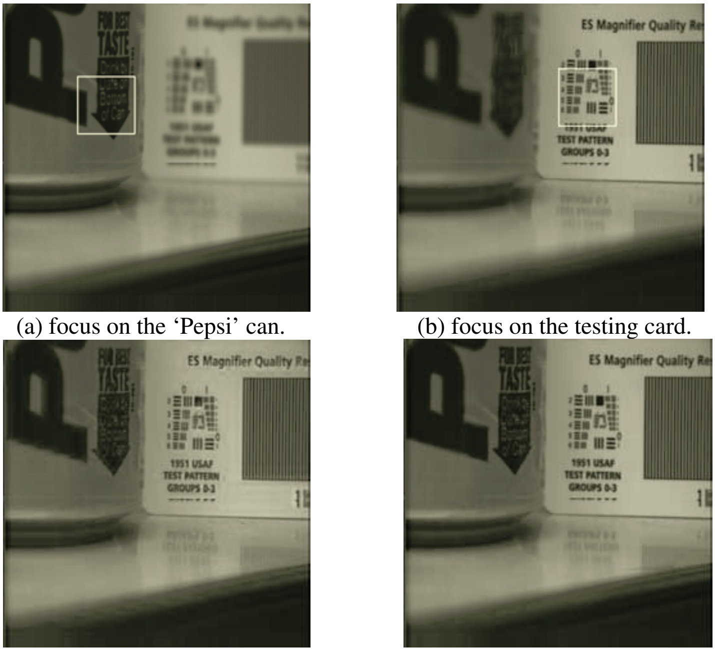
where out_i is the SVM output using the ith image block pair as input.

5. Verify the fusion result obtained in Step 4. Specifically, if the SVM decides that a particular block is to come from A but with the majority of its surrounding blocks from B, this block will be switched to come from B. In the implementation, a majority filter with a 3×3 window is used.

3 Experiments

Experiment is performed on 256 level source images shown in Fig.3(a) and (b). Their sizes are 512×512. The true gray value of the reference image is not available and so only a subjective visual comparison is intended here. Image blocks of size 32×32 are used. Two pairs of regions in Fig.3(a),(b), each containing 18 image block pairs, are selected as training set. In 9 of these block pairs, the first image is clearer than the second image, and the reverse is true for the remaining 9 pairs. The two spatial features are extracted and normalized to the range [0,1] before feeding into SVM. In the experiment, linear kernel is used.

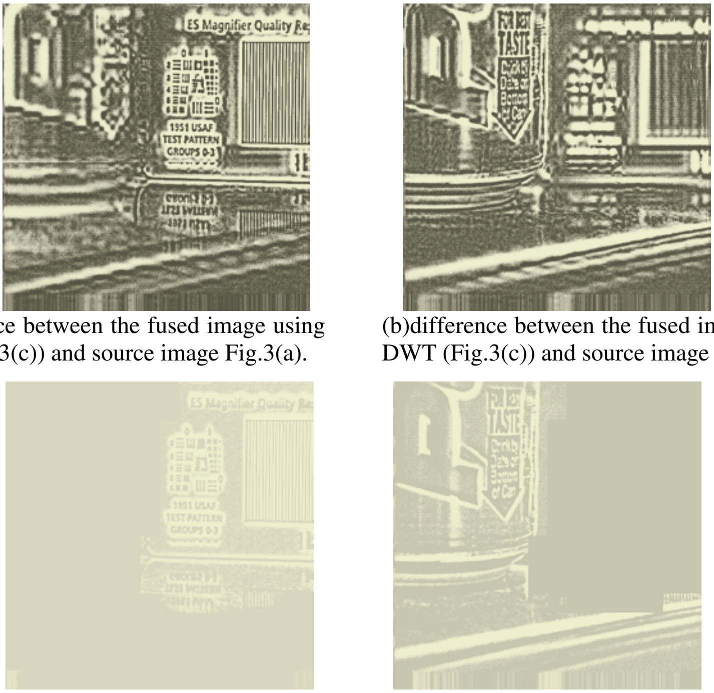
For comparison purposes, we also perform fusion using the DWT. The wavelet basis “db8”, together with a decomposition level of 5, is used. Similar to [6], we employ a region based activity measurement for the active level of the decomposed wavelet coefficients, a maximum selection rule for coefficient combination, together with a window based consistency verification scheme.



(a) focus on the ‘Pepsi’ can. (b) focus on the testing card.

(c) fused image using DWT (db8, level=5). (d) fused image using SVM.

Fig. 3. The “Pepsi” source images and fusion results. The training set is selected from regions marked by the rectangles in Fig. 3(a) and Fig. 3(b).



(a)difference between the fused image using DWT (Fig.3(c)) and source image Fig.3(a).

(b)difference between the fused image using DWT (Fig.3(c)) and source image Fig.3(b).

(c) difference between the fused image using SVM (Fig. 3(d)) and source image Fig.3(a).

(d)difference between the fused image using SVM (Fig.3(d)) and source image Fig. 3(b).

Fig. 4. Differences between the fused images in Fig.3(c),(d) and source images in Fig.3(a),(b).

Fusion results on using DWT and SVM are shown in Fig. 3(c),(d). Take the "Pepsi" images as an example. Recall that the focus in Fig. 3(a) is on the Pepsi can while that in Fig. 3(b) is on the testing card. It can be seen from Fig.3(d) that the fused image produced by SVM is basically a combination of the good focus can and the good focus board. In comparison, the result by DWT shown in Fig.3(c) is much inferior. Clearer comparisons of their performance can be made by examining the differences between the fused images and each source image (Fig.4).

4 Conclusions

In this paper, we proposed a method for pixel level multifocus image fusion by using the spatial features of image blocks and SVM. Features indicating the clarity of an image block, namely, spatial frequency and absolute central moment, are extracted and fed into the support vector machine, which then learns to determine which source image is clearer at that particular physical location. Experimental results show that this method outperforms the DWT based approach, particularly when there is object movement or registration problems in the source images.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No.60402024).

References

1. Burt, P.J., Kolczynski, R.J.: Enhanced Image Capture through Fusion. In Proc. of the 4th Inter. Conf. on Computer Vision, Berlin (1993) 173-182
2. Burt, P.J., Andelson, E.H.: The Laplacian Pyramid as a Compact Image Code. IEEE Trans. Comm., **31** (1983) 532-540
3. Toet, A., Ruyven, L.J., Valeton, J.M.: Merging Thermal and Visual Images by a Contrast Pyramid. Optic. Eng., **28** (1989) 789-792
4. Matsopoulos, G.K., Marshall, S., Brunt, J.N.H.: Multiresolution Morphological Fusion of MR and CT Images of the Human Brain. Proc. of IEE: Vision, Image and Signal, **141** (1994) 137-142
5. Li, H., Manjunath, B.S., Mitra, S.K.: Multisensor Image Fusion using the Wavelet Transform. Graph. Models Image Proc., **57** (1995) 235-245
6. Zhang, Z., Blum, R.S.: A Categorization of Multiscale-Decomposition-Based Image Fusion Schemes with a Performance Study for a Digital Camera Application. Proc. of the IEEE, **87** (1999) 1315-1325
7. Eskicioglu, A.M., Fisher, P.S.: Image Quality Measures and Their Performance. IEEE Trans. Comm., **43** (1995) 2959-2965
8. Shirvaikar, M.V.: An Optimal Measure for Camera Focus and Exposure. 36th IEEE Southeastern Symp. on Sys. Theory, Atlanta (2004) 472-475

A New Scheme for Fusion of Multispectral and Panchromatic Images Based on Residual Error

Zhirong Ge[1], Bin Wang[1,2], and Liming Zhang[1]

[1] Department of Electronic Engineering, Fudan University, Shanghai 200433, China
{032021034,wangbin,lmzhang}@fudan.edu.cn
[2] The Key Laboratory of Wave Scattering and Remote Sensing Information
(Ministry of Education), Fudan University, Shanghai 200433, China

Abstract. The standard data fusion methods may not be satisfactory to merge a high-resolution panchromatic image and low-resolution multispectral images because they can distort the spectral characteristics of the multispectral data**.** This paper proposes a new scheme based on residual error for fusion of such images. By merging high-resolution residual error extracted from panchromatic image and low-resolution residual errors from multispectral images through principal component analysis (PCA), the high-resolution residual errors of multispectral images can be restored. We point out that our scheme successfully solves the problem of spectral distortion. Finally, the performances of the proposed scheme are demonstrated experimentally, and the comparisons of the performances with standard IHS (intensity-hue-saturation), PCA, and wavelet transform-based fusion methods are made.

1 Introduction

There are several remote sensing applications that simultaneously require high spatial and high spectral resolution in a single image. Due to the sensor limitations, observations with high spectral resolution may exhibit low spatial resolution, and *vice versa*. This contradiction can be overcome by merging panchromatic and multispectral data to provide complementary information.

A kind of widely used methods for image fusion are component substitution methods such as IHS (intensity-hue-saturation) [1] and PCA (principal component analysis) [2], [3]. However, the IHS and PCA methods produce spectral degradation. With wavelet-based method [4], [5], the spectral degradation may be reduced to some extent, but the integration of the spectral and spatial feature usually does not appear smooth.

In this paper, we propose a scheme based on residual error which can produce more reliable results than the standard IHS, PCA, and wavelet-based methods by restoring the high-resolution residual errors of multispectral data. We point out that our scheme should successfully solve the problem of spectral degradation common in the fusion techniques mentioned above. The introduction of residual error into the problem of image fusion constitutes the first originality of our work. In addition, the PCA is used to fuse the residual error images which constitutes the second originality of our research. Finally, some experimental results are shown and some statistical parameters are adopted to evaluate the quality of the fusion results.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 759–764, 2005.

The remainder of this paper is organized as follows. Section 2 is used to describe the proposed scheme. Some experimental results are shown in section 3. Conclusion is given in section 4.

2 The Proposed Scheme

2.1 Standard PCA Method

Usually, in standard PCA method, the first principal component (PC^1) of multispectral data is substituted by the panchromatic image, whose histogram has previously been matched with that of PC^1. When the panchromatic wavelength does not exactly cover the spectral range of multispectral data, the intensity of the panchromatic image will be significantly different from PC^1. Such difference will inevitably results in spectral degradation presented in the merged image.

2.2 Residual Error

If a set of high-resolution multispectral data were available, in most case, their spatial details would be similar to those from the corresponding panchromatic image. Based on this similarity, the panchromatic spatial details can be injected into the low-resolution multispectral spatial details without changing the original information. This injection can be performed by the PCA transformation in a simple and efficient way. Our method is developed based on such principle, and its flow chart is shown in Fig.1.

In this part, we define residual error as the spatial detail obtained by the given process (***E***) in Fig. 2. Here, we use Bicubic interpolation (***Z***) to construct the expanded image from the original image, and then apply averaging filtering and down-sampling (***L***) on the expanded image to produce the degraded image. The residual error is defined as the difference between the original image and the degraded image. In this way, the panchromatic residual error $\boldsymbol{e_p}$ and multispectral residual error $\boldsymbol{e_g}$ are calculated respectively. The introduction of the residual errors into the image fusion constitutes the first originality of our work.

2.3 The Proposed Scheme Based on Residual Error

Before the fusion of the panchromatic and the multispectral residual errors, we re-sample the multispectral residual error $\boldsymbol{e_g}$ to the same size as the panchromatic residual error $\boldsymbol{e_p}$. This operation is also done by the Bicubic interpolation (***Z***) as follow

$$e_f = Z(e_g) \cdot \tag{1}$$

The expanded residual error $\boldsymbol{e_f}$ has low spatial resolution. Here, we use PCA transformation to inject the detail information of $\boldsymbol{e_p}$ into the residual error $\boldsymbol{e_f}$ as below which constitutes the second originality of our research. First, we apply all the bands of $\boldsymbol{e_f}$ as the input data to the PCA transformation, and non-correlated principal components PC^n, $n=1, 2, 3$ can be obtained. In general, the first principal component (PC^1) has the maximal variance and collects primary information. The other components PC^2 and PC^3 also contain a fraction of spatial information.

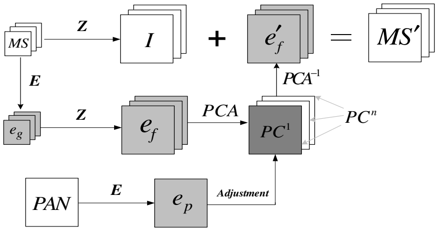


Fig. 1. Fusion scheme based on residual error

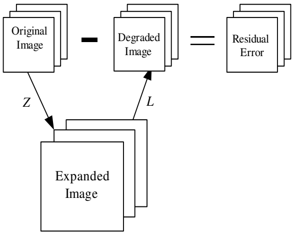


Fig. 2. Extraction of residual error

The PC^1 is substituted by $\boldsymbol{e_p}$, whose histogram has been previously matched with that of PC^1. Here, we expect that the mean and variance of $\boldsymbol{e_p}$ is the same as that of PC^1. Because the mean of any residual error is equally zero, we only need to adjust the variance of $\boldsymbol{e_p}$ to that of PC^1 as follow

$$e_p' = e_p \times \sigma_{PC^1} \big/ \sigma_{e_p} \ , \tag{2}$$

where $\sigma_{e_p}^2$ and $\sigma_{PC^1}^2$ represent the variances of $\boldsymbol{e_p}$ and PC^1 respectively.

Then, the inverse transformation is applied to the whole dataset formed by the modified panchromatic residual error $\boldsymbol{e_p}$*'* and the PC^2 and PC^3, obtaining the merged residual error $\boldsymbol{e_f}$*'* with high spatial resolution. It can be easily inferred that the mean value of $\boldsymbol{e_f}$*'* is also zero, which would help to avoid the spectral degradation.

Finally, we use Bicubic interpolation to construct the expanded image ***I*** from the original multispectral image. The high-resolution residual error $\boldsymbol{e_f}$*'* and the expanded image ***I*** can be added pixel by pixel as below.

$$MS' = I + e_f' \ . \tag{3}$$

The equation (3) produces a merged image MS' with high spatial and high spectral resolution simultaneously. By the high-resolution residual error $\boldsymbol{e_f}$*'*, the spatial detail of the panchromatic image missed in the multispectral data is incorporated into the latter. By the expanded multispectral image ***I***, our method preserves the primary spectral information of the original multispectral data. So our method not only increases the spatial details but also preserves the spectral information as much as possible.

3 Experiment Results

In this section, we use LANDSAT-7 multispectral and panchromatic images covering an area of Shanghai as experimental data. The 1st–3rd bands of multispectral data have a spatial resolution of 30 m respectively. Their false color composition (RGB = Bands 3, 2, & 1) is shown in Fig. 3(a), while the panchromatic image with 15 m pixels is shown in Fig. 3(b). Here, we merge the panchromatic and the multispectral images with four algorithms, the proposed scheme and the standard IHS, PCA, and wavelet-based methods. For the wavelet-based method, the Daubechies four coefficient wavelet basis and 3 levels of wavelet decomposition are adopted as described in [5]. Their results will be compared both visually and quantitatively as below.

When these standard methods are used, there exists spectral degradation. For example, in Fig. 3(e), the color of the river (at the top right corner) is close to black while the original white color in Fig. 3(a) is required to be preserved.

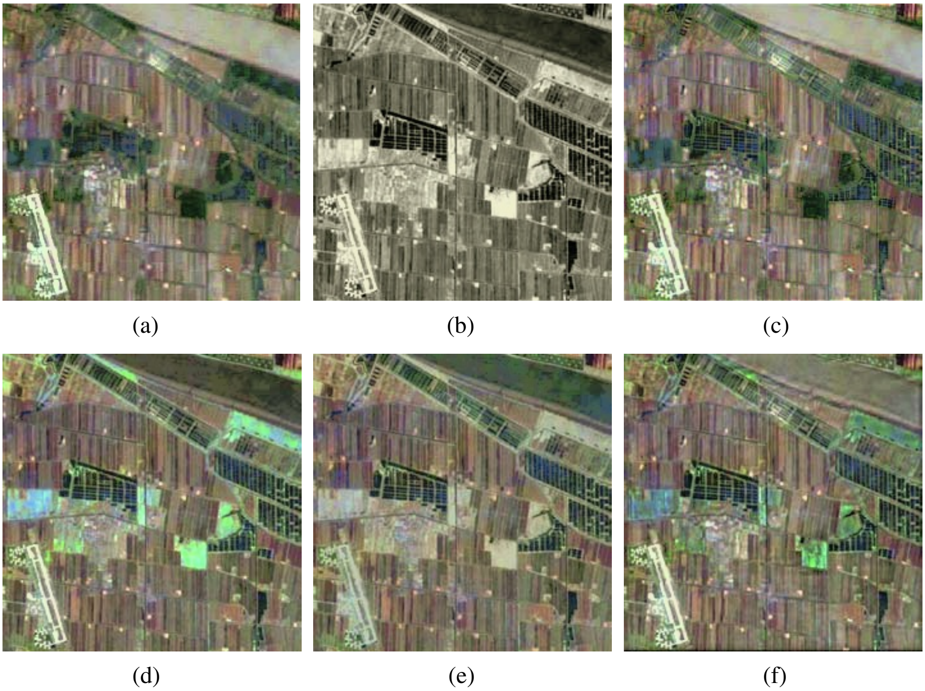


Fig. 3. Comparison of the fusion results by the proposed scheme and the standard methods. (a) Original multispectral false color image; (b) Panchromatic image; Fusion results by (c) the proposed scheme, (d) the standard IHS, (e) the standard PCA, and (f) the wavelet-based method

When the proposed scheme is used, we can see that the color of the fused image in Fig.3 (c) keeps almost the same as the original one for all objects. This shows that the behaviors of our method are better with respect to the spectral quality. In addition, our method also increases the spatial quality. For example, the interlaced agricultural fields (at the center region) are more sharply than the original image.

We also use two types of statistical parameters [6] to evaluate the performances of fusion methods.

The first type of statistical parameters reflects the spatial detail information, e.g.

Standard deviation $$\hat{\sigma}^2 = \frac{1}{M * N}\sum_{i=1}^{M}\sum_{j=1}^{N}(f(i,j)-\overline{f}) \cdot \tag{4}$$

Entropy $$H(P) = -\sum_{i=1}^{L} P(i) \times \log_2 P(i) \cdot \tag{5}$$

Definition $$\overline{g} = \frac{1}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N}\sqrt{\left(\frac{\partial f(i,j)}{\partial x}\right)^2 + \left(\frac{\partial f(i,j)}{\partial y}\right)^2} \cdot \tag{6}$$

The second type of statistical parameters indicates the capability of preserving the spectral characteristics. They are computed by comparing the merged image to the original image as below:

Distortion degree $$D_{degree} = \frac{1}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N}\left|f(i,j) - g(i,j)\right| \cdot \tag{7}$$

Distortion index $$D_{index} = \frac{1}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{\left|f(i,j) - g(i,j)\right|}{f(i,j)} . \tag{8}$$

Correlation $$Corr(f,g) = \frac{\sum(f(i,j)-\overline{f})\times(g(i,j)-\overline{g})}{\sqrt{\sum(f(i,j)-\overline{f})^2\sum(g(i,j)-\overline{g})^2}} \cdot \tag{9}$$

Table 1. Comparison of fusion performances based on statistical parameters

Images	Standard deviation	Entropy	Definition	Distortion degree	Distortion index	Correlation
MS(2nd band)	35.461	4.833	4.535			
PAN	49.008	5.207	12.669			
Our scheme	39.313	5.019	10.876	10.428	0.106	0.931
IHS	40.628	5.011	10.199	35.903	0.376	0.094
PCA	35.585	4.929	9.105	33.374	0.342	0.145
Wavelet	36.399	4.857	10.424	24.376	0.255	0.541

Table 1 shows the statistical parameters calculated on the 2nd band of different methods. Here, we have to point out that other two bands also show the similar results. From the Table 1, we can see that our method effectively improves the spatial detail because the increase of deviation, entropy, and definition is evident compared with those of original image.

As to spectral quality, our scheme produces the minimal distortion degree and distortion index in comparison with the standard solutions. In addition, the correlation of our scheme is higher than those of the standard methods. Such parameters indicate that the spectral information of our method is more similar to the original one than those of the standard methods.

From the Fig. 3 and the Table 1, we find out that the proposed scheme not only increases the spatial details, but also preserves the spectral information to a great extent. So our method should successfully solve the problem of the spectral degradation during the image fusion.

4 Conclusions

In this paper, we presented a fusion scheme based on the residual error which can be used to merge the panchromatic and the multispectral images. By introducing the residual error of the panchromatic and the multispectral images, the proposed method can perform the PCA transformation on the level of residual error and then restore the high-resolution multispectral residual error. From the experiment results, we have seen that the performances of our method are better than the standard IHS, PCA, and wavelet-based methods and we should conclude that our method successfully solve the problem of spectral degradation.

References

1. Carper, J., Lillesand, T.M., Kiefer, R.W.: The Use of Intensity-Hue-Saturation Transformations for Merging SPOT Panchromatic and Multispectral Image Data. Photogrammetric Engineering and Remote Sensing, **56** (1990) 459-467
2. Shettigara, V.K.: A Generalized Component Substitution Technique for Spatial Enhancement of Multispectral Images Using a Higher Resolution Data Set. Photogrammetric Engineering and Remote Sensing, **58** (1992) 561-567
3. Chavez, P.S., Stuart, J., Slides, C., Anderson, J.A.: Comparison of Three Different Methods to Merge Multiresolution and Multispectral Data: Landsat TM and SPOT Panchromatic. Photogrammetric Engineering and Remote Sensing, **57** (1991) 295-303
4. Nunez, J., Otazu, X., Fors, O., Prades, A., Pala, V., Arbiol, R.: Multiresolution-Based Image Fusion with Additive Wavelet Decomposition. IEEE Transactions on Geoscience and Remote Sensing, **37** (1999) 1204-1211
5. Gonzale, A.M.Z, Saleta, J.L., Catalan, R.G., Garcia, R.: Fusion of Multispectral and Panchromatic Images Using Improved IHS and PCA Mergers Based on Wavelet Decomposition. IEEE Transactions on Geoscience and Remote Sensing, **42** (2004) 1291-1299
6. Li, B., Wei, J., Peng, T.: Objective Analysis and Evaluation of Remote Sensing Image Fusion Effect. Computer Engineering & Science, **26** (2004) 42-46

Binocular 3D Reconstruction Based on Neural Network

Mingxing Lin[1], Yongrui Zhao[2], Zhiguang Guan[2],
Fenghua Ding[2], Qingxin Xu[2], and Xiaohua Wang[2]

[1] School of Mechanical Engineering, Shandong University, Jinan, Shandong 250061, China
mxlin2000@163.com
[2] Shandong University of Science and Technology, Qingdao, Shandong 266510, China

Abstract. This paper introduces the computer vision to 3D reconstruction of the object. A novel method is developed, which adopts the digital camera as the sensor and uses the techniques of image processing and vision calculation synthetically to realize the untouched 3D measurement. The binocular measurement theory and the camera calibration based on the neural network are also described. Finally, we give the procedures of the binocular 3D reconstruction and simulate a case with Matlab program.

1 Introduction

The 3D reconstruction technique of computer vision is a significant field in computer vision, and what's more, it can help realize 3D untouched measurement and fast reconstruction. The neural network has wide applications in computer vision detection and 3D reconstruction because of its strong non-linear mapping capacity. The binocular vision is used to obtain the different information of the object from two directions, which is the main way for the research of 3D reconstruction. In this paper, we apply the neural network to the binocular 3D reconstruction, establish the binocular imaging model, and complete two cameras calibration once. We first introduce the binocular measurement theory and the structure of the neural network, and then list the procedures of the 3D reconstruction. At last, we give all requisite experiments and results.

2 Binocular Measurement

The geometric relationship of the CCD1 and CCD2 cameras is shown in Fig.1, the points O_1 and O_2 are the centers of the CCD cameras, the level distance of the points O_1 and O_2 is D, and the points C_1 and C_2 are the centers of the projection plane. Z_{c1} and Z_{c2} is the respective optical axis of the CCD1 and CCD2, α_1 is the angle of Z_{c1} and Z_w axis, and α_2 is the angle Z_{c2} and Z_w, the camera focus is $f=O_1C_1=O_2C_2$, the angle between the projection of the spatial point $P(x_w\ y_w\ z_w)$ on the X_wZ_w plane and

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 765–771, 2005.

the axis Z_{c1} is β_1, and the axis Z_{c2} is β_2 the coordinate of the point P is (u_1 v_1) on the projection plane. According to Fig.1, the following equations can be obtained [1]:

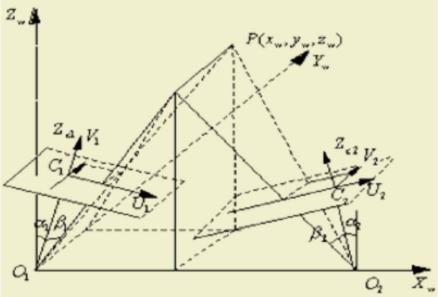


Fig. 1. Binocular measurement theory

$$z_w = \frac{D}{tg(\alpha_1 + \beta_1) + tg(\alpha_2 + \beta_2)} , \tag{1}$$

$$x_w = \frac{Dtg(\alpha_1 + \beta_1)}{tg(\alpha_1 + \beta_1) + tg(\alpha_2 + \beta_2)} , \tag{2}$$

$$y_w = v_1 \frac{\sqrt{x_w^2 + z_w^2}}{\sqrt{f^2 + u_1^2}} = \frac{v_1 D \cos \beta_1}{f\left[\sin(\alpha_1 + \beta_1) + tg(\alpha_2 + \beta_2)\cos(\alpha_1 + \beta_1)\right]} . \tag{3}$$

According to (3), we know that if D, f, α_1 and α_2 is invariant, the coordinate of the point P(xw yw zw) is the non-linear function of the corresponding image point (u_1 v_1)and (u_2 v_2). The neural network has a strong non-linear mapping function. It can be used for establishing the complicated math model. According to the neural network model, we can get the relationship between the image information and the 3D information.

3 3D Reconstruction

The spatial point is the basic cell of the 3D space. If all the spatial coordinates of the points on the object surface can be obtained, the shape and the position of the object can be confirmed. For the point P, the image point is P_1 observed with C_1 camera, in Fig. 2. The position P can not be determined by P_1, because the image point of any point P' on the line O_1P (O_1 is the center of C_1 camera) is P_1. While, if we use two cameras $C_1$and C_2 to observe the point P, the points P_1 and P_2 can be obtained, and the point P is the intersection point of the line O_1P_1 and O_2P_2, therefore the position is determined. If we use the points P_1 and P_2 as the neural network input, the space coordinates of the point P are the output.

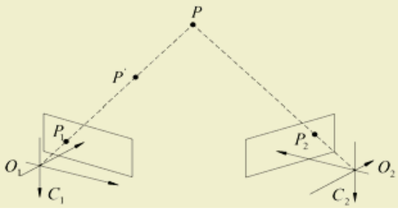


Fig. 2. Binocular reconstruction space point

The neural network of binocular calibration is a feed-forward BP neural network, which includes the input layer, hidden layer and output layer. Illustrated in Fig.3, the actual image points $U_1\alpha$, $V_1\alpha$, $U_2\alpha$ and $V_2\alpha$ are the input signal, the actual coordinates of the space measure points are the expired output signal.

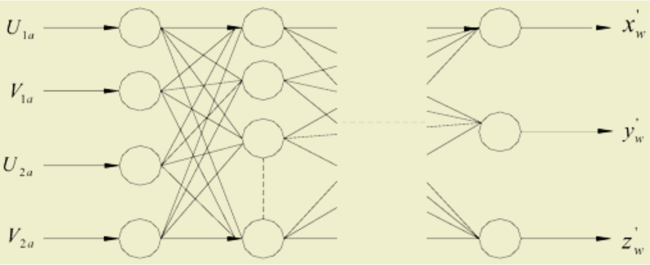


Fig. 3. Feed-forward BP neural network

During the training process, the expired output compares with the actual output x'_w , y'_w , z'_w , and the weights are adjusted all the time according to the error between x'_w , y'_w , z'_w and x_w , y_w , z_w until meet the following equation.

$$\frac{1}{m}\sum_{i=1}^{m}\left\{\left[x_w(i)-x'_w(i)\right]^2+\left[y_w(i)-y'_w(i)\right]^2+\left[z_w(i)-z'_w(i)\right]^2\right\}\le MSE \ . \qquad (4)$$

Where, m is the sum of the input/output stylebook data, MSE is the given average square error. At the end of the study, the weights of the neural network have saved the math model information of the binocular camera, and we think the output signal is equal to the actual coordinates, $x_w=x'_w$, $y_w=y'_w$ and $z_w=z'_w$. Therefore, after the calibration [2], the neural network can learn the non-linear relationship between the stereo vision system and the interior and exterior orientation elements, and save them. If we take $U_1\alpha$, $V_1\alpha$, $U_2\alpha$ and $V_2\alpha$ as the neural network input signal, the actual point coordinates x_w , y_w , z_w can be obtained in the output port.

4 Experiment

The procedures of the reconstruction are: binocular cameras calibration→get the binocular image→image preprocessing→image feature extract and match→ calculate the space coordinate→3D reconstruction. Table 1 is the result of the binocular calibration by the neural network. Fig.4 is the original image.

Table 1. BP neural network calibration results The binocular cameras are calibrated by BP neural network

u_1	v_1	u_2	v_2	x	y	z	x'	y'	z'
138.9003	311.1941	366.0121	299.2581	66	110	0	65.4086	111.5477	-0.0023
159.1199	322.8679	384.5780	311.5466	77	110	0	76.1340	112.5531	0.0095
178.9000	334.2880	402.8552	323.6440	88	110	0	86.6066	112.2446	0.0308
199.1197	345.9618	422.1241	336.3979	99	110	0	98.1526	110.7837	0.0478
219.3394	357.6356	441.8412	349.4483	110	110	0	109.1417	110.6544	0.0522

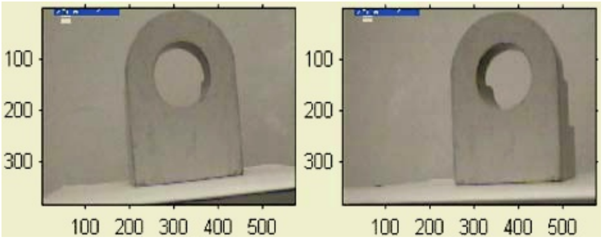


Fig. 4. Original image

4.1 Canny Edge Detection

The Canny operator judges the edge points by the first order derivative, and then determine the maximum by the first order differential operator. Fig.5 shows the edge extraction using the Canny operator.

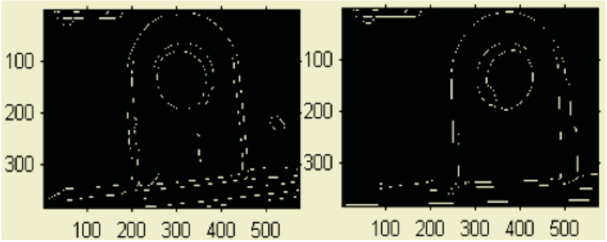


Fig. 5. Canny edge detection

4.2 Image Features Extraction and Matching

SUSAN corner extraction is a detection algorithm based on the gray image. The feature of any pixel is related to its local feature, and the continuity region has the consistent gray feature, also the local area concludes much image structure information. Therefore, the corner and edge can be inspected by the size, center and second order moments of the SUSAN [3]. The SUSAN method has strong anti-noise capacity, simple control parameters and can realize automated adjustment. Illustrated in Fig.6.

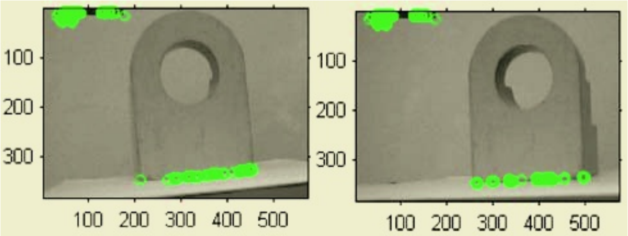


Fig. 6. SUSAN corner extraction

Freeman chain code is an expression method for edge points. It expresses the object by a series of jointed line segments with special length and direction. This method can greatly decrease the data, because a vector can replace two coordinates. The chain code may change when the object has a rotation, while the difference code can give a good solution to this problem. A method of extracting image features based on the difference code is developed by us [4], which can extract the corners, lines and curves. Fig.7 shows the results.

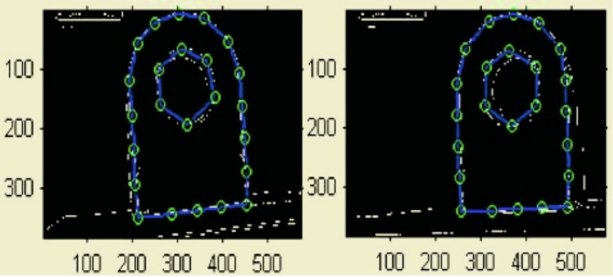


Fig. 7. Extract line and corner using Freeman

The feature points and line segments are matched by the relaxation iteration algorithm and the Freeman. The table 2, 3 shows the matching results.

4.3 3D Coordinates Reconstruction

If we apply the matched right and left image coordinate to the calibrated neural network, the 3D coordinates of the feature space can be obtained in the output port. The table 4 shows the results. Thus, part of the 3D coordinates has been obtained, and the full surface can be recovered by interpolation reconstruction.

Table 2. Matching of point by the relaxation iteration algorithm (U_1 V_1) and (U_2 V_2) are the left points and its right matching points respectively

U_1	V_1	U_2	V_2	U_1	V_1	U_2	V_2
303	6	372	7	204	294	251	287
358	14	426	22	201	236	249	233
412	53	474	66	200	178	247	179
437	108	485	120	192	120	246	125
443	163	488	174	210	56	266	66
448	218	491	228	250	22	317	20
453	273	493	282	309	66	360	69
452	328	489	335	364	87	422	98
397	334	434	337	382	149	420	165
342	339	379	339	322	194	368	198
287	345	324	341	261	161	309	165
213	351	256	343	257	100	313	98

Table 3. Matching of line by the freeman The slop and intercept of left line and its right matching line are p and q respectively

Left line		Right matching line		Left line		Right matching line	
Slop p	Intercept q	Slop p	Intercept q	Slop p	Intercept q	Slop p	Intercept q
0.1455	-38	0.2778	-96	19.3333	-3650	27.0000	-6490
0.7222	-245	0.9167	-369	58.0000	-11422	27.0000	-6490
2.2000	-853	4.9091	-2261	7.2500	-1272	54.0000	-13159
9.1667	-3898	18.0000	-8610	-3.5556	803	-2.9500	851
11.0000	-4710	18.0000	-8610	-0.8500	234	-0.9020	306
11.0000	-4710	27.0000	-13029	-0.3019	97	-0.2364	95
-55.0000	25188	-13.2500	6814	0.3818	-52	0.4677	-99
-0.1091	377	-0.0364	353	3.4444	-1166.8	-33.5000	14235
-0.0909	370	-0.0364	353	-0.7500	435.5	-0.6346	432
-0.1091	376	-0.0364	353	0.5410	19.8	0.5593	-8
-0.0811	368	-0.0294	351	15.2500	-3819.2	-16.7500	5341
6.3333	-998	11.2000	-2524	-0.6538	268	-0.6170	291

5 Conclusions

The main steps for recover the model surface conclude the establishment of the binocular model, calibration of the camera, calculation of the 3D coordinates. The introduced method can be used for the reconstruction of traffic accident, converse project, 3D medicine reconstruction, film stunt, face recognition etc. The technology of 3D reconstruction will be used widely in many areas in the future, though it has many shortages at present.

Table 4. 3D coordinates reconstruction 3D coordinates of the feature space are obtained in the neural network's output

x_w	y_w	z_w	x_w	y_w	z_w
7.6006	26.9391	-11.8589	-13.4709	8.5199	3.5855
20.8111	30.7001	-10.8255	-22.9461	0.2461	-0.4312
24.9344	31.0632	-10.6328	2.0590	27.2364	-8.9574
25.7266	31.6661	-10.3756	13.0352	38.5042	-9.3860
23.3768	36.5065	-8.9770	-0.0989	32.8277	-10.8649
32.1193	36.8759	-7.5266	-1.5410	34.4281	-11.7739
27.8864	29.3907	-5.2512	8.6061	37.9493	-12.2527
36.1831	30.1131	-5.1115	22.6846	37.1196	-10.6484
-5.1925	0.6345	2.5070	20.8756	44.2458	-10.7123
-15.5210	-0.2660	0.3834	17.3128	36.6793	-11.1437
-17.6795	5.2140	-1.2360	15.9170	37.5574	-11.1011
-14.4841	8.8593	3.5728	7.2227	39.4171	-12.1483

Acknowledgements

The work presented in this paper is supported by a grant from the Science and Technology Developing Project of Shandong Province (research on rapid integrated manufacture system for mass customization).

References

1. Wang, X., Ye, M., Wang, X., Wang, H.: Binocular Vision Sensor Modeling Based on Neural Network, Optical Instruments, **24** (2002) 42-46
2. Zhao, Q., Sun, Z., Lan, L.: Neural Network Technique in Camera Calibration, Control and Decision, **17** (2002) 336-338
3. Smith, S.M., Brady, J.M.: SUSAN-A New Approach to Low Level Image Processing, Internal Technical Report TR95SMS1c, (1995)
4. Lin, M., Wang, X., Guan, X.: Research on the Feature Extraction for Image Based on Differential Code, Yiqi Yibiao Xuebao, **25** (2004) 465-467

A Neural Network Based Lossless Digital Image Watermarking in the Spatial Domain

Jun Sang[1,2] and Mohammad S. Alam[2]

[1] School of Software Engineering, Chongqing University
Chongqing 400044, China
sangjun@tom.com
[2] Department of Electrical and Computer Engineering, University of South Alabama
Mobile, Alabama 36688, USA
malam@usouthal.edu

Abstract. A lossless digital image watermarking algorithm based on neural network is proposed. The algorithm uses the neural network technique and the exclusive-or (XOR) operation to model the relationships among some randomly selected pixels with their neighborhoods, and to extract the features of the cover image instead of embedding watermark into it. It does not degrade the visual quality of the cover image, i.e., lossless watermarking. The algorithm may be used for automatic piracy detection as well as copyright demonstration by co-operating with a copyright authentication center.

1 Introduction

Digital watermarking is one of the main research areas of information hiding, aiming at copyright protection, image authentication, piracy tracing, etc. So far, many various watermarking techniques have been proposed for different applications [1-5].

In most of the digital image watermarking techniques, a watermark is embedded into the original cover image. The embedded watermark should not degrade the visual quality of the cover image seriously, while can be recovered or detected from the watermarked image even after some non-malicious or malicious attacks.

To keep the visual quality of the watermarked image, Chin-Chen Chang *et al.* proposed a lossless watermarking scheme [2]. They analyzed and extracted a binary pattern from the original cover image, and applied the XOR operation to the obtained binary pattern and the binary watermark to generate the secret key, which will be used for watermark recovery. This scheme does not actually embed the watermark into the cover image. Thus, the 'watermarked image' will be no difference with the original one; namely, the watermark 'embedding' is lossless.

In [3], Gwoboa Horng *et al.* presented a neural network based lossless watermarking for copyright protection. They trained a neural network to establish the relationship model among some Discrete Wavelet Transform (DWT) coefficients of the original/attacked images and the watermark. Due to the learning and adaptive capabilities, the trained neural network will be able to recover the watermark.

Zhang *et al.* developed a neural network based fragile watermarking technique [4]. A neural network system is utilized to model the cover image in the spatial domain.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 772–776, 2005.

They embedded the watermark by adjusting the relationship among the desired output and the corresponding output of the neural network model. Based on the extracted watermark, the illegal modification can be detected and located.

Based on the research works mentioned above, we propose a neural network based lossless digital image watermarking algorithm in the spatial domain. The algorithm models the cover image with a neural network as [4] did. However, unlike [4], we do not embed any information into the original cover image. Instead, we use the neural network model and the XOR operation to extract features from the cover image to be the secret key for watermark recovery.

2 The Proposed Watermarking Algorithm

The proposed watermarking algorithm includes two main procedures.

2.1 Secret Key Generation

The main steps involved in the secret key generation procedure include the following:

Step 1: Subset selection

Consider a grayscale cover image *I* with the size of *M* x *N* and an M_w x N_w binary watermark image *W*. We randomly select M_w x N_w pixels from *I*. The only requirement is that none of the pixels is at the border of the image to guarantee all the pixels have valid 3 x 3 neighborhoods. Some methods can be used for random pixels selection [4], [5]. The set of the parameter(s) used in this step may be represented as *K*.

Step 2: Neural network model establishment

An 8 x 10 x 1 back-propagation (BP) neural network *BPNN* (8 nodes for the input layer, 10 nodes for the hidden layer, and 1 node for the output layer) is established.

For the training pattern,

The input is:

$(p_{x-1,y-1}, p_{x-1,y}, p_{x-1,y+1}, p_{x,y-1}, p_{x,y+1}, p_{x+1,y-1}, p_{x+1,y}, p_{x+1,y+1})$, for all selected pixels (x, y).

The desired output is: $(p_{x,y})$, for all selected pixels (x, y).

Where $p_{i,j}$ is the intensity value of the pixel (i, j) divided by 255 to constrain the input and output values of the neural network to be in the interval [0, 1], since we want to use sigmoid function as the transfer functions for the hidden and output layers.

The training goal is set as *mean absolute error (MAE) < T/255*

Step 3: Binary sequence obtaining

Suppose $p_{i,j}$ is a desired output value of the neural network, while $p'_{i,j}$ is the corresponding output value. A binary sequence *b* can be obtained by applying the following constraint:

$$b_{i,j} = \begin{cases} 1 & if\ p_{i,j} > p'_{i,j} \\ 0 & otherwise \end{cases} \qquad i = 1,2,...,M_W, \quad j = 1,2,...,N_W \ . \tag{1}$$

Step 4: Secret key computation

Apply the XOR operation to *b* and *W* to compute the secret key *KEY* as:

$$KEY_{i,j} = b_{i,j} \oplus W_{i,j} \quad i = 1,2,...,M_W, \quad j = 1,2,...,N_W \ . \tag{2}$$

where $\oplus$ represents the XOR operation.

The set of the parameter(s) *K*, the trained neural network *BPNN*, and the secret key *KEY* will be registered at a copyright authentication center to prevent the malicious attackers from forging illegal neural network with their own watermarks [3].

2.2 Watermark Recovery

Watermark recovery procedure includes three steps: subset selection, binary sequence obtaining, and watermark recovery.

Step 1: Subset selection

A subset of pixels is selected from the test image using *K*.

Step 2: Binary sequence obtaining

With the selected pixels, the input and the desired output patterns for the neural network *BPNN* are obtained. By comparing the desired output and the corresponding output of the neural network, a binary sequence b' may be obtained using the similar method shown in Eq. (1).

Step 3: Watermark recovery

Applying the XOR operation to b' and the secret key *KEY*, which is actually the inverse operation of Eq. (2), the candidate watermark W' is recovered.

With the recovered watermark, one may verify whether the test image is a copy of the cover image or not, and the potential piracy may be detected.

3 Experimental Results

We carried out a series of simulations to test the algorithm. The 512 x 512 grayscale image ‘Peppers’ shown in Fig. 1 is used as the cover image and a 256 x 256 binary watermark image is shown in Fig. 2. The training goal *T* for the neural network in step 2 of subsection 2.1 was set as 16.

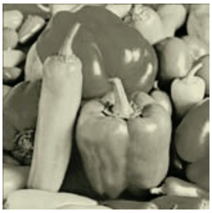


Fig. 1. The original ‘Peppers’ image

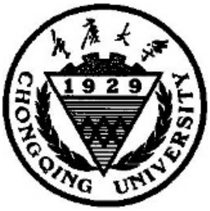


Fig. 2. The watermark image

Since we do not embed any data into the cover image, there will be no visual quality degradation to the 'watermarked image'. Actually, the 'watermarked image' is identical to the original image. We will only discuss the robustness of the proposed algorithm but the imperceptibility.

Obviously, from the original cover image, the watermark can be recovered completely.

Due to the space limitation, we just show representative experimental results in Fig. 3.

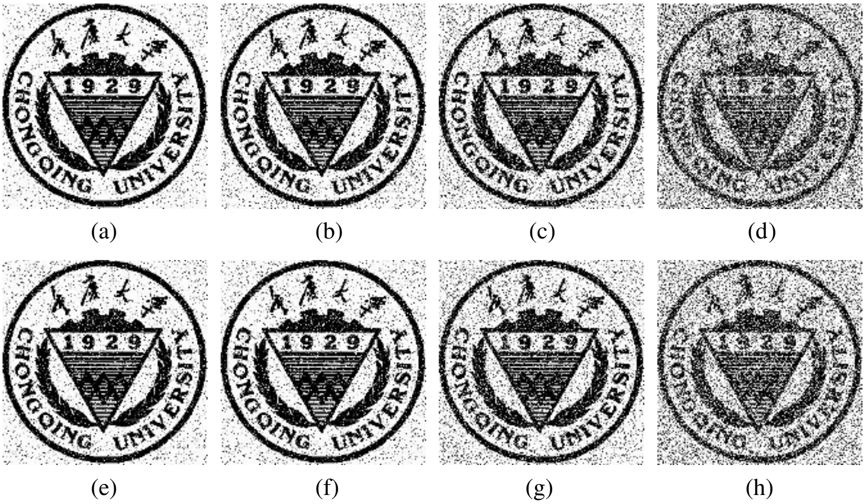


Fig. 3. Recovered watermarks from the cover images after various image processing manipulations (a) and (b) JPEG compression with quality factor = 50%, 10%, respectively. (c) and (d) Gaussian noise with zero mean and with variances σ = 0.001, 0.005, respectively. (e) 7 x 7 median filtering (f) scaling with rate = 0.8 (g) rotation with angle = 27 (h) cropping with ratio = 40%, i.e., crop 40% of the cover image from the bottom

The algorithm shows excellent resistant performances to JPEG compression and median filtering; good performances to scaling and rotation; and average performances to additive Gauss noise and cropping.

Some brief discussions about the algorithm:

(1) The robustness of the algorithm seems to be firmly related to the training goal in the neural network training process. We will do further study on the relationship between the trained goal and the algorithm robustness.

(2) Unlike [4], due to no watermark embedded, the neighborhoods of the selected pixels may be overlapped, which means more information can be 'embedded'.

(3) The main drawbacks of the proposed technique are the slow computation and the extra needed storage. The future research will try to improve them.

4 Conclusions

A neural network based lossless digital image watermarking algorithm is proposed in this paper. The algorithm extracts features from the cover image instead of embedding watermark, so the visual quality of the cover image keeps invariant. However, since without any watermark embedded in the cover image, an attacker may embed his information into the cover image and claim holding the copyright. Therefore, the lossless watermarking technique can only be used for piracy detection but copyright demonstration, unless it is cooperated with an authentication center.

Adrian Perrig *et al.* introduced the idea that used watermarks for piracy detection other than copyright demonstration in [6]: Since no watermarking method can withstand all possible attacks, we may shift the importance of watermarking from copyright protection to fraud detection by using watermarking as an efficient way of automatic matching; On the other hand, copyright protection may be achieved by registering the images at the copyright office as the traditional copyright protection method does. This is the motivation for us to develop such a algorithm.

References

1. Swanson, M.D., Kobayashi, M., Tewfik, A.H.: Multimedia Data-Embedding and Watermarking Technologies. Proceedings of the IEEE, **6** (1998) 1064-1087
2. Chang, C.-C., Hwang, K.-F., Hwang, M.-S.: A Block Based Digital Watermarks for Copy Protection of Images. In: Fifth Asia-Pacific Conference On Communications/Fourth Optoelectronics And Communications Conference, Beijing, October (1999)
3. Horng, G.-B., et al: http://www.csie.cyut.edu.tw/TAAI2002/TAAI2002PDF/Parallel%20 Session(C)/C2%20Neural%20Networks(III)/C2-5.pdf
4. Zhang, J., Wang, N.-C.: Neural Network Based Watermarking for Image Authentication (In Chinese). Journal of Computer-aided Design and Computer Graphics, **3** (2003) 307-312
5. Ding, K., He, C., Jiang, L.-G., Wang, H.-X: A Novel Fragile Watermark Applying in Verification. In: Proceedings of the International Conference on Neural Networks and Signal Processing, **2** Nanjing (2003) 1501-1504
6. Perrig A., Willmott, A.: Digital Image Watermarking in the 'Real World'. http://www.ece.cmu.edu/~adrian/projects/wmark-realworld/wmark-realworld.pdf

A Copy Attack Resilient Blind Watermarking Algorithm Based on Independent Feature Components*

Ju Liu[1], Huibo Hu[1], Jiande Sun[1], and Yu Huang[2]

[1] School of Information Science and Engineering, Shandong University
Jinan, Shandong 250100, China
{juliu,huibo_hu,jd_sun}@sdu.edu.cn
[2] Informatization Work Office, Shandong University, Jinan, Shandong 250100, China

Abstract. Digital watermarking is an enabling technique to prove the ownership of copyrighted digital products. But its utility may be reduced greatly by copy attack which can cause ownership ambiguity. In this paper, a blind digital watermarking algorithm resistant to copy attack is proposed. By adopting the independent component analysis (ICA) technique, the most stable independent feature component (IFC) of the host image can be obtained in which to embed the watermark. Symmetrically, the watermark detection is just the investigation of the marked image's IFCs without the host image. In addition, the threat of copy attack can be mitigated, because no watermark can be detected from the copy attacked image's IFCs. Theoretical analysis and simulation results show that the algorithm proposed here can resist copy attack as well as the common image manipulations.

1 Introduction

While the internet provides ready access to digital products, it also makes unauthorized copying relatively easy. As a feasible resolution, digital watermark can protect copyright in an open network environment. Copy attack is a kind of malicious watermark attacks, which copies the watermark from one legal marked image into another illegal non-marked image [1]. Conventionally, image signature could help cope with this attack, using which the legality of the marked image could be verified [2]. However, it might cause a loss of SNR. A watermarking scheme resisting copy attack directly was proposed in [3], in which no watermark could be detected from the copy attacked image. But it was just a semi-blind scheme.

Independent component analysis (ICA) is a novel signal processing and data analysis method in the research of blind source separation [4]. Some ICA-based watermarking systems used ICA in the watermark extraction [5], [6], but their robustness to copy attack would be wrecked because of their linear-embedding models. In addition, their detectors were not real blind. Others used ICA to extract the image

* This work was supported by the Excellent Young Scientist Award Foundation of Shandong (No. 01BS04).

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 777–782, 2005.

independent feature components (IFCs) in which to embed the watermark [7], [8]. However, the separating matrix obtained in the embedding was stored and directly used for detection, which would cause the failure of resistance to copy attack if the embedding strength was high enough.

In this paper, by analyzing the separating matrix calculated by ICA, the most stable IFC can be picked up, and afterwards is modified according to watermark. Therefore, in the detection only by performing ICA once again, the marked IFC can be found without accessing to the host image or the separating matrix calculated in the embedding. What is more, since different images possess different IFCs, no watermark could be detected from the copy attacked images.

2 Independent Component Analysis

The ICA process can derive features that best represent the image via a set of components that are as statistically independent as possible. The main assumption behind ICA is that any observation signal $\mathbf{X}=[x_1,x_2,\cdots,x_m]^T$ may be modeled as the linear mixture of the statistically independent source signal $\mathbf{S}=[s_1,s_2,\cdots,s_n]^T$ by an unknown $m\times n$ - dimensional mixing matrix $\mathbf{A}$.

$$\mathbf{X}=\mathbf{AS}\,. \tag{1}$$

The aim of ICA is to calculate the separating matrix $\mathbf{B}$ and recover the source by

$$\mathbf{Y}=\mathbf{BX}=\hat{\mathbf{S}}\,. \tag{2}$$

Joint approximate diagonalization of eigen-matrices (JADE) algorithm is a classical ICA algorithm [4], which will be used to extract the IFCs in this paper.

3 Watermarking Algorithm

3.1 Watermark Embedding

Firstly, the host image $\mathbf{G}_H$ is divided into m blocks, each block is expressed as a row vector $\mathbf{x}_{Hk}^T$, $k=1,\cdots,m$ and regarded as a component of the observation signal $\mathbf{X}_H$. Then performing ICA on $\mathbf{X}_H$, we have $\mathbf{S}_H=\mathbf{B}_H\mathbf{X}_H$, where $\mathbf{S}_H=\left(\mathbf{s}_{H1}^T;\mathbf{s}_{H2}^T;\cdots;\mathbf{s}_{Hm}^T\right)^T$, $\mathbf{X}_H=\left(\mathbf{x}_{H1}^T;\mathbf{x}_{H2}^T;\cdots;\mathbf{x}_{Hm}^T\right)^T$, and $\mathbf{s}_{Hk}^T, k=1,\cdots,m$ are the derived IFCs, $\mathbf{B}_H=\left(\mathbf{b}_{H1}^T;\mathbf{b}_{H2}^T;\cdots;\mathbf{b}_{Hm}^T\right)^T$ is the separating matrix.

If $\left\|\mathbf{b}_{Hp}^T\right\|_2=\min\limits_{1\le k\le m}\left\|\mathbf{b}_{Hk}^T\right\|_2$, then we select the p^{th} IFC $\mathbf{s}_{Hp}^T$ in $\mathbf{S}_H$ to perform the embedding, and now

$$\mathbf{s}_{Hp}^T=\mathbf{b}_{Hp}^T\mathbf{X}_H\,. \tag{3}$$

The stability of $\mathbf{s}_{Hp}^T$ will be discussed in section 4.

Let $\mathbf{c}_{Hp}^T$ denote the DCT coefficient vector of $\mathbf{s}_{Hp}^T$. And the watermark is $\mathbf{W} = \{W(i) | W(i) \in \{-1,+1\}, i = 1,\cdots,l\}$. In this paper, we only modify the signs of the intermediate frequency of $\mathbf{c}_{Hp}^T$ according to $\mathbf{W}$. If the marked DCT coefficient vector is $\mathbf{c}_{Wp}^T$, then the embedding process can be written as

$$\begin{aligned} c_{Wp}(K+i) &= \operatorname{sign}(W(i)) \cdot \left| c_{Hp}(K+i) \right|, \quad i = 1,\cdots,l \\ \mathbf{s}_{Wp}^T &= \operatorname{IDCT}(\mathbf{c}_{Wp}^T) \end{aligned} , \tag{4}$$

where K denotes the jumping-off point of embedding and is stored as a key, $c_{Wp}(K+i)$ is the $(K+i)^{th}$ element of $\mathbf{c}_{Wp}^T$, and $\mathbf{s}_{Wp}^T$ is the marked IFC which is the inverse DCT of $\mathbf{c}_{Wp}^T$. And $\operatorname{sign}(\cdot)$ is the sign function.

Finally, let $\mathbf{S}_W = (\mathbf{s}_{H1}^T;\cdots;\mathbf{s}_{Wp}^T;\cdots;\mathbf{s}_{Hm}^T)^T$, then by using $\mathbf{X}_W = \mathbf{B}_H^{-1}\mathbf{S}_W$ the watermark can be spread into all the blocks. And after combining them together, the marked image $\mathbf{G}_W$ can be obtained.

3.2 Watermark Detection

The watermark detection is notably symmetrical to the aforecited embedding procedures. Here, the received image $\mathbf{G}_R$ can be expressed as $\mathbf{X}_R$ by blocking, and in virtue of ICA its IFCs $\mathbf{S}_R = (\mathbf{s}_{R1}^T;\mathbf{s}_{R2}^T;\cdots;\mathbf{s}_{Rm}^T)^T$ can be calculated by $\mathbf{S}_R = \mathbf{B}_R\mathbf{X}_R$.

Analyzing $\mathbf{B}_R$, if $\left\|\mathbf{b}_{Rq}^T\right\|_2 = \min_{1\le k\le m}\left\|\mathbf{b}_{Rk}^T\right\|_2$, the q^{th} IFC $\mathbf{s}_{Rq}^T$ would be regarded as the marked IFC. After DCT, $\mathbf{s}_{Rq}^T$ is transformed into $\mathbf{c}_{Rq}^T$. Using K , the marked intermediate frequency coefficients could be picked up to extract the watermark $\mathbf{W}'$.

$$W'(i) = \operatorname{sign}(c_{Rq}(K+i)) \qquad i = 1,\cdots,l \, . \tag{5}$$

To measure the similarity between $\mathbf{W}$ and $\mathbf{W}'$, normalized cross-correlation (NC) is used as the objective evaluation criterion, which is defined as

$$\mathrm{NC} = \left(\sum_{i=1}^{l} W(i)\cdot W'(i)\right) \Big/ \sqrt{\sum_{i=1}^{l} W^2(i) \sum_{i=1}^{l} W'^2(i)} \, . \tag{6}$$

Comparing it with a predefined threshold, we can determine whether the received image is marked or not. And the false positive probability can be derived as

$$p_{fp} = \left(\operatorname{erf}\left(\frac{1-\mu}{\sqrt{2}\sigma}\right) - \operatorname{erf}\left(\frac{Th-\mu}{\sqrt{2}\sigma}\right)\right) \Big/ \left(\operatorname{erf}\left(\frac{1+\mu}{\sqrt{2}\sigma}\right) + \operatorname{erf}\left(\frac{1-\mu}{\sqrt{2}\sigma}\right)\right) . \tag{7}$$

Details of the equation can be found in [3]. In this paper, the threshold $Th = 0.1$, which can ensure $p_{fp} \approx 1.7\times 10^{-5}$.

4 Performance Analysis

4.1 Stability Analysis of IFCs

In (3), because the norm of $\mathbf{b}_{Hp}^T$ is minimal, when $\mathbf{X}_H$ is interfered, the influence to $\mathbf{s}_{Hp}^T$ is also the smallest. Thus better robustness can be gained. Here, $\mathbf{s}_{Hp}^T$ may not be the sparsest. Following equation can be used to measure the stability of IFCs.

$$\text{Std} = \sqrt{\sum_k \left[\mathbf{s}_{Hi}^T(k) \Big/ \left\|\mathbf{s}_{Hi}^T\right\|_2 - \mathbf{s}_{Rj}^T(k) \Big/ \left\|\mathbf{s}_{Rj}^T\right\|_2\right]^2} \; . \tag{8}$$

where $\mathbf{s}_{Hi}^T$ and $\mathbf{s}_{Rj}^T$ are a pair of corresponding IFCs.

Table 1 shows the stability comparison of different IFCs under some common attacks, where (a) is our method, (b) chooses the largest-energy IFC, and (c) chooses the smallest-energy IFC. The Std is smaller, the IFC is more stable against the common attacks. Because of the blind extraction and the indeterminacies of ICA [4], the extracted IFCs in (b) and (c) may not correspond to the original ones, which means they may not be the real marked IFCs. In Table 1, 'Y' means correct correspond, while 'N' means wrong. The data show that it is the wrong correspond that mainly causes the fall of stability.

Table 1. Stability comparison of IFCs

	Gaussian noise		Wiener filtering		Cropping		JPEG	
(a)	0.1552	Y	0.0776	Y	0.1035	Y	0.0628	Y
(b)	0.1435	Y	0.1663	N	0.1816	N	0.0366	Y
(c)	1.5052	N	1.4043	N	1.3878	N	1.4129	N

4.2 Robustness Analysis of Watermark

In copy attack introduced in [1], the attacker estimates the host image $\hat{\mathbf{G}}_H$ from the marked image $\mathbf{G}_W$ and then gets the estimated watermark $\hat{\mathbf{W}} = \mathbf{G}_W - \hat{\mathbf{G}}_H$. By mixing $\hat{\mathbf{W}}$ with another non-marked image $\mathbf{G}_I$ linearly, the copy attacked image $\mathbf{G}_C$ could be got by $\mathbf{G}_C = \mathbf{G}_I + a\hat{\mathbf{W}}$, where a is the embedding strength.

Under common image manipulations, we have $\mathbf{G}_R \approx \mathbf{G}_W$. Hence, $\mathbf{G}_R$ and $\mathbf{G}_W$ can be regarded as different mixtures of the same IFCs. However, when copy attack happens, $\mathbf{G}_R$ and $\mathbf{G}_W$ are obviously different, so are their IFCs. That means we can extract watermark from the usual attacked images but the copy attacked image.

In [7] and [8], $\mathbf{B}_H$ is assisted to perform the detection in stead of $\mathbf{B}_R$, so better correspond of IFCs can be kept and higher robustness to common attacks can be obtained. But we avoid using them here, because of the following factors: (a) using $\mathbf{B}_H$ and p can endanger the security of this system; (b) copy attack can result in the invalidity of these algorithms.

5 Computer Simulation Experiments

The host image is standard ‘Lena’ image, which is divided into 16 equal blocks to derive the IFCs. Here, we randomly generate 300 sequences with length of 1024 and value ± 1, and embed the 200^{th} into the host image. The PSNR between the host image and the marked image is maintained at 62.99dB.

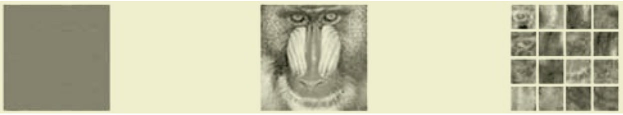


Fig. 1. (a) Difference (b) Copy attacked image (c) IFCs

In this experiment, the difference (see Fig. 1(a)) between the host image and the marked image is added directly into ‘Baboon’ image ($a = 2$) to get the copy attacked image, shown in Fig. 1(b). And the PSNR between the original ‘Baboon’ image and Fig. 1(b) is 55.95dB. Fig. 1(c) is the IFCs extracted from Fig. 1(b), which are obviously different from those of ‘Lena’ image. And now the tested NC from Fig. 1(b) is 0.0039, which means no watermark can be detected from the copy attacked image. In addition, we choose 3 standard images and 3 natural images to test the robustness of our algorithm against copy attack. The calculated NC values are shown in Table 2.

Table 2. Robustness to copy attack

	Couple	Peppers	Goldhill	Natural 1	Natural 2	Natural 3
NC	0.0938	0.0840	-0.0566	0.0059	0.0430	0.0176

Table 3 shows the detection results under common image manipulations, such as (a) Gaussian noise, (b) 3×3 median filtering, (c) cropping, (d) scaling, (e) JEPG compression, and (f) histogram equalization. It indicates that under above attacks the watermark can still be detected successfully from the attacked images.

Table 3. Detection results under common attacks

	(a)	(b)	(c)	(d)	(e)	(f)
NC	0.2402	0.1328	0.5645	0.6816	0.1660	0.7578
PSNR(dB)	29.0581	49.6126	40.0337	51.6788	49.7852	29.0675

Fig. 2 is the robustness comparison under above common image manipulations, where (A) is our algorithm, (B) uses $\mathbf{B}_H$ and p to extract the watermark [7], [8]. The x-axis 1-6 denote the above attacks (a)-(f). In this experiment, (B) shows better robustness. Fig. 3 indicates the detection result under copy attack of method (B). Here, the tested NC is 0.3125, obviously above the predefined Th. It means (B) can not resist copy attack. And these results also mean in our method the resistance to copy attack and the realization of the blind detection are achieved at the cost of the robustness to common image manipulations.

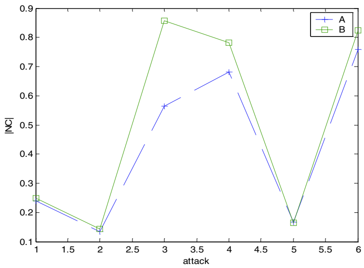


Fig. 2. Comparison on robustness

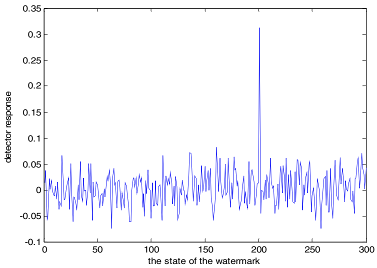


Fig. 3. Detection result in (B) under copy attack

6 Conclusions

In this paper, we investigate a blind watermarking algorithm against copy attack. By utilizing to ICA technique, the IFC that is interfered least by attacks can be picked up, in which to embed the watermark. And performing ICA on the marked image, the watermark can be detected blindly. Meanwhile, the resistance to copy attack can also be realized. However, because of the indeterminacies in ICA and the "blind" request, the robustness of this algorithm to the common image manipulations may be influenced. How to further mitigate the influences is our future work.

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. Lu, C., Liao, H. M., Kutter, M.: Denoising and Copy Attacks Resilient Watermarking by Exploiting Prior Knowledge at Detector. IEEE Trans. on Image Processing, **11** (2002) 280-290
4. Cardoso, J. F., Souloumiac A.: Blind Beamforming for Non-Gaussian Signals. IEEE Proceedings-F, **140** (1993) 362-370
5. Yu, D., Sattar, F., Ma, K. K.: Watermark Detection and Extraction using Independent Component Analysis Method. EURASIP JASP, **1** (2002) 92-104
6. Liu, J., Zhang, X., Sun, J., Lagunas, M. A.: A Digital Watermarking Scheme based on ICA Detection. ICA2003, (2003) 215-220
7. Conzález-Serrano, F. J., Murillo-Fuentes, J. J.: Independent Component Analysis Applied to Digital Image Watermarking. ICASSP '01, **3** (2001) 7-11
8. Liu, J., Sun, J., Du Z., Wan Y.: Embodying Information into Images by an MMI-Based Independent Component Analysis Algorithm. ICSP'02, **2** (2002) 1600-1603

Watermarking Capacity Analysis Based on Neural Network

Fan Zhang and Hongbin Zhang

Institute of Computer, Beijing University of Technology, Beijing 100022, China
bpuzf@sohu.com

Abstract. The watermarking capacity of a digital image is an evaluation of how much information can be hidden within digital images. This paper presents a blind watermarking algorithm based on the Hopfield neural network and discusses the bounds of watermark information. The Hopfield neural network is used to store the original image during the watermark embedding and to retrieve it during the watermark extracting. According to the research, the attraction basin of neural network determines the watermarking capacity.

1 Introduction

The watermarking capacity of a digital image is an evaluation of how much information can be hidden within images. The watermarking can be considered as a communication process. The image is the communication channel in which the watermark is transmitted. The watermarking capacity corresponds to the communication capacity of the "watermark channel"[1],[2]. Recently, some works on watermarking capacity have been presented. Almost all the previous works used this information-theoretic model. 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 watermarking capacity of digital image in DCT and DFT domain[4]. Moulin's work introduces a game-theoretic approach for the evaluation of watermarking capacity problem under attacks[5],[6]. Lin presents a capacity analysis of zero-error information hiding in 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].

This paper presents a blind watermarking algorithm based on the Hopfield neural network and discusses the bounds of watermark information.

The rest of this paper is organized as follows. In Section 2, a neural network based blind watermarking algorithm is presented. In Section 3, the watermarking capacity of this algorithm is discussed. The experiments results are shown in Section 4. Section 5 gives the conclusion of this paper.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 783–788, 2005.

2 Blind Watermarking Algorithm

In this watermarking algorithm, a Hopfield neural network is used to store the original image during the watermark embedding process. And the same neural network is used to retrieve the image during the watermark extraction process.

Watermark: The watermark is a binary sequence, which is generated by a pseudo-random sequence generator. The length of a watermark sequence is less than or equals to the number of pixels.

Neural network learning: In order to store a standard gray test image using Hopfield neural network, the original image is decomposed into eight bit-planes. A discrete Hopfield network is trained with the eight bit-planes are the input patterns. The number of neurons equals to the number of pixels. The neural network learns and stores the original image. The watermark sequence can also be stored at the same time if the original watermark is necessary in the watermark detection.

Watermark embedding: The watermark is embedded into an image according to the bits of the watermark sequence. Each bit of the watermark 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 watermark invisible. In this algorithm, the magnitude of modification is simply added to the amplitude of corresponding pixels.

Watermark extraction: Firstly, the neural network retrieves the original image from the watermarked image or the noised image. Then, the watermark data are extracted by subtracting the retrieved image from the watermarked image. Finally, the watermark sequence is reconstructed according to a threshold. This algorithm is essentially blind watermarking so that the original image is not needed during the extraction process.

3 Watermarking Capacity Analysis

3.1 Basin of Attraction

The Hopfield network is a recurrent neural network that stores information in a dynamically stable configuration. 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. Once a pattern is stored, the Hopfield network can reconstruct the original pattern from the degraded or incomplete pattern.

The Hamming distance is the number of components that differ between two vectors. The distance between two vectors S^1 and S^2 is $d_h(S^1, S^2)$. The r-Hamming sphere contains all possible vectors that the Hamming distance to S^p is less than r,

$$B_r(S^p) = (S^q | d_h(S^p, S^q) = r). \tag{1}$$

The Hamming distance or the Hamming sphere can be used to measure the basin of attraction. Let P denotes the number of stored patterns and N denotes the number of neurons. The Hopfield network model can be expressed as,

$$x_i^{t+1} = sgn\left[\sum_{j\neq i}^{N} W_{ij} x_j^t\right], \tag{2}$$

where x_i^t is the state of neurons at time t; sgn is the sign function. The weight matrix can be computed according to the Hebb rule,

$$W_{ij} = \frac{1}{N}\sum_{k=1}^{P} u_i^k u_j^k, \tag{3}$$

where u^1, u^2,..., u^P denote stored patterns of neural network.

Let $X^0 = \{x_1^0, x_2^0, \ldots, x_N^0\}^T$ denotes the initial state of neural network, $X^t = \{x_1^t, x_2^t, \ldots, x_N^t\}^T$ denotes the state at time t. If assume that the probe pattern is one of the stored patterns, then:

$$\begin{aligned}\sum_{j\neq i}^{N} W_{ij} x_j^0 &= \sum_{j\neq i}^{N} \frac{1}{N}\sum_{k=1}^{P} u_i^k u_j^k x_j^0 \\ &= \frac{1}{N}\sum_{k=1}^{P}\left[\left(\sum_{j=1}^{N} u_i^k u_j^k x_j^0\right) - u_i^k u_j^k x_i^0\right] \\ &= \frac{1}{N}\left[u_j^k (u^k)^T X^0 + \sum_{l\neq k}^{P} u_i^l (u^l)^T X^0 - P x_i^0\right].\end{aligned} \tag{4}$$

Because $(u^k)^T X^0 = N - 2d_h(X^0, u^k)$ and $(N-P)/2P < N/2$, If assume x_i are orthogonal for each other,

$$-2d_h(X^0, u^k) \leq (u^l)^T X^0 \leq 2d_h(X^0, u^k). \tag{5}$$

When $d_h(X^0, u^k) < (N-P)/2P$,

$$N - 2d_h(X^0, u^k) - 2(P-1)d_h(X^0, u^k) - P > 0. \tag{6}$$

So, when $u_i^k = +1$,

$$\begin{aligned}\sum_{j\neq i}^{N} W_{ij} x_j^0 &= \frac{1}{N}\left[u_j^k (u^k)^T X^0 + \sum_{l\neq k}^{P} u_i^l (u^l)^T X^0 - P x_i^0\right] \\ &> \frac{1}{N}\left[N - 2d_h(X^0, u^k) - 2(P-1)d_h(X^0, u^k) - P\right] > 0.\end{aligned} \tag{7}$$

When $u_i^k = -1$,

$$\sum_{j \neq i}^{N} W_{ij} x_j^0 < \frac{1}{N} \left[-N + 2d_h(X^0, u^k) + 2(P-1) d_h(X^0, u^k) + P \right] < 0. \tag{8}$$

So, $u_i^k = sgn \left[\sum_{j \neq i}^{N} W_{ij} x_j^0 \right]$, and then,

$$x_i^1 = sgn \left[\sum_{j \neq i}^{N} W_{ij} x_j^0 \right] = u_i^k. \tag{9}$$

According to Eq. 9, if the Hamming distance between the probe pattern and a stored pattern,

$$d_h \leq \frac{N - P}{2P}, \tag{10}$$

the neural network will converge to this stored pattern.

3.2 Bounds of the Watermark Information

In the watermarking schemes, a watermark can be viewed as noise that pollutes the original image. The modification of amplitude of some pixels will appear as changes in the image. This means that the image is polluted. The more watermark data are embedded, the bigger Hamming distance is between the watermarked image and the original image. When the Hamming distance is out of the bounds of the attraction basin, the neural network can not retrieves the original image correctly. So, the basin of attraction restricts the number of points that can be modified in an image, therefore, restricts the capacity of watermarking.

According to this watermarking algorithm, there are eight stored patterns, $P = 8$. If the size of a test image is 256×256, the number of neurons is 65,536, according to Eq. 10, $d_h \leq 4,096$. If the number of modified points in a bit-plane is less than d_h, the neural network can reconstructs this bit-plane successfully. So, the maximum number of modifiable points in each bit-plane is 4,096.

In order to keep the invisibility of watermarking, not all of the eight bit-planes can be modified. For example, the modification of the most significant bit-plane (MSB) will result in the human-perceptible change of the original image. Generally, the maximum watermark magnitude is no more than 30. At most five bit-planes can be modified in the watermark embedding. In an extreme case, the modified points of five bit-plane may differ from each other, then the maximum number of watermarked pixels is $n = 5 \times 4,096 = 20,480$.

In this watermarking algorithm, the watermark is a binary sequence. Each bit of the sequence is random (0 or 1). There are 2^n combinations in an n-length binary sequence. The probability of each combination is $1/2^n$. According to the information theory, the information of an n-length binary sequence is,

$$C = -log_2(1/2^n). \tag{11}$$

So, in this watermarking algorithm, the maximum watermark information is 20,480 bits.

4 Experiments

In experiments, the 256 × 256 standard gray test images are used. The watermarked Baboon image and the reconstruction image are shown in Fig. 1. The peak signal to noise ratio (PNSR) is 33.26 dB.

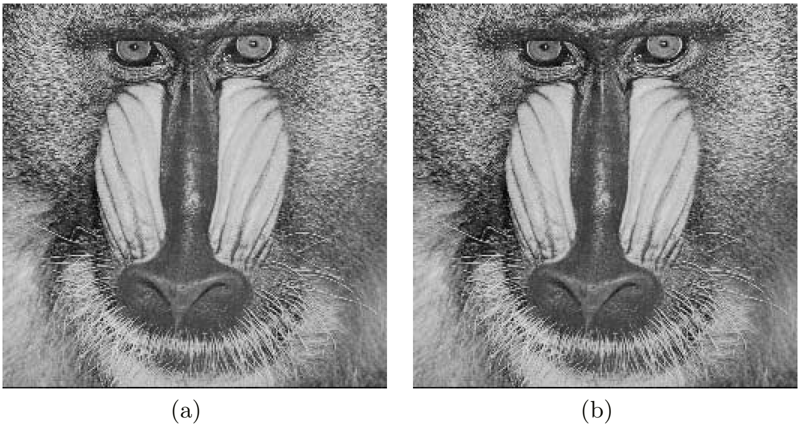

(a) (b)

Fig. 1. The watermarked Baboon image (a) and the reconstruction image (b)

Most of reported works on the watermarking capacity are based on an information theoretic model. Because of the different analytical methods used, and the different experimental conditions and parameters, the results of watermarking capacity of those works are quite diverse. The reported watermarking capacity of a 256×256 image is from about 20,000 bits to 80,000 bits. The result of our watermarking capacity is relatively small compared to those results. We think that the smaller watermark information is due to the limited competence of a neural network.

5 Conclusions

This paper presents a blind watermarking algorithm based on the Hopfield neural network and discusses the bounds of watermark information. The watermark can be viewed as noise that pollutes the original image. The more watermark data are embedded, the bigger Hamming distance is between the watermarked image and the original image. When the Hamming distance is out of the bounds of the attraction basin, the neural network could not retrieve the original image correctly. So, the basin of attraction restricts the number of points that can be modified in an image, therefore, restricts the bounds of the watermark information.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. 60075002 and the National High Technology Research and Development 863 Program of China under Grant No. 2003AA144080.

References

1. Cox, I., Kilian, J., Leighton, F., Shamoon, T.: Secure spread spectrum watermarking for multimedia. IEEE Transactions on Image Processing, **6** (1997) 1673–1687
2. Cox, I., Milller, M., McKellips, A.: Watermarking as communications with side information. Proceedings of the IEEE, Special Issue on Identification and Protection of Multimedia Information, **7** (1999) 1127–1141
3. Servetto, S., Podilchuk, C., Ramchandran, K.: Capacity Issues in Digital Image Watermarking. In: IEEE International Conference on Image Processing, Vol. 1, New York (1998) 445–449
4. Barni, M., Bartolini, F., De Rosa, A., Piva, A.: Capacity of the watermarking-channel: how many bits can be hidden within a digital image. In: Wong, P., Delp, E. (eds.): Security and Watermarking of Multimedia Contents, Proceedings of SPIE, Vol. 3657, San Jose, USA (1999) 437–448
5. Moulin, P., Mihcak, M.: A Framework for Evaluating the Data-Hiding Capacity of Image Sources. IEEE Transactions on Image Processing, **9** (2002) 1029–1042
6. Moulin, P.: The Role of Information Theory in Watermarking and Its Application to Image Watermarking. Signal Processing, **6** (2001) 1121–1139
7. Lin, C., Chang, S.: Zero-error Information Hiding Capacity of Digital Images. In: IEEE International Conference on Image Processing, Vol. 3, New York, USA (2001) 1007–1010
8. Zhang, F., Zhang, H.: Digital watermarking capacity and reliability. In: Bichler, M., Chung, J. (eds.): IEEE International Conference on e-Commerce Technology, San Diego, CA, USA, (2004) 295–298
9. Zhang, F., Zhang, H.: Wavelet Domain Watermarking Capacity Analysis. In: 12th European Signal Processing Conference, Vol. 3, Vienna, Austria, (2004) 1469–1472

SVR-Based Oblivious Watermarking Scheme

Yonggang Fu[1,3], Ruimin Shen[1], Hongtao Lu[1], and Xusheng Lei[2]

[1] Deperment of Computer Science and Engineering, Shanghai Jiaotong Univ.
Shanghai 200030, China
[2] Department of Automation, Shanghai Jiaotong University, Shanghai China, 200030
[3] Department of Software, Xiamen University, Xiamen, Fujian 361000, China
{fyg,rmshen,htlu,xushenglei}@sjtu.edu.cn

Abstract. With the development of Internet, digital medias can be easily redistributed which renders copyright protection issues. A novel support vector regression (SVR) based oblivious watermarking scheme for color image is proposed in this paper. The watermark is embedded into the blue channel of color images by applying the good learning ability of SVR. Using the information of unmodified reference positions, the SVR can be trained well. Thanks to the good generalization ability of SVR, the watermark can be correctly extracted under several different attacks. Experimental results show that the proposed scheme have outperforming performance over Kutter's method against several different attacks including noise addition, shearing, luminance & contrast enhancement, distortion, etc.

1 Introductions

Recently, efforts are made to take advantage of machine learning techniques for watermark embedding and extraction. Neural Networks are firstly introduced into watermarking in [1], which makes the watermark detection more robust against common attacks. Genetic algorithm is proposed for the selection of the best embedding positions in block based DCT domain watermarking [2]. In [3] we have introduced support vector machine (SVM) for watermark detection in the spatial domain, which shows good robustness.

In this paper, a novel oblivious watermarking scheme based on SVR is proposed. The watermark is embedded into the spatial domain by the aids of the trained SVR. The watermark extraction can be finished with the help of the well-trained SVR. The experimental results show good robustness against different attacks, but with better image quality than common watermarking schemes in the spatial domain.

2 Proposed Watermarking Scheme

SVM is a universal classification algorithm developed by Vapnik and his colleagues [4], [5]. SVR is an application of SVM on function approximation.

For a natural image, since every pixel usually has high correlation with its neighbors, then central pixel value can be predicted by its neighbors. When the image undergoes some attacks, such as image processing, or distortion, this relationship

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 789–794, 2005.

leaves unchanged or changes a little. So we can apply this feature for watermark embedding and extraction

Let I be a color image with size $M \times N$, denoted by $I=[R,G,B]$, where R,G,B are the three image components corresponding to red, green, and blue channel, respectively. Suppose $p=(i,j)$, $i=1,2,...,M$ and $j=1,2,...,N$ be the pixel position for further image accessing. The watermark embedding scheme can be summarized as following five steps:

Step1. Firstly, the logo watermark is permuted according to copyright owners key K_1 and reshaped into line order, denoted as: $W=\{w_i\}|_{i=1..n}$, where n is the watermark length.

Step2. Suppose $\{p_t=(i_t,j_t)\}|_{t=1,2,....L+n}$ be the randomly selected position sequence according to another secret key K provided by the copyright owner. The position sequence proposed here is composed of two subsequences, one part $\{p_t=(i_t,j_t)\}|_{t=1,2,....L}$ is called reference position sequence that is not modified in the embedding process, and whose role is to train the SVR. The remaining part $\{p_t=(i_t,j_t)\}|_{t=L+1,L+2,....L+n}$ is called embedding positions, which is used for embedding the true watermark. The main reason for the selection of reference position sequence is to maintain the same trained SVR in the embedding and extraction procedure.

Step3. Training the SVR

For each reference position $p_t=\{i_t,j_t\}$, we can form the training data in every 3×3 window and reach the training dataset as follows:

$$\begin{aligned}D=\{d_t\,|\,d_t=[&B(i_t-1,j_t-1),B(i_t-1,j_t),B(i_t-1,j_t+1),B(i_t,j_t-1),\\&B(i_t,j_t+1),B(i_t+1,j_t-1),B(i_t+1,j_t),B(i_t+1,j_t+1)]^T,t=1,2,...L\}\end{aligned} \tag{1}$$

where $B(\cdot,\cdot)$ is the pixel value in the blue channel of the original color image. After we set the objective $O=\{B(i_t,j_t)\}|_{t=1,2,...L}$ of this dataset, the SVR can be trained:

$$f(x)=\sum_{t=1}^{L}(\alpha_t-\hat{\alpha}_t)Ker(d_t,x)+b \tag{2}$$

where $\alpha_t,\hat{\alpha}_t$ are trained coefficients, and b is the bias.

Step4. Watermark embedding

For each embedding positions, we first collect the predicting dataset in the blue channel of host image:

$$\begin{aligned}P=\{d_t\,|\,d_t=[&B(i_t-1,j_t-1),B(i_t-1,j_t),B(i_t-1,j_t+1),B(i_t,j_t-1),\\&B(i_t,j_t+1),B(i_t+1,j_t-1),B(i_t+1,j_t),B(i_t+1,j_t+1)],t=L+1,L+2,....L+n\}\end{aligned} \tag{3}$$

Then, by the trained SVRwe can find the predicting value at each embedding position:

$$y_t=f(d_t),\ t=L+1,L+2,....L+n \tag{4}$$

Comparing the predicted pixel values and the actual ones, the watermark bits are embedded in following manner:

$$B(i_t, j_t) = y_t + \alpha_t (2w_{t-L} - 1) \qquad t = L+1, L+2, ..., L+n \tag{5}$$

where α_t is the embedding strength.

To maximize the admissible amplitude of the imperceptible information embedded, we vary α_t according to the characteristics of the host image. This is due to the feature of human visual system (HVS) [6]:

$\alpha_t = \beta L_{p_t}$, where β is an embedding constant controlling the compromise between robustness and visual quality of the watermarked image, and

$L_{p_t} = 0.299 R_{p_t} + 0.587 G_{p_t} + 0.114 B_{p_t}$, is the luminance component of the RGB image at position p_t .

Step5. The watermarked blue channel is integrated with red and green ones, and forms the final watermarked image.

Watermark Extraction

The watermark extraction is similar to the embedding procedure.

Step1. The randomly selected position sequence $\{p_t = (i_t, j_t)\}|_{t=1,2,....L+n}$ is generated again according to the same secret key K as used in the embedding process.

Step2. For each reference position $p_t = \{i_t, j_t\}$, we can extract the training dataset from the possibly tampered image:

$$\begin{aligned}\tilde{D} = \{\tilde{d}_t \mid \tilde{d}_t = [&\tilde{B}(i_t - 1, j_t - 1), \tilde{B}(i_t - 1, j_t), \tilde{B}(i_t - 1, j_t + 1), \tilde{B}(i_t, j_t - 1), \\ &\tilde{B}(i_t, j_t + 1), \tilde{B}(i_t + 1, j_t - 1), \tilde{B}(i_t + 1, j_t), \tilde{B}(i_t + 1, j_t + 1)]^T, t = 1,2,....L\}\end{aligned} \tag{6}$$

By applying the objective $O = \{\tilde{B}(i_t, j_t)\}|_{t=1,2,...L}$ of this dataset, the SVR can be trained:

$$f(x) = \sum_{t=1}^{L} (\alpha_t - \hat{\alpha}_t) Ker(\tilde{d}_t, x) + b \tag{7}$$

Step3. Watermark extraction. For each embedding positions, we first collect the predicting dataset:

$$\begin{aligned}\tilde{P} = \{\tilde{d}_t \mid \tilde{d}_t = [&\tilde{B}(i_t - 1, j_t - 1), \tilde{B}(i_t - 1, j_t), \tilde{B}(i_t - 1, j_t + 1), \tilde{B}(i_t, j_t - 1), \\ &\tilde{B}(i_t, j_t + 1), \tilde{B}(i_t + 1, j_t - 1), \tilde{B}(i_t + 1, j_t), \tilde{B}(i_t + 1, j_t + 1)], t = L+1, L+2, ...L+n\}\end{aligned} \tag{8}$$

Then, we can find the predicting value at each embedding position:

$$y'_t = f(\tilde{d}_t), \; t = L+1, L+2, ...L+n \tag{9}$$

The watermark bits can be extracted by comparing the relation between the predicted pixel value in the blue channel and the actual pixel value:

$$w'_{t-L} = \begin{cases} 1 & if \;\; \tilde{B}(i_t, j_t) > y'_t \\ 0 & else \end{cases} \quad t = L+1, L+2, ..., L+n \tag{10}$$

Step4. The extracted watermark data is inversely permuted with the same key K_1 as used in the embedding process and reorganized into the logo image. Then the watermark detection results can be decided visually by comparing with the given watermark logo or by BER (Bit Error Rate) defined as follows:

$BER = (\sum_{i=1}^{n} w_i \oplus w'_i) \big/ n$, where $\oplus$ is binary XOR operation.

3 Experimental Results

We have tested the proposed scheme on different images including “F16”, “Lena”, and “Pepper”, which represent different texture complexity. Due to the limitation of space, here we only demonstrate the experimental results on “F16”.

Firstly, we should decide some parameter in the experiment. For SVR, the kernels is decided from the generally used candidate kernels, including “RBF”, “Polynomial”, and “Linear”. We have tested the regression performance on different parameters of these kernels, and find that the “RBF” kernels outperform others. With the increasing of the width parameter σ for RBF kernel, the regression performance becomes better when $\sigma < 300$ and decreases when $\sigma > 300$. So we select the width parameter $\sigma = 300$. Since the penalty parameter C in the SVR training has little influence on the trained outcome, we set it as 100 here. As to the length of the reference position sequence, a small number of train patterns can’t train the SVR well, but too much training patterns may cause over-fitting. At the same time, a long vacant reference position sequence may be helpful to improve the quality of the watermarked image. When we set the training pattern number 100, the SVR can be trained well.

The original image what we tested is 512*512 color image (Fig1.(a)). The watermark is a 32*25 binary logo shown in Fig.1(c). Fig.1 (b) depicts the watermarked image with PSNR42.3dB, which can not be distinguished from the original one by human eyes. Kutter [7] has proposed a watermarking scheme in spatial domain which applies the relationship between the neighbor pixels. The comparative results with Kutter’s method against different attacks are shown in following details. Under the same watermarking strength, due to the SVR’s prediction error, the PSNR of watermarked image generated by our method is a little lower than Kutter’s. For fair comparison, we adjust the embedding strength manually. In the comparison, the strength is 0.4 and 0.43 for our method and Kutter’s respectively.

When there is no attack, the extracted watermarks are shown in Fig.1(d) and (e) respectively. Our method can extract the watermark without any bit errors (Fig.1(d), whereas there are about 38 bit errors in extracted watermark by Kutter’s method(Fig.1(e)).

Due to the limitation of space, we only demonstrate two visual experimental results, including luminance & contrast enhancement and shearing. When the image is enhanced in luminance and contrast with 70%, Fig.2(a)(b) show the extracted watermarks. The watermark extracted by our method has only 23 bit errors, while the wa-

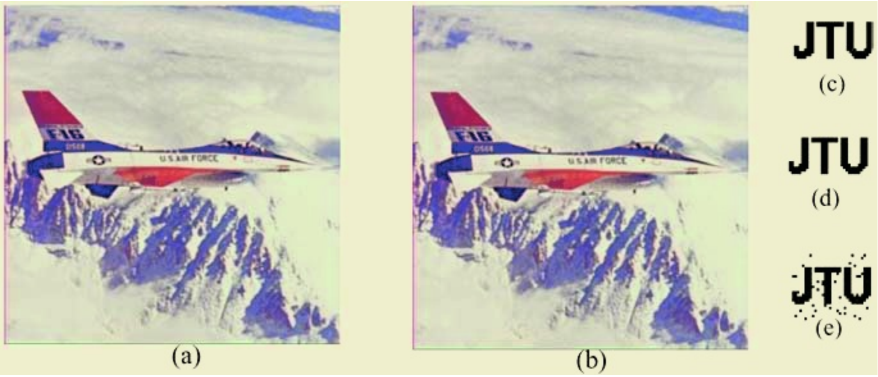


Fig. 1. (a) Original F16 image(512*512); (b) Watermarked image(PSNR=42.3dB); (c) Watermark; (d) Extracted watermark by our method; (e) Extracted watermark by Kutter's method

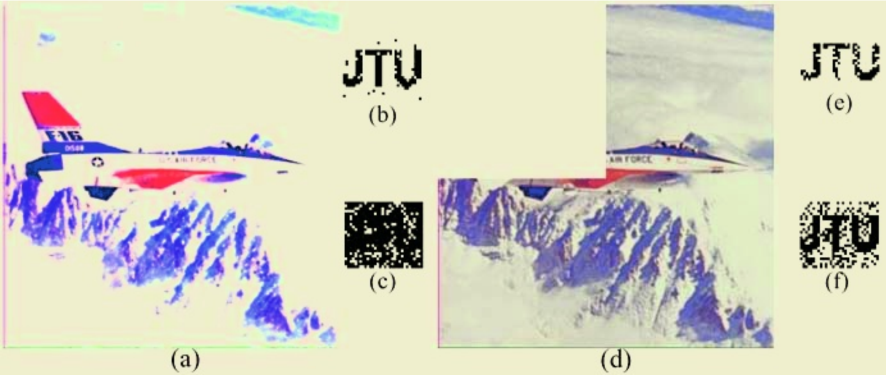


Fig. 2. (a) Luminance &contrast enhanced image; (b) Extracted watermark from(d) by our method(BER 10); (c)Extracted watermark from(d) by Kutter's method(BER 457); (d) Shearing attacked image; (e) Extracted watermark from(a) by our method(BER 57); (f)Extracted watermark from (a) by Kutter's method(BER 210)

termark extracted by Kutter's method has totally about 457 bit errors. Fig.2(d) shows the sheared image with 3/4 left. Our method can extract the watermark with bit errors 57, and the watermark extracted by Kutter's method has about 210 bit errors. Some other quantity results are tabulated in Table 1. When the watermarked image is Jpeg compressed or filtered, the performance of the proposed method is not better than Kutter's one. The main reason is that the neighbor pixels relations are destroyed to some extent when the image is filtered or compressed. But in most situations, the performance of the proposed watermarking scheme is superior to Kutter's.

4 Conclusions

In this paper, we proposed a SVR based oblivious watermarking scheme. Since the natural image pixels have high correlations with it neighbors, these relation can be

Table 1. Bit error rate(BER) comparison under different attacks

Attacks	PSNR(dB)	Proposed method	Kutter's method
Attack Free	42.3	0	0.0475
Salt & Pepper(0.04)	19.08	0.0350	0.0737
Luminance 70% & contrast 70% enhancement	7.53	0.0125	0.5713
Shearing 25%	11.45	0.0712	0.2625
Distortion 5°	25.15	0.2275	0.4025
Scaling 150%	43.06	0	0.0488
Scaling 50%	33.04	0.0605	0.08
Jpeg compression QF=80	33.05	0.2712	0.2725
Jpeg compression QF=50	31.60	0.3625	0.3425
Filtering(3*3)	32.69	0.0675	0.0525
Mosaic(3*3)	28.34	0.0762	0.1313
Bluring	38.26	0	0.0537

predicted by the training of SVR. When the watermarked image undergone some attacks, with the aids of the information comes from the vacant reference positions, the relationship among the neighbor pixels can be learnt well by the training of the SVR and these information are helpful for the extraction of the embedded watermark. Due to the good generalization ability of SVR, the watermark can be correctly extracted unless the watermarked image is attacked severely. The experimental results under several attacks compared with Kutter's method show good robustness of the proposed method.

References

1. Yu, P.-T., Tsai, H.-H., Lin, J.-S.: Digital Watermarking Based on Neural Networks for Color Images. Signal Processing, **81** (2001) 663-671
2. Shieh, C.S., Huang, H.C., Wang, F.H., Pan, J.S.: Genetic Watermarking Based on Transform-Domain Technique. Pattern Recognition. **37** (2004) 555-565
3. Fu, Y.G., Shen, R.M., Lu, H.T.: Watermarking Scheme Based on Support Vector Machine for Colour Images. Electronics letters, **40** (2004) 986-987
4. Vapnik, V.: The Nature of Statistical Learning Theory. New York, Springer-Verlag (1995)
5. Vapnik, V.: Statistical Learning Theory. John Wiley. New York (1998)
6. Swanson, M.D., Kobayashi, M., Tewfik, A.H.: Multimedia Data Embedding and Watermarking Technologies. Proceedings of IEEE, **86** (1998) 1064-1087
7. Kutter, M., Jordan, F., Bossen, F.: Digital Signature of Color Images Using Amplitude Modulation. J. Electronics Imaging, **7** (1998) 326-332

An Audio Watermarking Scheme with Neural Network

Chong Wang, Xiaohong Ma, Xiangping Cong, and Fuliang Yin

School of Electronic and Information Engineering
Dalian University of Technology, Dalian, Liaoning 116023, China
maxh@dlut.edu.cn

Abstract. An audio watermarking scheme with neural network is presented in this paper. The hiding watermark, which is the combination of the chaotic sequence and the watermark sequence related to original watermark image, is embedded into the DCT coefficients of the original audio signal. Meanwhile, to improve the ability of de-synchronization attack, the synchronous code is embedded into the original audio signal in the time domain. To extract the watermark sequence, first we select DCT coefficients corresponding to the pseudorandom sequence as the training sample, which can be used to train the neural network. Then the DCT coefficients relating to the watermark sequence is taken as the validation sample. Experimental results show that the proposed method has good robustness under general signal manipulations.

1 Introduction

The widespread use of the Internet, proliferation of the low-cost and reliable storage devices, and deployment of long-distance Gbps networks has made the copyright protection of digital audio works more and more important and urgent. Therefore, a robust audio watermarking technology to protect copyrighted digital audio datum from illegal sharing and tampering is needed. The audio watermarking techniques proposed so far can be divided into two main groups: those embed the watermark directly in the time domain and those operate in the transformed domain. The representative time domain schemes are the Least Significant Bits (LSB) [1] and echo hiding [2], [3]. The schemes of transformed domain usually use DFT [4], DCT [5] or DWT [6]. Recently, a new transformed domain audio watermarking scheme was proposed in the cepstrum domain [7]. In [8], an audio watermarking scheme based on artificial neural network (ANN) was presented.

A scheme of digital audio watermarking using neural network is proposed in this paper. The original audio signal is divided into the synchronous modules and watermark modules. The synchronous code and the hiding watermark are embedded into the synchronous modules and the DCT coefficients of the watermark modules separately. To extract the watermark sequence, after we detect the exact position of the watermark module, the neural network is trained using the DCT coefficients related to chaotic sequence until it achieves convergence. The watermark sequence is extracted using the trained neural network with the DCT coefficients relating to it.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 795–800, 2005.

2 Watermark Embedding

The block diagram of watermark embedding part is shown in Fig.1. The original audio signal is divided into synchronization modules and watermark modules, and each watermark module is between two synchronization modules. There are three main stages in this part, the hiding watermark generation, the synchronous code embedding and the hiding watermark embedding.

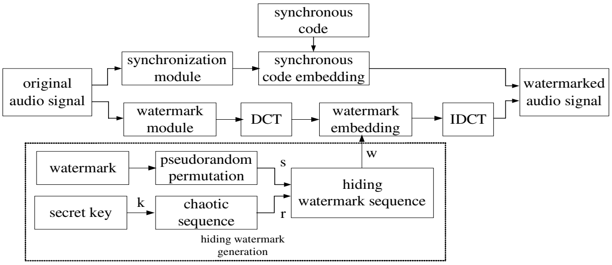


Fig. 1. Block diagram of watermark embedding part

2.1 Hiding Watermark Generation

Firstly, we convert the two-dimensional watermark image of size $\mathrm{M}_1 \times \mathrm{M}_2$ into one-dimensional sequence. To eliminate the correlation of the neighbor elements, we permute it with m sequence. The permuted watermark sequence can be described as

$$\mathbf{s} = \{s_1, s_2, \cdots, s_p\}, \quad p = \mathrm{M}_1 \times \mathrm{M}_2 \tag{1}$$

Secondly, since chaotic sequences is sensitive to the initial condition, we use it as part of the hiding watermark sequence

$$\mathbf{r} = \{r_1, r_2, \cdots, r_q\} . \tag{2}$$

Lastly, the two sequences $\mathbf{s}$ and $\mathbf{r}$ are combined together to form the hiding watermark sequence

$$\mathbf{w} = \left\{\mathbf{r}, \mathbf{s}\right\} = \left\{r_1, r_2, \cdots, r_q, s_1, s_2, \cdots, s_p\right\} = \left\{w_0, w_1, \cdots, w_{p+q-1}\right\} . \tag{3}$$

To improve the security of our scheme, we take the initial value and iterative coefficient of the chaotic sequence as part of the secret key in watermark extraction.

2.2 Synchronous Code Embedding

To improve the robustness, the embedding position of the first synchronization module is selected randomly. In addition, the information of embedding position is also regarded as the secret key in watermark extraction. The synchronous signal must have δ-like autocorrelation and the shorter code size. We select the Bark code of 12 bits as the synchronous code, which is embedded into the sampling values of original audio signal by quantization method in the time domain.

2.3 Hiding Watermark Embedding

There are three steps in this stage. First, the DCT transform is performed to the watermark module to obtain the DCT coefficients set

$$\mathbf{D} = \left\{d_j, j = 0,1,\cdots,N, N > p+q-1\right\}. \tag{4}$$

Second, the hiding watermark sequence $\mathbf{w} = \{w_0, w_1, \cdots, w_{p+q-1}\}$ are embedded into the DCT coefficients of low and middle frequency by following method

$$\begin{cases} e_j = [d_{j-2} + d_{j-1} + d_{j+1} + d_{j+2}]/4 \\ d'_j = \begin{cases} [d_j + e_j + | d_j - e_j |]/2 + \alpha(2w_j - 1), & w_j = 1 \\ [d_j + e_j - | d_j - e_j |]/2 + \alpha(2w_j - 1), & w_j = 0 \end{cases} \end{cases} \tag{5}$$

where α is a weighting factor.

Last, the inverse DCT transform is performed on $\mathbf{D}' = \{d'_j, j = 0,1,\cdots,N\}$, and the watermarked audio signal is obtained by the combination of the synchronization modules after embedding synchronous code and watermark modules which have been embedded with the hiding watermark.

3 Watermark Extraction

The block diagram of watermark extraction part is shown in Fig.2. There are two steps in this stage, the synchronous code extraction and the watermark extraction.

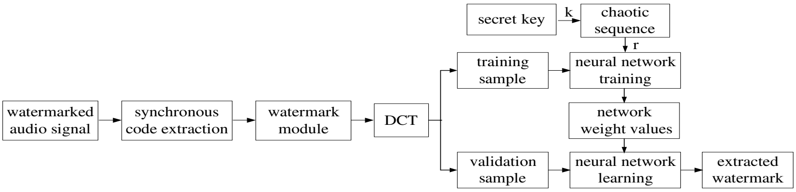


Fig. 2. Block diagram of watermark extraction part

3.1 Synchronous Code Extraction

We may obtain the position of the two nearest synchronous code according to the secret key. Then the synchronous code can be extracted using inverse quantization method. Through the correlation between the extracted synchronous code and the synchronous code, we could determine whether the watermarked audio signal was attacked. If the watermarked audio signal hasn't been attacked, then the position of the watermark module can be determined.

3.2 Watermark Extraction

As the watermarked audio signal may be attacked, we use different symbol to represent the watermarked audio signal in the watermark detecting part. Firstly, the DCT transform is applied to the watermark module to obtain the DCT coefficients set

$$\mathbf{D}'' = \left\{ d''_j, j = 0,1,\cdots,N \right\} . \tag{6}$$

Secondly, we extract watermark by using a nonlinear feed-forward neural network shown in Fig.3.

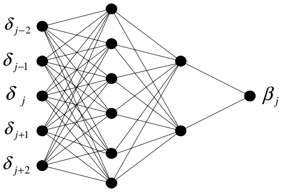


Fig. 3. The model of the neural network

Define the sample set which is composed of training sample and validation sample as

$$\mathbf{I}(j) = \left\{ \delta_{j-2}, \delta_{j-1}, \delta_j, \delta_{j+1}, \delta_{j+2}, \beta_j \right\}, j = 0,1,\cdots, p+q-1 \tag{7}$$

where

$$\begin{cases} \delta_{j\pm1,j\pm2} = (d''_j - d''_{j\pm1,j\pm2}) / \max\limits_{0\le j\le p+q-1} \{d''_j\} \\ \delta_j = (d''_j - e''_j) / \max\limits_{0\le j\le p+q-1} \{d''_j\} \\ e''_j = [d''_{j-2} + d''_{j-1} + d''_{j+1} + d''_{j+2}] / 4 \end{cases} . \tag{8}$$

The subset of training sample is $\mathbf{Q} \subset \mathbf{I}$, it can be described as

$$\mathbf{Q}(j) = \left\{ \delta_{j-2}, \delta_{j-1}, \delta_j, \delta_{j+1}, \delta_{j+2}, \beta_j \right\}, j = 0,1,\cdots, q-1 \tag{9}$$

where

$$\beta_j = \begin{cases} (d''_j - \delta_j) / \max\limits_{0\le j\le p+q-1} \{d''_j\} & w_j = 1 \\ -(d''_j - \delta_j) / \max\limits_{0\le j\le p+q-1} \{d''_j\} & w_j = 0 \end{cases} \quad j = 0,1,\cdots, q-1 \tag{10}$$

The neural network is trained using the training sample shown in (9) until it achieves convergence.

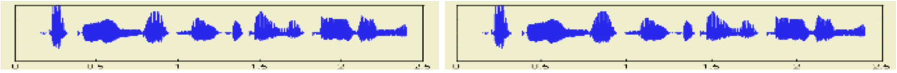
The remainder sample set $\mathbf{I}-\mathbf{Q}$ is used as the validation sample because it contains the watermark information which is wanted to be extracted. Thus we regard the sample set $\mathbf{I}-\mathbf{Q}$ as the neural network inputs, and then extract the watermark information according to the result of the network output $\hat{\beta}_j$ as follows

$$\hat{w}_j = \begin{cases} 1, & \hat{\beta}_j \geq 0 \\ 0, & \hat{\beta}_j < 0 \end{cases} \quad j = q, q+1, \cdots, q+p-1 \tag{11}$$

Lastly, do the reverse pseudorandom permutation on the obtained watermark sequence $\{\hat{w}_j\}$, and then convert the one-dimensional watermark sequence into the two-dimensional watermark image. The chaotic sequence used here is the same as the watermark embedding part.

4 Experimental Results

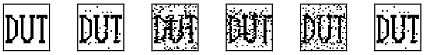
In our experiments, the sample frequency of the original audio signal was 8kHz. The waveform of the original audio signal and watermarked audio signal are shown in Fig. 4(a) and 4(b) respectively. The watermark is a binary image of size 32×32, showing in Fig.5(a). From Fig.4 we can see that there doesn't have any difference between them. And the same is in the hearing test.



(a)Original audio signal (b) Watermarked audio signal

Fig. 4. Waveforms of the audio signal

Fig.5(b) shows the extracted watermark image without signal processing operations. When the neural network achieves convergence, the correct rate of training sample set and the validation sample set are 99.45% and 96.78% respectively.



(a) (b) (c) (d) (e) (f)

Fig. 5. Various watermark images. (a) Original watermark. (b) Extracted watermark without signal processing operation. (c)-(f) Extracted watermark under various attacks

A variety of experiments are performed to demonstrate the robustness and effectiveness of the proposed method under different attacks. Experiment results under different attacks are shown in Fig.5(c)-(f). Fig.5(c) is the extracted watermark image after low pass filtering (the cut off frequency is 3 kHz, and the length of hamming is 1024). Fig.5(d) is the extracted watermark image under the operation of re-sampling. Fig.5(e) is the extracted watermark image for watermarked audio signal with white

Gaussian noise (the SNR is 20 dB). Fig.5(f) is the extracted watermark image after cropping. The results show that our scheme survives every attack.

5 Conclusions

A new watermarking scheme based on neural network is presented in this paper. Using the secret key, we can generate a training sample to train the neural network until it achieves convergence. The watermark can be extracted by the trained neural network without using original audio signal. Experiment results show that our approach can preserve inaudibility and is robust to noise interference, low pass filtering, temporal re-sampling as well as de-synchronization attack. Moreover, as using the synchronous code in the time domain, the watermarked audio signal can also be withstand cropping and losing package on the network transmission.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. 60172073, and No. 60372082.

References

1. Gerzon, M.A., Graven, P.G.: A High-Rate Buried-Data Channel for Audio CD. Journal of the Audio Engineering Society, **43** (1995) 3-22
2. Kim, H.J., Choi, Y.H.: A Novel Echo-Hiding Scheme with Backward and Forward Kernels. IEEE Trans. on Circuits and Systems for Video Technology, **13** (2003) 885-889
3. Ko, B.S., Nishimura, R., Suzuki, Y.: Log-scaling Watermark Detection in Digital Audio Watermarking. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04), **3** (2004) III-81-84
4. Malik, H., Khokhar, S., Rashid, A.: Robust Audio Watermarking Using Frequency Selective Spread Spectrum Theory. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04), **5** (2004) V-385-388
5. Yeo, I.K., Kim, H.J.: Modified Patchwork Algorithm: A Novel Audio Watermarking Scheme. IEEE Transactions on Speech and Audio Processing, **11** (2003) 381-386
6. Wu, S.Q., Huang, J.W., Huang, D.R., Shi, Y.Q.: Self-synchronized Audio Watermark in DWT Domain. Proceedings of the 2004 International Symposium on Circuits and Systems (ISCAS '04), **5** (2004) V-712-715
7. Lee, S.K., Ho, Y.S.: Digital Audio Watermarking in the Cepstrum Domain. IEEE Trans. On Consumer Electronics, **46** (2000) 744-750
8. Yang, H.J., Patra, J.C., Chan, C.W.: An Artificial Neural Network-Based on Scheme For Robust Watermarking of Audio Signals. Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '02), **1** (2002) I-1029-1032

Subsampling-Based Robust Watermarking Using Neural Network Detector

Wei Lu[1], Hongtao Lu[1], and FuLai Chung[2]

[1] Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai 200030, China
luweicn@sjtu.edu.cn, lu-ht@cs.sjtu.edu.cn

[2] Center for Multimedia Signal Processing and Department of Computing
Hong Kong Polytechnic University, Hong Kong, China
cskchung@comp.polyu.edu.hk

Abstract. This paper presents a robust digital image watermarking scheme by using neural network detector. First, the original image is divided into four subimages by using subsampling. Then, a random binary watermark sequence is embedded into DCT domain of these subimages. A fixed binary sequence is added to the head of the payload watermark as the samples to train the neural network detector. Because of the good adaptive and learning abilities, the neural network detector can nearly exactly extract the payload watermark. Experimental results show good performance of the proposed scheme on resisting common signal processing attacks.

1 Introduction

With the increasing demands for information security and copyright protection on Internet, traditional cryptology can not meet all of these applications for security services. On one hand, all of the encrypted information is noise-like random bit stream which can not be understood by anyone without the correct key. On the other hand, if the encrypted information is decrypted, it will lose all its security and can be pirated, distributed and modified without any protection. As a new technique, digital watermarking embeds some extra information, such as identification code, logo and other authenticated data, into digital products to protect their security [1]. While the quality of the watermarked data is retained, the watermark still can be detected under kinds of intentional and unintentional attacks, the watermarking technique provides an effective technique for copyright protection and authentication of multimedia digital works. Since 1990's, watermarking technique has attracted increasing interests [1, 2].

Some watermarking schemes based on neural networks have been developed in the past few years [3–5]. In [3] and [4], two different approaches using neural network detector are developed respectively to improve the robustness of the spatial domain watermarking scheme proposed by Kutter [6]. Fan *et al.* proposed a fragile watermarking scheme based on neural networks [5]. In this paper, we propose a novel digital image watermarking scheme based on the subsampling

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 801–806, 2005.

technique by using neural network detector for watermark extraction, which greatly improves the robustness of the watermarking scheme.

In [7], Chu developed a subsampling based digital image watermarking scheme. Suppose I represents the image to be watermarked with size $M \times N$, four subimages can be obtained by subsampling:

$$\begin{array}{ll} I_1(i,j) = I(2i-1, 2j-1) & I_2(i,j) = I(2i-1, 2j) \\ I_3(i,j) = I(2i, 2j-1) & I_4(i,j) = I(2i, 2j) \end{array} \tag{1}$$

where $i = 1, 2, \ldots, M/2$, $j = 1, 2, \ldots, N/2$. In [7], an assumption is made that the DCT coefficients for these subimages are approximately equal, i.e., $D_a \approx D_b$, where $a, b \in \{1, 2, 3, 4\}$, $a \neq b$, and D_a denotes the DCT coefficients of the subimage I_a. In most cases, the assumption is reasonable, because of the neighborhood correlation of image pixels.

2 Watermark Embedding

Suppose I is an image of size $M \times N$ to be protected by embedding a random binary watermark sequence W of length l. The actual watermark sequence to be embedded in the original image is composed of two parts:

$$S = P + W = p_1, p_2, \ldots, p_k, w_1, w_2, \ldots, w_l = s_1, s_2, \ldots, s_m \tag{2}$$

where $P = p_1, p_2, \ldots, p_k$ is a fixed pattern sequence with length k, $W = w_1, w_2, \ldots, w_l$ is the payload watermark, $m = k+l$. The pattern sequence P provides samples to train neural networks.

The proposed watermark embedding procedure is shown in Fig. 1, and can be described in detail as follows:

1. Subsampling the original image I to get four submiages, I_1, I_2, I_3, and I_4. Then, DCT is applied to these subimages to obtain four coefficient sets, D_1, D_2, D_3, and D_4, where the watermark S is embedded.
2. Generating randomly and secretly a coordinate set $Z = \{(x_t, y_t)\}$, where $1 \leq x_t \leq M$, $1 \leq y_t \leq N$, and $t = 1, 2, \ldots, m$. Using a secret key to generate a random sequence $C = \{c_i\}$ with length m, $1 \leq c_i \leq 4$, and $i = 1, 2, \ldots, m$.
3. Suppose s_i is the current watermark bit to be embedded, the embedding position is the i-th element of the set Z, i.e., (x_i, y_i), and the embedding scheme is
$$D_{c_i}(x_i, y_i) \leftarrow D_{c_i}(x_i, y_i)[1 + \alpha(2s_i - 1)] \tag{3}$$
4. IDCT is applied to the watermarked coefficient sets, D_1, D_2, D_3, and D_4 to get four watermarked subimages, then recompose these subimages to obtain the watermarked image.

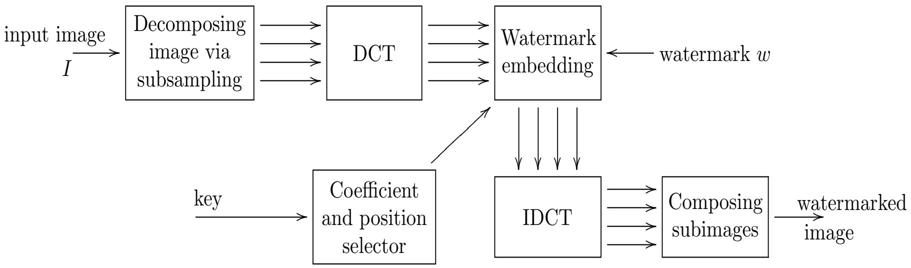


Fig. 1. Watermark embedding procedure

3 Watermark Detection

In order to use the neural network to detect whether a prescribed watermark exists or not in a given image I', we follow the following steps.

1. Using the same method to subsample the input image I', and get four submiages, I'_1, I'_2, I'_3, and I'_4. Then, obtain four coefficient sets, D'_1, D'_2, D'_3, and D'_4 by applying DCT to these subimages.
2. Generate the same random coordinate sets $Z = \{(x_t, y_t)\}$ and the random sequence $C = \{c_i\}$ using the same secret keys as those used in the watermark embedding procedure.
3. Before detecting the i-th watermark bit, we compute Eq. (4).

$$V_a = \sum_{e \in \Omega} \frac{D_e(x_i, y_i)}{3} \tag{4}$$

where $\Omega = \{1, 2, 3, 4\} - \{c_i\}$, and Eq. (5)

$$d_i = \frac{D_{c_i} - V_a}{V_a} \tag{5}$$

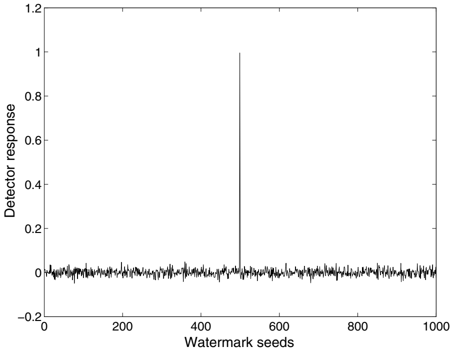
4. In order to detect the watermark, we use the traditional Back Propagation (BP) Neural Network, which is a multilayer perceptron. The training set is built from the first k input sets $\{d_i\}$ and pattern sequence P as $\{d_i, 2p_i - 1\}$, $i = 1, 2, \ldots, k$.
5. For the extraction of the payload watermark sequence, the last l input sets are fed to the trained neural network, each bit w^*_n is determined by the output w^o_{k+n} of the neural network detector as follow:

$$w^*_n = \begin{cases} 1 & \text{if } w^o_{k+n} \geq 0 \\ 0 & \text{else} \end{cases} \qquad n = 1, 2, \ldots, l \tag{6}$$

Thus we obtain the extracted watermark sequence $W^* = w^*_1, w^*_2, \ldots, w^*_l$.

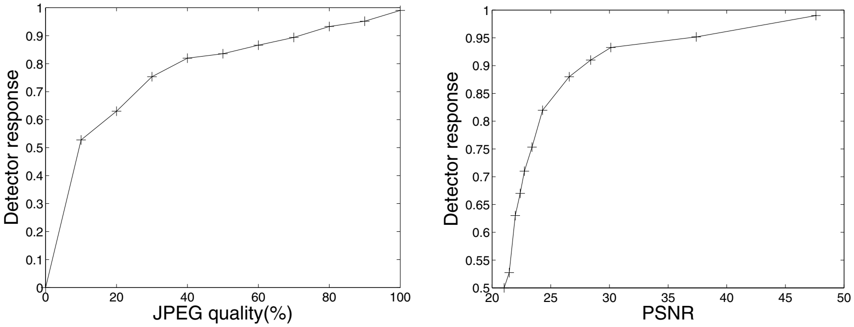
(a) (b)

Fig. 2. (a) The original image. (b) The watermarked image



(a)

Fig. 3. Detector response for 1000 watermark seeds



(a) (b)

Fig. 4. Detector response. (a) JPEG compression. (b) Gaussian lowpass filtering

Table 1. Robustness tests of the proposed scheme under signal processing attacks

	Lena		Baboon	
Attacks	*PSNR*	*NC*	*PSNR*	*NC*
JPEG 80	39.2501	0.9326	36.4701	0.9102
JPEG 60	36.0590	0.8660	34.9267	0.8350
JPEG 40	31.5791	0.8197	32.5016	0.7804
JPEG 20	31.5791	0.6301	29.3559	0.5602
Lowpass filter (7 × 7)	24.3491	0.8197	24.0126	0.7593
Average filter (5 × 5)	27.2730	0.8637	25.5792	0.7350
Median filter (5 × 5)	26.5301	0.8032	22.4021	0.8175
Gaussian noise (0.1)	31.6035	0.9276	32.4061	0.8857
Gaussian noise + Lowpass filter	25.0275	0.6720	26.2601	0.5659
Salt & Pepper noise (0.05)	25.0391	0.9307	25.1087	0.9250
Salt & Pepper noise + Median filter	33.0554	0.7430	29.5024	0.7147
JPEG 80 + Lowpass filter	31.5474	0.7241	30.3032	0.7032
JPEG 80 + Noise addition	32.2095	0.7803	32.3210	0.7193
JPEG 80 + Median filter	28.3771	0.6804	29.2091	0.6401
JPEG 80 + Average filter	31.0681	0.7120	22.3099	0.6402

In order to describe the similarity between the original watermark W and the extracted watermark W^* for making a binary decision on whether a given extracted watermark exists or not, we calculate the Normalized Correlation (NC) value [1] defined by

$$NC = \frac{\sum\limits_{i=1}^{l} W_i \cdot W_i^*}{\sum\limits_{i=1}^{l} W_i^2} \tag{7}$$

4 Experimental Results

We have tested the robustness of the proposed scheme by using the image shown in Fig. 2(a). In our experiments, the watermark pattern sequence is $0101 \cdots 01$ with length $m = 100$, the length of the payload watermark sequence $l = 1000$, and the watermark embedding strength $\alpha = 0.1$. For the BP neural network detector, we use the $9-5-1$ architecture, and learning parameter $lr = 0.2$. The watermarked image is shown in Fig. 2(b). Fig. 3 shows the detector response for 1000 watermark seeds, where only one seed is the correct watermark, whose NC is 1, which shows the exact watermark extraction. Furthermore, the detector response of NC between the original watermark and the extracted watermark from Fig. 2(b) is also 1.

As is shown in Tab. 1, the proposed scheme performs well under common signal processing attacks, including JPEG compression, filtering and noise addition etc., when using the common Lena and Baboon as host images. The proposed scheme can resist almost all these attacks. Fig. 4 shows the watermark NC detector response under JPEG compression and Gaussian lowpass filtering, where the proposed watermarking scheme can detect the correct watermark even under JPEG quality 10, and when $PSNR = 25$ under lowpass filtering. All these show that the proposed watermarking scheme has strong robustness against common signal processing attacks.

There are three controlling parameters in the proposed scheme. One is the length k of the pattern sequence, which influences the learning effect of the neural network detector. The second is the length l of the payload watermark sequence, which influences watermark detector's response. The third is the watermark embedding strength α, which influences the robustness and can not be too small or too large, otherwise, the 'mesh' phenomenon will degrade the visual quality of the marked image [7].

5 Conclusions

In this paper, we propose a subsampling-based robust digital image watermarking scheme using neural network detector. The experimental results show that it is more robust to common image processing. Since the watermark embedding is controlled by a secret key, our scheme is also secure.

Acknowledgments

This work is supported by Shuguang Program of Shanghai Education Development Foundation under grant 03SG11.

References

1. Petitcolas, F.A.P., Anderson, R.J., Kuhn, M.G.: Information Hiding - A Survey. Proc. IEEE, **87** (1999) 1062–1078
2. Hartung, F., Kutter, M.: Multimedia Watermarking Techniques. Proc. IEEE, **87** (1999) 1079–1107
3. Yu, P.T., Tsai, H.H., Lin, J.S.: Digital Watermarking based on Neural Networks for Color Images. Signal Processing, **81** (2001) 663–671
4. Lu, W., Lu, H., Shen, R.: Color Image Watermarking Based on Neural Networks. In: Advances in Neural Networks - ISNN 2004,. Volume 3174 of LNCS. (2004) 651–656
5. Fan, Y.C., Mao, W.L., Tsao, H.W.: An Artificial Neural Network-based Scheme for Fragile Watermarking. In: Proc. IEEE Int. Conf. Consumer Electronics,. Volume 8. (2003) 210–211
6. Kutter, M., Jordan, F., Bossen, F.: Digital Watermarking of Color Image using Amplitude Modulation. J. Electronic Imaging, **7** (1998) 326–332
7. Chu, W.C.: DCT-based Image Watermarking using Subsampling. IEEE Trans. Multimedia, **5** (2003) 34–38

Image Feature Extraction Based on an Extended Non-negative Sparse Coding Neural Network Model

Li Shang[1,2], Deshuang Huang[1], Chunhou Zheng[1], and Zhanli Sun[1]

[1] Hefei Institute of Intelligent Machines, Chinese Academy of Sciences
P. O. Box 1130, Anhui 230031, Hefei, China
{shangli,dshuang,zhengch,zhl_sun}@iim.ac.cn

[2] Department of Automation, University of Science and Technology of China
Anhui 230026, Hefei, China
shangli@ustc.edu.cn

Abstract. This paper proposes an extended non-negative sparse coding (NNSC) neural network model for natural image feature extraction. The advantage for our model is to be able to ensure to converge to the basis vectors, which can respond well to the edge of the original images. Using the criteria of objective fidelity and the negative entropy, the validity of image feature extraction is testified. Furthermore, compared with independent component analysis (ICA) technique, the experimental results show that the quality of reconstructed images obtained by our method outperforms the ICA method.

1 Introduction

Olshausen and Field forcefully argued that the receptive field is emerged by sparse coding (SC) [1] and they applied successfully a sparseness-maximization network to input data to testify their theory. Thus, sparse coding technique can be exploited to perform image feature extraction [2]. Subsequently, Bell and Sejnowski [3] applied independent component analysis (ICA) to image data analysis. This technique leads to a local representation quite similar to that obtained through the SC. However, the fact that features can be ‘cancelled each other out’ using subtraction operator is contrary to the intuitive notion of combining parts to form a whole [4]. Thus, Lee and Seung [4] forcefully argued for non-negative representations. Other arguments for non-negative representations come from biological modeling [5]. Later, P. O. Hoyer [5] combined sparse coding and non-negative matrix factorization (NMF) techniques into non-negative sparse coding (NNSC), and the receptive fields can be modeled in this way. However, Hoyer's model didn't consider the prior of the receptive fields and the self-adaptive sparseness measure function. Just so, this paper proposes an extended NNSC neural network model to extract features from natural images successfully. Moreover, our approach has a little quicker convergent speed than Hoyer's. Otherwise, by comparing the quality of restored images, it was found, in the simulation experiments, that our NNSC algorithm outperforms the standard ICA algorithm.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 807–812, 2005.

2 The Extended NNSC Neural Network Model

Any natural image can be modeled as a linear superposition of some features a_i:

$$x = \sum_{i}^{m} a_i s_i + n \cdot \tag{1}$$

where x denotes the natural image data, a_i are called basis vectors, s_i are mutually independent sparse variables, and n is Gaussian noise. The image model of NNSC is the same as that shown in Eqn. (1). The significant point is here that the input matrix X, basis vectors A and latent sparse coefficients S are non-negative in NNSC model. The fact that each unit s_i is either positively or negatively active means that every feature contributes to representing the stimuli of opposing polarity. This poses a contrast to the behavior of simple-cells in V1. Furthermore, V1 receives the visual data from the lateral geniculate nucleus (LGN) in the form of separated ON-channel and OFF-channel, and each channel's input data are positive.

2.1 Modeling the Objective Function

On the basis of the Hoyer's NNSC model (see [5], [6]), we propose an extended NNSC neural network model. Here, we also use the minimum reconstruction error and the sparseness like Hoyer, but the prior distribution of the receptive field and the self-adaptive sparseness measure is also considered. As such, the objective function can be constructed as follows:

$$J(A,S) = \frac{1}{2}\sum_{x,y}\left[X(x,y) - \sum_{i} a_i(x,y)s_i\right]^2 + \lambda\sum_{i} f\left(\frac{s_i}{\sigma_i}\right) + \gamma\sum_{i} a_i^T a_i \cdot \tag{2}$$

subject to the constraints: $X(x,y) \geq 0)$, $\lambda > 0$, $\gamma > 0$, $\forall_i$: $a_i \geq 0$, $s_i \geq 0$, and $\|a_i\| = 1$. Where $\sigma_i{}^2 = \langle s_i^2 \rangle$, $X(x,y)$ denotes an image, a_i and s_i denotes respectively the *ith* column of A and the *ith* row of S, λ is the tradeoff between sparseness and accurate reconstruction, and γ has to do with the variance of the prior distribution of a_i. And the self-adaptive attenuation function $f(\cdot)$ is chosen as:

$$f(u) = -e^{\left(-\frac{1}{2}au^2 - bu\right)} \quad (u \geq 0) \ . \tag{3}$$

where a and b are estimated according to the inner product theorem of Pham:

$$\left.\begin{aligned} a &= \frac{1 - E\{u\}b}{E\{u^2\}} \\ b &= \frac{2f(0)E\{u^2\} - E\{u\}}{E\{u^2\} - E\{u\}^2} \end{aligned}\right\} \cdot \tag{4}$$

where $f(0)$ is the value of the function $f(\cdot)$ at zero. and it can be estimated using a single kernel at 0 and it can be estimated using a single kernel at 0.

2.2 Neural Network Model of NNSC

According to Eqn. (2), we can obtain the derivatives of $\dot{a}_i$ and $\dot{s}_i$, shown as follows:

$$\dot{s}_i = a_i^T(x,y)\left[X(x,y)-\sum_{i=1}^{n} a_i(x,y)s_i\right]-\frac{\lambda}{\sigma_i}f'\left(\frac{s_i}{\sigma_i}\right)=a_i^T e-\frac{\lambda}{\sigma_i}f'\left(\frac{s_i}{\sigma_i}\right) \cdot \tag{5}$$

$$\dot{a}_i=\left[X(x,y)-\sum_{i=1}^{n} a_i(x,y)s_i\right]s_i^T-\gamma a_i(x,y)=e\, s_i^T-\gamma a_i \cdot \tag{6}$$

where $\sigma_i=\sqrt{\langle s_i^2\rangle}$, $e=X-AS$ is the residual error between the original image and the reconstructed image of this model. There is a simple network interpretation of Eqn. (5). Each output unit s_i is determined from a combination of a feedforward input term $\sum_{x,y} a_i^T(x,y)X(x,y)$, a recurrent term $\sum_{x,y}\left\{a_i^T(x,y)\left(\sum_j a_j(x,y)s_j\right)\right\}$, and a nonlinear self-inhibition term $f'(\cdot)$. The output values s_i are then fed back through the basis function a_i to form a reconstruction image. This recurrent neural network implementation of this differential Eqn. (5) is shown in Fig. 1.

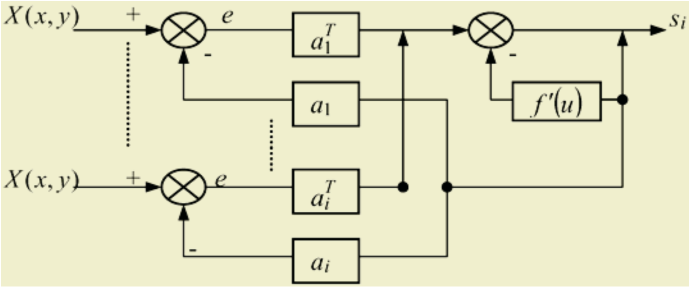


Fig. 1. The network implementation of our NNSC model

3 Algorithm Implementation

Now we discuss how to implement the algorithm for the extended NNSC network model. Firstly, we must get the non-negative input data matrix. Selecting randomly 10 noise-free nature images with 512×512 pixels, which are available on the Internet http:// www.cns.nyu.edu/lcv/denoise, we construct the sample set with 262144×10. Then, we sampled randomly subwindows of 8×8 pixels 10000 times from each original image, and converted every patch into one column. Thus, the input data set with the size of 64×100000 is acquired, here denoted by matrix X . Considering the non-negativity of the input data, we separate X into *ON*-channel and *OFF*-channel, denoted respectively by matrixes Y and Z . So, the nonnegative input data matrix $I=(Y;Z)$ with the size of 2×64×100000 is obtained. Consequently, each image patch is represented by a 2×8×8=128 dimensional vector. Secondly, Using the updating rules of A and S in turn, we minimized the objective function given in Eqn. (2) under the constraints of 64 sources, $\lambda = 0.1\sigma_i$, $\gamma = 0.02$, and with each channel scaled to have unit average squared activation. In experiment, we exploited conjugate gradient algorithm to update basis vectors A and used Eqn. (5) to update S. If the reconstruction error $e = \left\| I(x,y) - \sum_i a_i s_i \right\| < \varepsilon = 0.01$, end the iteration. Then, using the learned A and S, we can deal with the work of image reconstruction.

4 Experiments

4.1 Image Feature Extraction

Using the above algorithm and the non-negative database I mentioned in Section 3, the obtained results for 64 basis functions extracted from natural scenes are the first three panels shown in Fig. 2. It is easy to see that the learned basis vectors behave the legible locality and orientation in the frequency and spatial domain. For comparison, we also used the ICA method to extract image features from the same database, and the experimental result was the rightmost panel of Fig. 2. Apparently, the ICA basis can find the features that resemble local edges and bars, but compared with that shown in the third panel of Fig. 2, the latter behave much clearer line edge and orientation. Therefore, the bases obtained by our NNSC algorithm are better than ICA bases.

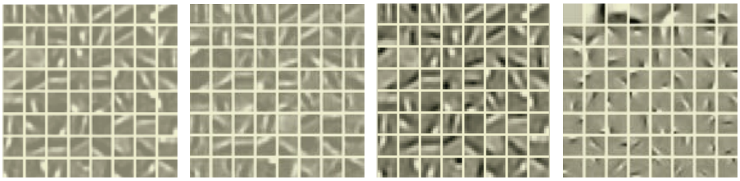


Fig. 2. Feature basis vectors estimated for natural scene data. The first three panels are orderly ON-channel, OFF-channel and ON-channel minus OFF-channel obtained by our NNSC algorithm. The last basis image is ICA basis

4.2 Image Reconstruction

In the work of image reconstruction, three test images, commonly known as Lenna, Einstein, and Boat, were selected from those ten images used for estimating the basis vectors in Fig. 2. Assume that each image is randomly sampled 50000 times with 8×8 pixels to get the non-negative training data set, denoted by matrix $\hat{X}$. Note that to find the accurate position of any image patch, we must remember the positions of each image patch appeared. Because of sampling randomly, the same pixel might be founded in different image patches. Therefore, for the sample pixel, we averaged the sum of the values of all reconstructed pixels, and used the averaged pixel value as the approximation of the original pixel. The reconstructed results were shown in Fig. 3. Clearly, the reconstructed results are satisfying. It is difficult to tell reconstructed images from the original images only with naked eyes. Furthermore, this shows that our proposed NNSC algorithm can be successfully applied on image feature extraction.

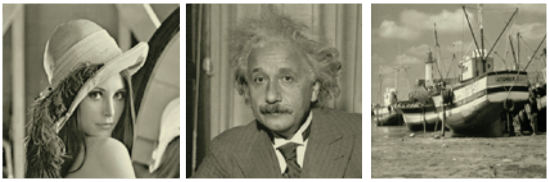


Fig. 3. The reconstructed images (Lenna, Einstein and Boat) by our NNSC network algorithm

Table 1. The negative entropy of reconstructed images and original images

Images	H of the original images	H of the reconstructed images
Lenna	7.4451	7.4417
Einstein	6.9408	6.9388
Boat	7.1914	7.1907

Table 2. Values of SNR_{rms} of reconstructed images gained by different methods

Reconstructed images	SNR_{rms} (Our NNSC)	SNR_{rms} (Standard ICA)
Lenna	38.5368	27.2750
Einstein	41.6952	30.2472
Boat	37.9523	27.2154

In addition, the effectiveness of reconstruction can be further confirmed by the measure of the negative entropy H . And H can be usually defined as:

$$H = -\sum_{i}^{255} p_i \log\left(p_i\right) \cdot \qquad (7)$$

where p_i is the probability of the appearing gray pixels in an image. The calculated values of H of different images are listed in Table 1. It is clear to see that the H value of any reconstructed image is significantly approximate to that of its original image, and the loss of image information is highly small. Thus, our proposed NNSC algorithm can reserve the original image information as possible as. On the other hand, we also utilized the criterion of the objective fidelity to compare the reconstruction results implemented by methods of ICA and our NNSC. In general, the objective fidelity used usually in practice is the mean square signal to noise ratio of the output image, denoted by SNR_{rms}. Let $I(x,y)$ denote the input image data set, and $\tilde{I}(x,y)$ denote the reconstruction image data, then the SNR_{rms} is defined as:

$$SNR_{rms} = \frac{\sum_{x=1}^{M}\sum_{y=1}^{N}\tilde{I}(x,y)^2}{\sum_{x=1}^{M}\sum_{y=1}^{N}\left[\tilde{I}(x,y)-I(x,y)\right]^2} . \tag{8}$$

where M and N denote the size of the image data. The calculated values of SNR_{rms} are shown in Table 2. Clearly, the SNR_{rms} value of each reconstructed image obtained by our method is greater than that obtained by ICA method. Therefore, the quality of reconstructed images of the former outperforms that of the latter.

5 Conclusions

In this paper, we proposed an extended NNSC neural network model for extracting natural image features. The basis vectors obtained by our method much resemble the receptive fields of neurons in primary visual cortex, which behaves clearer lines and edges by comparing with ICA bases. As a practical application of this method, we conducted the experiments of image reconstruction. The simulation results showed that, using our NNSC algorithm, the sparseness of the hidden components is greatly enhanced, and the effectiveness of reconstructed image is also much better than the ICA method. However, the test images used are all gray-scale images. Then, how about colour and stereo images? In the future works, the research task will focus on how to extract features of colour and stereo images using our NNSC algorithm.

References

1. Olshausen, B. A., Field, D. J.: Emergence of Simple-cell Receptive Field Properties by Learning A Sparse Code for Natural Images. Nature, **381** (1996) 607-609
2. Hyvärinen, A., Oja, E., Hoyer, P., Horri, J.: Image Feature Extraction by Sparse Coding and Independent Component Analysis. In Proc. Int. Conf. on Pattern Recognition (ICPR' 98), Brisbane, Australia (1998) 1268-1273
3. Bell, A., Sejnowski, T. J.: The 'Independent Components' of Natural Scenes Are Edge Filters. Vision Research, **37** (1997) 3327-3338
4. Lee, D. D., Seng, H. S.: Learning The Parts of Objects by Non-negative Matrix Factorization. Nature. **401** (1999) 788-891
5. Hoyer, P. O.: Non-negative Sparse Coding. In Neural Networks for Signal Processing XII, Martigny, Switzerland (2002) 557-565

Evolving Optimal Feature Set by Interactive Reinforcement Learning for Image Retrieval

Jianbo Su[1,2], Fang Liu[1], and Zhiwei Luo[2]

[1] Department of Automation, Shanghai Jiao Tong University
Shanghai 200030, China
{jbsu,fangliu}@sjtu.edu.cn

[2] Bio-Mimetic Control Research Center, RIKEN
Anagahora, Shimoshidami, Moriyama-ku, Nagoya 463-0003, Japan
{su,luo}@bmc.riken.jp

Abstract. This paper proposes a new image retrieval strategy based on the optimal feature subset that is iteratively learned from the query image. The optimal feature set that can well describe the essential properties of the query image with respect to a retrieved image database is obtained from reinforcement learning procedure with the help of human-computer interaction. Through human-computer interaction, user can provide similarity evaluation between the query and retrieved images, which actually gives the relevance feedback for a contend-based image retrieval method, and further serves as environmental rewards to feature set evolution actions in reinforcement learning procedure. Experiment results show the effectiveness of the proposed method.

1 Introduction

To find images both semantically and visually relevant to a query image from a massive image database has been an important topic in image processing and knowledge engineering communities. Owing to drastically increased scales of image databases, content-based [1] image retrieval method, which index images by their own visual contents, such as color, texture, etc, has gradually become a dominant method instead of text-based retrieval techniques [2] since it basically releases the vast amount of tedious labor to manually annotate images with keywords. Due to increasingly richer contents in the images, manual annotation of an image is becoming more difficult to carry on, and it is also more subjectively individual related.

It is critical to obtain proper image features specifying retrieval interest for the performance of a retrieval system. The relevance feedback technique is widely accepted in content-based retrieval in that it enables the system to adaptively adjust the relevance between queried and retrieved images [3]. Hence, research in relevance feedback is active in content-based image retrieval topic. The relevance feedback so far includes query shifting, relevance estimation, Bayesian inference [4], AdaBoost [5] and SVM-based methods [6], etc. These methods mainly emphasize on enhancing adaptability of the retrieval methods to different applications.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 813–818, 2005.

An image can be described by a variety of features, such as color, texture and shape, each of which may have several components, such as color layout, color histogram, texture co-occurrence matrix, FFT descriptor and Hough transformation. These features construct the feature space for the image, no matter whether they are explicitly expressed and extracted or not. It is presumable that there exists an optimal feature set that is both compact and sufficient to define the query image with respect to the retrieved image database [7]. Therefore to infer the optimal feature set for the query image from its complete feature space is radical to an efficient image retrieval system [8]. Since the optimal feature set of a query image may be changeable according to the image database retrieved, an online feature learning and extraction procedure should be involved. Since it is also closely related to the user's subjectivity, it is very natural to introduce human-computer interaction (HCI) to incorporate user's knowledge into feature learning procedure. To construct a "Human-in-the-Loop" image retrieval system is also a new direction worthy to be investigated [3].

This paper studies online learning method for evolving the optimal feature set of the query image by reinforcement learning method [9]. Since the standard reinforcement learning method suffers from average update complexity depending on the size of the state-action space, HCI is involved to provide direct evaluation for the learning performance to speed up learning rate [10]. Similarity of the queried and retrieved images is evaluated through HCI and serves as the relevance feedback, which further provides rewards for feature evolution actions in reinforcement learning procedure. Section 2 describes the feature space and peers into the optimal feature set. Section 3 studies image retrieval based on feature learning algorithm. Experimental results are shown in Section 4, followed by conclusion in Section 5.

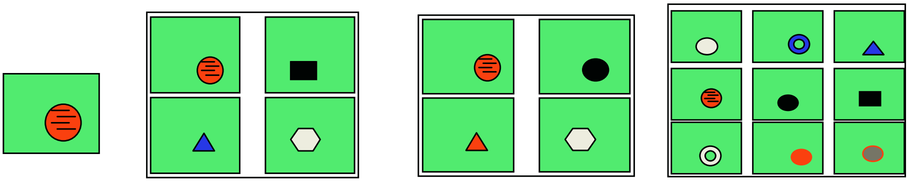
2 Feature Space and Optimal Feature Set

The feature space of an image may have infinite dimensions and is independent of image context. Nevertheless, it is only practical to describe the properties of an image with finite number of features for image retrieval. This finite feature space should be large enough to include the optimal feature set. This implies that construction of the finite feature space depends on image context. Dimensions and elements of the feature space for a query image and an image database are difficult to make out due to unknown properties and relations between them. If we do not have any prior knowledge on the retrieval task, we can only employ some general features that are not domain-specific, such as color, texture and shape, etc.

Definitions and computations of these features can be found in [1] and references therein. The feature space that is formed from these features can be expressed as:

$$\mathbf{f} = (f_{color}, f_{texture}, f_{shape}, \cdots), \tag{1}$$

where $\mathbf{f}$ denotes the feature space composed of color related features f_{color}, texture related features $f_{texture}$, and shape related features f_{shape}.



(a) A query image (b) The first image database (c) The second image database (d) The third image database

Fig. 1. Optimal feature sets with respect to query image and image database

As mentioned before, the optimal feature set is related to content of query image and the image database to be retrieved. Fig. 1 gives a simple explanation on this. Fig. 1(a) is the query image that will be retrieved in three image databases shown in Fig. 1(b)-(d). It is easy to see that for the first image database in Fig.1 (b), either shape, or texture, or color, can be the optimal index feature for image retrieval. In this case, it is important to choose one that is more computationally cheap. Due to a vast number of images included in database, even a little less computation cost for single feature extraction might lead to a dramatic improvement for retrieval speed. For the second image database, we can have two available sets of features. One is color and shape features, another is texture feature. The color and shape feature set has higher dimension that that of texture feature set. Nonetheless, texture feature might be more expensive to be obtained than color and shape features. For the third image database, it is apparent that texture feature is the optimal one. However, it is more important to see that if a redundant feature, shape, is included, performance of the whole system will be deteriorated, even if the optimal feature is also utilized in retrieval. This means, when we use a feature set to recognize query image and get unsatisfactory result, it does not mean that all of the component features in the feature set are not effective. In addition, this example also shows that even if the texture feature is the optimal one and the color feature is the second optimal one, yet color feature is fast to be obtained to save total retrieval time. Hence, color feature might be adopted in applications with much faster retrieval speed but a little worse retrieval performance.

3 Image Retrieval Based on Feature Learning

The feature selection is a sequential decision-making procedure that interacts with uncertain environment by actions and feedback. We define *state* as the feature set vector F_t that is defined in feature space and used as index to perform image retrieval. The *action* a is defined as adding, deleting, or changing a feature. Reinforcement learning is to select optimal action for specific task through online learning. Practically, the Q-learning algorithm is adopted. For a specific state of the system, F_t, an action a_t is triggered, which will lead the system to a new state F_{t+1}, and obtain a reward r_t from environment, then the evaluation function $Q(F, a)$ can be iterated after it is initialized:

$$Q_{t+1}(F_{t+1}, a_{t+1}) = Q_t(F_t, a_t) + \alpha \left[r_t + \gamma \max_{a_{t+1}} Q_t(F_{t+1}, a_{t+1}) - Q_t(F_t, a_t) \right], \quad (2)$$

where α is the learning rate, and we have $0 < \alpha < 1$. γ is a discount rate, and $0 < \gamma < 1$. It has been proved that $Q(F, a)$ can converge to an optimal value under a Markov process with infinite search, and an optimal strategy $\pi^*(F)$ can be reached:

$$\pi^*(F) = \arg\max_a Q(F, a). \quad (3)$$

To incorporate specialty knowledge of human operator into the learning process and speed up the convergence of it, we propose to give reward of the action by human-computer interaction (HCI). The reward is defined according to different evaluation criterions. Suppose there are n criterions to be evaluated:

$$r_t = k_1 r_{1t} + k_2 r_{2t} + \cdots + k_n r_{nt}, \quad (4)$$

where $r_i, (i = 1, \ldots, n)$ is the reward provided by user according to i-th criterion, and k_i is the weight of r_i , which is normally determined from prior knowledge.

Reinforcement learning with HCI can obviate dependence of its learning performance on the initial state. Fig. 2 shows the scheme of the proposed method. Evolvement behavior of a temporary feature subset includes feature addition, subtraction and variation. The query image and the images in massive image database are compared according to a present available feature set. Comparison results are evaluated by user to provide relevance feedback to compute query accuracy as the reward to the reinforcement learning procedure. The feature subset is iteratively evolved according to the rewards to its evolvement behavior until a satisfactory feature set is reached. The optimal evolvement is from maximizing the accumulated rewards for the evolvement behavior. It is worth mentioning that the feature set resulted from the learning procedure is only a subset of the complete feature space, not one of all features with different weights.

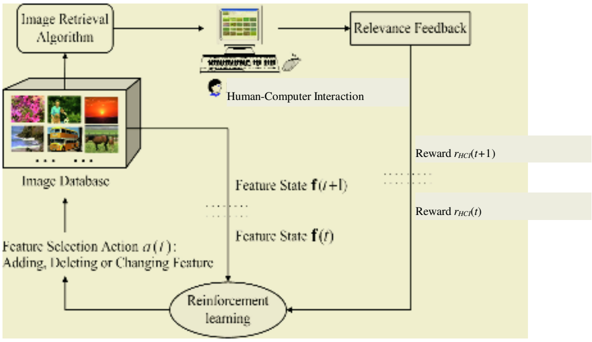


Image Retrieval Algorithm
Relevance Feedback
Human-Computer Interaction
Image Database
Reward $r_{HCI}(t+1)$
Feature State $\mathbf{f}(t+1)$
Reward $r_{HCI}(t)$
Feature State $\mathbf{f}(t)$
Feature Selection Action $a(t)$:
Adding, Deleting or Changing Feature
Reinforcement learning

Fig. 2. Feature learning of HCI-based reinforcement learning and relevance feedback

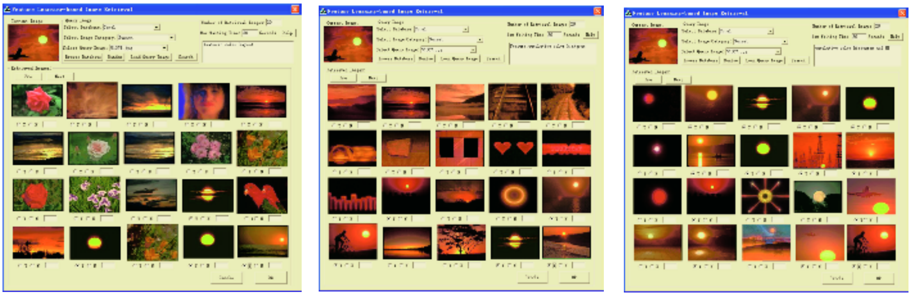


Fig. 3. Retrieval results by color layout feature; color histogram feature; and cumulative color histogram and Hough transform features (From left to right)

4 Experiments

The algorithm proposed is evaluated on the well-known Corel image database. We are trying to obtain from the image database as many images as possible similar to the query image. We adopt seven features in the experiment, which come from three categories. The color features are cumulative color histogram and color layout. The texture features are co-occurrence matrix and boundary 3-D feature. The shape features involved are Hough transformation, boundary histogram and Fourier descriptor. All of these features along with any of their combinations construct the feature space, which altogether has 35 dimensions. In the experiments, the reward from the image retrieval result is simply determined by

$$r_{RF} = \begin{cases} 1 & if\ retrieval\ results\ are\ satisfactory \\ \frac{result_{positive}}{result_{total}} & otherwise \end{cases} \tag{5}$$

where $result_{positive}$ is the number of images that satisfies the user and $result_{total}$ is the total number retrieved.

Different image features are tested to show that the performance of the retrieval system is gradually improved with the learning procedure. Fig. 3 shows the query results for the query image shown in upper-left corner by using different feature sets, respectively. These features are iteratively selected from learning procedure based on HCI.

Fig. 4 shows the performance of the learning procedure for locating specific features with respect to leaning times. It is seen that iteration number for an initial feature set to converge to the optimal feature set decreases in terms of the trial times. It is worthy noting that a consistent feedback from user is important for a successful feature learning procedure. In addition, the learned feature set can be used as reference to select initial feature set for similar query images.

5 Conclusions

A new content-based image retrieval method is proposed by adopting an online feature learning strategy from reinforcement learning algorithm. The human-computer interaction is involved into the reinforcement learning framework to

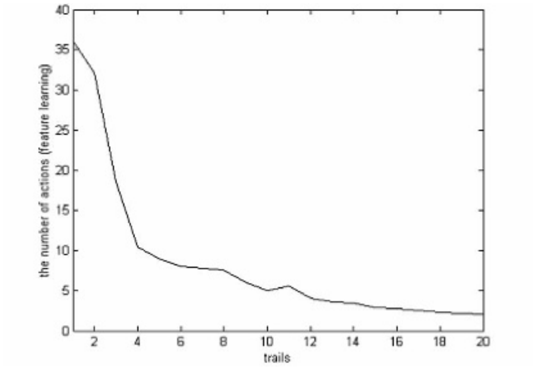


Fig. 4. Times to reach optimal feature set versus number of test

embed user's knowledge in order to improve the feature learning efficiency. Moreover, it can overcome the problem that reinforcement learning is sensitive to initial values of the parameters involved. Relevance feedback in content-based image retrieval method is thus provided directly by user and then serves as environmental reward for the learning procedure. The method is evaluated to be effective in experiments on real data. Future work lies in how to construct a sufficiently complete feature space from learning and investigate further applications.

References

1. Rui, Y, Huang, T. S., Chang, S.F.: Image Retrieval: Current Techniques, Promising Directions, and Open Issues. Journal of Visual Communication and Image Representation, **10** (1999) 39–62
2. Chang, S.K., Hsu, A.: Image Information Systems: Where Do We Go From Here?. IEEE Trans. on Knowledge and Data Engineering, **4** (1992) 431–442
3. Rui, Y., Huang, T. S., Ortega, M., Mehrotra, S.: Relevance Feedback: a Power Tool for Interactive Content-Based Image Retrieval. IEEE Transactions on Circuits and Systems for Video Technology, **8** (1998) 644–655
4. Cox, I. J., et al: The Bayesian Image Retrieval System, PicHunter: Theory, Implementation, and Psychophysical Experiments, IEEE Transactions on Image Processing. **9** (2000) 20–37
5. Tieu, K., Viola, P.: Boosting Image Retrieval. In: Proc. of IEEE Conf. on Computer Vision and Pat. Rec., (2000) 228–235
6. Tong, S. and Chang, E.: Supporting Vector Machine Active Learning for Image Retrieval. In: Proc. of ACM Int. Conf. on Multimedia, (2001) 107–118
7. Toussaint, G. T.: Note on the Optimal Selection of Independent Binary Features for Pattern Recognition. IEEE Trans. on Information Theory, **17** (1971) 618
8. Vafaie, H., Jong, K. D.: Robust Feature Selection Algorithms, In: Proc. of IEEE Int. Conf. on Tools with Artificial Intelligence, 356-363 (1993)
9. Sutton, R.S., Barto A.G.: Reinforcement Learning. MIT Press (1998) 51–81
10. Liu, F., Su, J.B.: An Online Feature Learning Algorithm Using HCI-Based Reinforcement Learning, Lecture Notes in Computer Science, **3173** (2004) 293–298

Perception-Oriented Prominent Region Detection in Video Sequences Using Fuzzy Inference Neural Network

Congyan Lang, De Xu, Xu Yang, Yiwei Jiang, and Wengang Cheng

Dept. of Computer Science, Beijing Jiaotong University
Beijing 100044, China
gltree@263.net

Abstract. In this paper, we propose a new approach for the prominent region detection from the viewpoint of the human perception intending to construct a good pattern for content representation of the video sequences. Firstly, we partition each frame into homogeneous regions using a technique based on a non-parameter clustering algorithm. Then, in order to automatically determine the prominent importance of the different homogenous regions in a frame, we extract a number of different mise-en-scene-based perceptual features, which influence human visual attention. Finally, a modified Fuzzy Inference Neural Network is used to detect prominent regions in video sequences, due to its simple structure and superior performance for automatic fuzzy rules extraction. The extracted prominent regions could be used as a good pattern to bridge semantic gap between low-level features and semantic understanding. Experimental results show the excellent performance of the approach.

1 Introduction

With the current advance of video database technique, efficient video retrieval and browsing have become crucially important, especially with the development of the video content description standard, such as MPEG-7. Unfortunately, our current approaches to video processing suffer from one or more shortcomings that stem from the semantic gap between low-level features and high-level semantic concepts.

To bridge the semantic gap, most previous works select semantic video objects[1] as the underlying video patterns for video content representation and feature extraction. However, the major problem using semantic video object as video patterns is that automatic semantic video object extraction in general still needs for human's interaction at the current stage.

Studies of visual attention and eye movements[2],[3] have show that humans generally only attend to a few areas in an image. Even when given unlimited viewing time, subjects will continue to focus on these few areas rather than scan the whole image. According that fact, many research efforts have been given in detecting salient region in image or video intend to overcome the limitations of semantic object extraction. Generally, most existing approaches [4],[5] aim at detecting the salient region for images, which are mainly based on the construction of a saliency map modeled as an integration of different measurable, low-level image features such as color, orientation, depth information etc. The purpose of the saliency map is to represent the

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 819–827, 2005.

conspicuity of each locations of the visual field, that is, salient regions extracted have higher prominent importance than the other regions. And a salient region detection and tracking method is presented in[13], which extract salient regions based on color and orientation maps followed by a tracking mechanism.

Salient region extraction based on saliency map provides a good starting point for semantic-sensitive content representation. However, perceived salient region extraction for image or video is still an unsolved problem. One reason is that low-level features are often not enough to classify some regions unambiguously without the incorporation of high-level and human perception-oriented information into the classification process. Another reason for the problems is perception subjectivity. Different people can differ in their perception of high-level concepts, thus a closely related problem is that the uncertainty or ambiguity of classification in some regions cannot be resolved completely based on measurements methods.

In this paper, we propose a new method for extraction perception prominent regions in video sequences intend to remove limitations explained above. For each frame, a pre-segmentation composed of homogeneous regions is produced, and then the segmented image is analyzed by a number of perceptual attributes which are used by the filmmakers to guide our attention across the screen, shaping our sense of the space that is represented and emphasizing certain parts of it [6]. It is known that fuzzy logic can provide a flexible and vague mapping from linguistic terms to represent the human perception, and neural networks have superior learning ability to extract fuzzy rules automatically. Hence, to enable alleviate the semantic gap and the perception subjectivity problems, our method for automatically determining the perceptual importance of different regions in an image is constructed based on fuzzy inference neural networks (FINNs).

The rest of the paper is organized as follows: Pre-segmentation process of image frames is described in the next section. In Sect. 3, the effective and efficient features extraction for primitive homogenous regions is implemented. And then, prominent region detection based on FINNs is presented in Sect.4. The effectiveness of the proposed approach is validated by experiments over real-word video clips are expressed in Sect.5. Concluding remarks are given in Sect. 6.

2 Pre-segmentation of Image Frames

In this section, each frame is initially segmented into homogeneous regions based on mean shift algorithm, the color and texture information is incorporated into the mean shift segmenter [7].To segment an image, we first partition it into 4*4 blocks and extracts a 6-D feature vector for each block. In particular, three of them are the average color components, which we compute in CIE LUV color space. The other three represent energy in high frequency bands of wavelet transforms[12]. And then, three wavelet features are computed as the square root of the 2nd-order moment of the Wavelet coefficients in the HL, LH, and HH frequency bands. After the 6_D feature vector is extracted for all 4*4 blocks, we apply the mean shift clustering approach [7] to segment the image into homogenous regions. Fig.1 shows the results of partitioned homogenous regions for two frames.

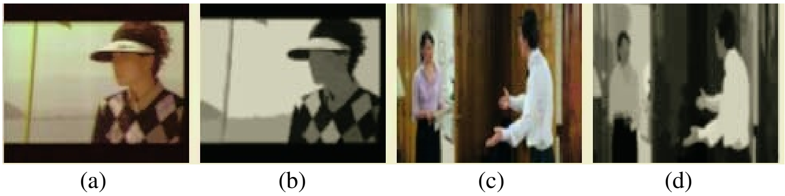


Fig. 1. Homogenous region segmentation examples: (a), (c) original images; (b), (d) their respective homogenous regions segmentation

3 Region Feature Description

Human selective attention makes us distinguish important features of input stimulus from others and focus on a critical feature of an input stimulus. Most basically, our visual system is attuned to perceiving change both in time and space, since one of the main goals of human visual system is to minimize uncertainty. This is also in agreement with Gestalt Organization Theories. By taking advantage of these facts, the filmmaker uses the arrangement of the *mise-en-scene* to attract our attention by means of changes in light, shape, movement and other aspects of the image. Thus, in order to get suitable features to represent region we extract a number of perceptual features described as the following.

1) Contrast of Region with Surroundings (CSr). Regions, which have a high contrast with their surroundings, are likely to be greater visual importance and attract more attention. The filmmaker can also exploit principles of color contrast to shape our sense of screen space. For instance, bright colors set against a more subdued background are likely to draw the eye [6]. The contrast importance $CSr(R_i)$ of a region R_i is calculated as:

$$CSr(R_i)=\sum_{m=1}^{n} I^*(R_i)-I^*(R_{i-neighbours_m}). \tag{1}$$

2) Orientation Conspicuity of Region (OCr). Gabor filtering allows getting information about local orientation in the image. Here, this technique is also employed to descript region orientional information importance.

Local Orientations O_θ are obtained by applying Gabor filters to the original images from particular orientations $\theta=\{0^\circ,45^\circ,90^\circ,135^\circ\}$. The four oriented images are added up to generate the desired orientation map, and then normalized orientation map $\overline{I}_{orientations}$ is achieved by using a traditional normalization operator.

$$OCr(R_i)=\frac{\sum_{p}\overline{I}_p}{N_{pixel}(R_i)}, p\in R_i. \tag{2}$$

where $N_{pixel}(R_i)$ denotes the number of pixels in the region R_i .

3) Shape Indicator of Region (SIr). The shape indicator of region can be calculated as:

$$SIr(R_i) = \frac{N_{edge}(R_i)}{N_{pixel}(R_i)}. \tag{3}$$

where $N_{edge}(R_i)$ is the number of pixels in the region R_i which border with other regions. Generally speaking, a small $SIr(R_i)$ signifies a long, thin region, which further implies a high prominent importance, while for rounder regions it will be lower.

4) Compositional Balance Indicator of Region (CIr). Compositional Balance is an importance factor of the *mise-en-scene* to guide our attention and shape our viewing of the image, it can be interpreted as the extent to which the areas of screen space have equally distributed masses and points of interest. Generally the filmmakers often center the frame on the character's body which is a simplest way to achieve compositional balance[6]. However, no existing works pay attention to this information. Based on the above observations, the compositional balance factor is determined according to the following measure:

$$CIr(R_i) = \begin{cases} \dfrac{CSr(R_i)}{\left\| gc(R_i) - \overline{gc(R)} \right\|}, \overline{gc(R)} \in R_i & . \\ \dfrac{CSr(R_i)}{\left\| CSr(R_i) - CSr(R'_i) \right\| + \left\| gc(R_i) - \overline{gc(R)} \right\|}, \overline{gc(R)} \notin R_i \end{cases} \tag{4}$$

where $gc(R_i)$ is the gravitational center of the region R_i and the mean of gravitational center for all regions within a frame is denoted as $\overline{gc(R)}$. And the region R'_i is selected whose gravitational center is the nearest neighbor of the symmetrical points of   with respect to the midline of the frame. The larger *CIr* means that the higher possibility that it will be a character portion of the frame.

5) Motion Prominent Indicator (MIr) of Region. Almost invariably, a moving item draws our attention more quickly than a static item does. Motion information is an important cue for human to perceive video content. In our work, block motion vectors are used for motion analysis.

Since of a motion vector reflects motion velocity of the block, we define the $IM\ (R_i)$ as the mean of the magnitude of the motion vectors in the region R_i . An approach similar to[9] is adopted to descript our motion consistency of region, denoted as $MC(R_i)$. Specially, the motion consistency is calculated based on the entropy, which should be consulted for the details in [9], estimation of the probability is obtained by summation over rows and columns of an eight-bin phase histogram of the region R_i. Then we define the motion prominent importance as following:

$$MIr(R_i) = IM \bullet MC \quad . \tag{5}$$

4 Prominent Region Detection Based on FINN

After region features are extracted, the perceptual prominence is required to assign to each region. Fuzzy logic has the capability of modeling perception vagueness, uncertainty and can support human-like reasoning. On the other hand, studies of fuzzy neural networks that combine both advantages of the fuzzy systems and the learning ability of the neural networks have been carried out. These techniques can alleviate the matter of fuzzy modeling by learning ability of neural networks [10],[11]. FINNs is ideal for our problem as it can extract and use rules automatically with a simple structure and superior performance.

4.1 Architecture of FINN

Fig.2 shows the basic structure of FINNs [10]. It consists of two layers. One is the input-output (I/O) layer and another is the rule-layer. The I/O layer consists of the input- and the output- part. Each node in the rule-layer represents one fuzzy rule. Weights from the input-part to the rule-layer and those from the rule-layer to the output-part are fully connected and they store fuzzy if-then rules.

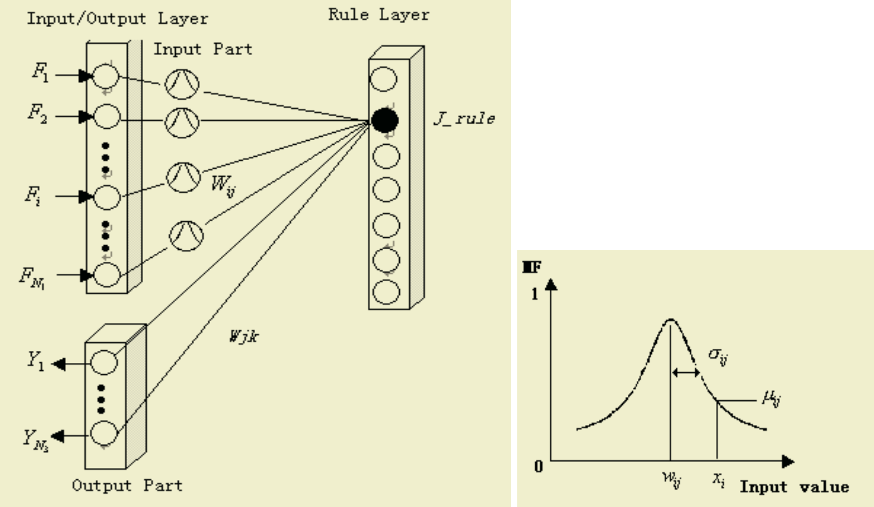


Fig. 2. The structure of the FINN and its membership function

For the prominent region detection problem each segmented image region was described using a set of five dimension features, comprising of the perceptual features defined in section 3. Let $F=(F_1, F_2, F_3, F_4, F_5)$ denote the perceptual features *CSr*, *OCr, SIr, CIr and MIr* of region R_i. For every region, FINNs receives a total of 5-dimensional input data (N_1=5), and outputs the class label: *PR* and *NPR* (N_3=2). Then FINN adjusts the center value and width of its fuzzy membership function automatically during the learning phase. The logical form of the fuzzy inference if-then rules is given as

If f_1 is $\tilde{w}_{1j}$ and f_2 is $\tilde{w}_{2j}$, …, and f_N is $\tilde{w}_{Nj}$ **then** $\widehat{y}_k$ is w^o_{jk} , where $\tilde{w}_{ij}$ means the value near w_{ij} depended on the value of σ_{ij} .

4.2 Learning Process of FINN

The learning process of the FINN consists of the three phases. First, the center values of membership functions which correspond to the if-part and estimated values which correspond to the then-part are determined temporarily in the self-organizing phase. And then, the system merger the similar rules at the rule extraction phase. Finally, Least Mean Square (LMS) is executed to reduce the total mean-square error of the network to finely adjust the weights and the shapes of the membership functions. In our implementation, the self-organizing learning phase and the LMS phase are almost the same as that of FINNs in [10]. In order to get better generalization ability, we employ a relative entropy-based approach to measure similarity between two rules described below.

For two probability densities functions p_k and q_k , the Kullback-Leibler divergence is defined as

$$D_{p_k \| q_k} = \sum_k p_k \log\left(\frac{p_k}{q_k}\right). \tag{6}$$

the Kullback-Leibler divergence indicates how distinguishable p_k is from q_k by maximum likelihood hypothesis testing when the actual data obeys p_x. It is well know that $D_{p_k \| q_k}$ is a nonnegative, additive but not symmetric. To obtain a symmetric measure, one can define similarity measure as:

$$SW(p_k, q_k) = \frac{D(p_k \| q_k) + D(q_k \| p_k)}{2}. \tag{7}$$

Since small values indicate the densities are ‘close’. And then, for each weight vector w_j ($j = 1,...,N_2$), we calculate a six-bin ($N_h = 6$) weight histogram $H_w(j,h)$ ($h = 1,...,N_h$), therefore, estimation of weight probability distribution function is calculated as $p_j(h) = \dfrac{H_w(j,h)}{\sum_{h=1}^{N_h} H_w(j,h)}$. We merge two rules when $SW_{j,j+1}$ is smaller than the threshold δ_w which is selected by experiments described in the Figure 3. Therefore, the FINN can combine rules to extract generalized rules.

5 Experimental Results

The proposed algorithm was integrated into a system that was tested using several video sequences. In order to perform training, we randomly select three video sequences including 1152 frames. Rather than processing every frame of the video clip, our technique analyses only one out of every N (N=10) frames since there is typically little difference among consecutive frames. For each sampled frame, we label the ground truth manually; hence, the training data is composed of 117 sampled frames. As the result of the learning, the FINNs extracted 36 fuzzy inference rules. Since no perceptual theory exists regarding how to select the threshold δ_w in rules extraction phase, the parameter is determined by experimental tests illuminated as Fig.3. The results obtained from three digital video clips taken from a sports video*(Clip1)*, two movies: "*Season I of Friend*s"*(Clip2)* and "*Great Forrest Gump*"*(Clip3)*. Due to the very strong subjectivity of human visual attention, the precision of our method is subjectively examined by ten testers and is averaged. Figure 3 shows the average precision curve with the use of different threshold δ_w , as we can see more reasonable results of our method could be achieved by setting δ_w =0.15.

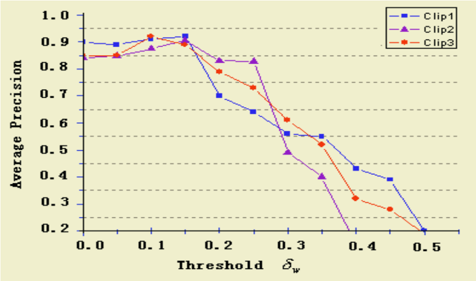


Fig. 3. Average precision related to the estimated threshold δ_w

Fig.4 shows the results for two successive sampled frames taken from movie *Season I of Friend*s. Specifically, Fig.4 (a) shows the original frames, (b) gives corresponding results of homogenous region segmentation, and (c) shows prominent region detection results. Prominent regions are located and used to obtain a binary image that contains white and black pixels to represent prominent region and non-prominent regions, respectively. As shown in the fig.4, one of the background regions not only has high color contrast but also locates at near the center of image, so both of this region and the character Chandler draw more attention and have high perceptual prominence, which are correctly extracted.

We also compared our results for a sports video clip with the results of the algorithm using low-level information (luminance, color, texture) described in [14]. Fig.5 shows the comparison results. As we can see, some noisy areas in Fig.5(b) are removed correctly. Our results also accord to the fact of human selective attention, that

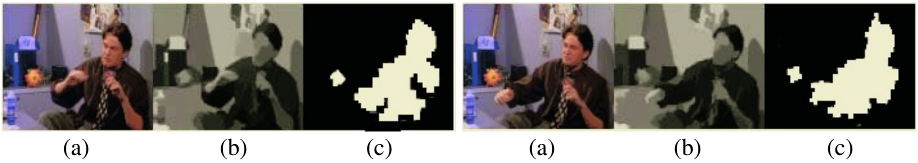


Fig. 4. Prominent region detection in video sequence from *Season I of Friend*s

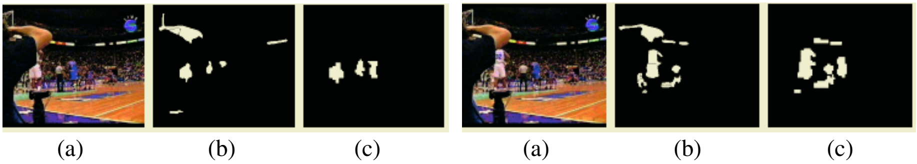


Fig. 5. Prominent region detection results for two frames taken from sports video clip: (a) Original image (b)Detection results using low-level information (c) Our detectioin results

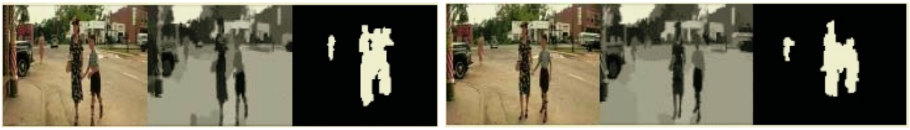


Fig. 6. Prominent region detection in video sequence from "*Great Forrest Gump*"

is, when viewing the video clip, human will put little attention on these noisy areas, even though they have a high contrast. That is different from viewing single image.

In our experiments, several limitations are found. One major problem is caused by the similar motion consistency of the background region as well as the prominent region. As demonstrated in Fig.6, a region of background is assigned mistakenly a high perceptual prominence. However, this drawback can be improved by using spatial relations analysis, which is one of our future works. The shadow region of object is the other limitations of our system, which is difficult to handle because, in many cases, shadow region may locate at the center of an image, mean that it has a higher value of compositional balance inductor. In Fig.6, the shadow region between two characters is regarded as a prominent region mistakenly. Notice that these shadow region almost have poor orientation conspicuity, hence, dynamic feature selection is expected to improve this limitation.

6 Conclusions

A perception-oriented approach for identifying prominent regions in video sequences is presented in this paper. Instead of extraction low-level feature, we extract a number of different perceptual features by the taking advantage of the *mise-en-scene* principles. Furthermore, the promising results are achieved by the development of fuzzy inference neural networks. The technique has been designed to easily accommodate application specific requirements. Although the experimental results show the encouraging performance, the conducted research also shows that there is plenty of room for improvement. Future work will be focused on how to handle the limitations in the approach and improve the results. Additional research work includes a suitable pattern description for the video sequences with the use of the extracted prominent regions.

References

1. Fan, J., Aref, W.G., Elmagamid, A.K., Hacid, M.S., Marzouk, M.S., Zhu, X.: Multi View Multi-level Video Content Representation and Retrieval. J. Electron. Imaging .Special Issue on Multimedia Database, **10** (2001) 895-908
2. Senders, J.: Distribution of Attention in Static and Dynamic Scenes. Proceedings of SPIE, **3026** (1997) 186-194
3. Yarbus, A.: Eye Movements and Vision. Plenum Press, New York, NY (1967)
4. Itt, L.i, Koch, C.: Feature Combination Strategies for Saliency-based Visual Attention Systems. Journal of Electronic Imaging, **10** (2001) 161-169
5. Parkhurst, D., Law, K., Niebur, E.: Modeling the Role of Salience in the Allocation of Overt Visual Attention. Proceedings of ICIP (2003)
6. Bordwell, D., Thompson, K.: Film Art: An Introduction. McGraw-Hill Higher Education (2001)
7. Comaniciu, D., Meer, P.: Mean Shift: A Robust Approach toward Feature Space Analysis. IEEE Trans. Pattern Analysis Machine Intelligence, **24** (2002) 603-619
8. Marcelja, S.: Mathematical Description of the Responses of Simple Cortical Cells. Journal of Optical Society of America, **70** (1980) 1169-1179
9. Ma, Y.F., Zhang, H.J.: A Model of Motion Attention for Video Skimming. Proceedings of ICIP (2002) 22-25
10. Nishina, T., Hagiwara, M.: Fuzzy Inference Neural Network. Neurocomputing, **14** (1997) 223-239
11. lyatomi, H., Hagiwara, M.: Scenery Image Recognition and Interpretation Using Fuzzy Inference Neural Networks. Pattern Recognition, **35** (2002) 1793-1806
12. Li, J., Wang, J.Z., and Wiederhold, G.: Simplicity: Semantics-Sensitive Integrated Matching for Picture Libraries. IEEE Trans. Pattern Analysis Machine Intelligence, **23** (2001)
13. Li, Y., Ma, Y.F., Zhang, H.J.: Salient Region Detection and Tracking in Video. Proceedings of ICME (2003) 269-272
14. Dimai, A.: Unsupervised Extraction of Salient Region-Descriptors for Content Based Image Retrieval. Proceedings of ICIAP (1999) 686-672

The Application of Neural Network and Wavelet in Human Face Illumination Compensation*

Zhongbo Zhang, Siliang Ma, and Danyang Wu

Department of Mathematics, Jilin University
Changchun 130012, China
zhongbozhang@sina.com

Abstract. The performance of face recognition system is greatly affected by the illumination changes. In this article, we propose a method of face illumination compensation based on neural network and wavelet. It sufficiently combines multi-resolution analysis of wavelet and the self-adaptation learning and good spread ability of BP neural network, thus this method carries out the face illumination compensation. The theory and experiments prove that this method solves the problem of illumination compensation efficiently in the face detection and recognition process. It improves the face detection and recognition under different illumination conditions. Moreover, it has good robustness and can be used in a wide range.

Keywords: Neural network; Wavelet analyze; Illumination compensation; Face recognition; Face detection.

1 Introduction

To meet the needs of individual security and the fast development of computer science, the biometric techniques have been applied in many fields for personal identification. Among them, studies of face detection and recognition are very popular because of its simplicity, no touch and no violation of privacy, which make it different from other biometric techniques. However, little work has been done on illumination compensation. Traditional face detection algorithms presume that the illumination is uniform, but unfortunately it's not the way in the natural world. It may become too bright or too dark because of the deflective light and sidelight, and detection rates commonly drop quickly under these conditions. Both FERET [1] test and FRVT [2] test show the illumination changes is still one of the bottlenecks for practical face recognition system. So we investigated the compensation of the illumination in this study.

Illumination compensation is always a problem important but difficult to solve in face recognition. Mose and others have proved theoretically that there is no function to keep constant for illumination [3]. In their research, Warrington and Taylor verified that the process of compensating for illumination changes

* Supported by the National High-Tech Research and Development Plan of China under Grant No.2004AA001110(863).

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 828–835, 2005.

involves not only the primary visual cortex, but also the participation of higher-level areas [4]. Generally, the illumination compensation methods include [5] extracting the constant features of the illumination [6], [7], setting up a model for the illumination changes [4], [8], [9], [10], and normalizing the illumination condition [11], [12], [13].

The first method is to extract the facial features, which do not change with or at least are not sensitive to the illumination variation. But Mose and Adini's have proved theoretically and experimentally that there is no expression to overcome the illumination changes [3], [14], indicating that it is impossible to extract any face features without being affected by the illumination changes. We can only extract the features not sensitive to the illumination changes. The gradient direction feature was used as the constant illumination feature in one study [6], while the constant illumination feature was the ratio image feature in another study [7]. But they all have some limitations in the practice of illumination compensation.

The second method is to set up a model for the illumination changes. The fundamental philosophy is to use a proper subspace to represent the changes due to the illumination. The process is to estimate the parameters in the test face model and then choose the subspace closest to the test face.

The third method is to eliminate the influence of the illumination changes before a face image is recognized, and then the normalized face is recognized. A typical example is SFS (shape-from-shading) method. After the illumination direction and the reflection rate were estimated in the surrounding, the shape of the human face is reconstructed [13]. The idea of SFS is very good, but it has some inherent shortcomings: firstly the model becomes too simple after being highly simplified; secondly it can only apply to single dot light which might not be satisfied in practice; thirdly the non-face images can be changed into human faces after the compensation; fourthly it is computationally expensive.

We used the third method in our study and propose a face compensation approach based on wavelet and neural network, from rough to accurate and from whole to local. A rough linear illumination compensation was first performed for the given face image, which can only compensate the lower frequency features and the effect is limited. The higher frequency features can't be compensated, as seen in the frequency analysis. But it can reduce the learning pressure of the neural network, accelerate the convergent rate and improve the learning accuracy as well as the extensibility of the network. Imitating results show this step is very important. To refine the result obtained, the face image is compensated based on the wavelet and neural network. This step compensates the different scale features of the face image by using the multi-resolution characteristic of the wavelet and learning ability of the neural network. Experimental results show that our algorithm has almost overcome the shortcomings in the methods mentioned above and improved the effect of illumination compensation greatly as well as the face detection and face recognition under different illumination.

2 Algorithm

2.1 Eliminating the Sidelight by Homomorphic Filtering Roughly

Most objects in the natural world are the ideal refuse reflection bodies, including human face. Lambert's cosine law concludes the reflection rule of an ideal refuse reflection body under a dot light. According to Lambert's law, the refuse reflection light intensity from an ideal refuse reflection body is in direct ratio [18] to cosine of the angle between the reflection light and the normal vector,that is

$$I(x,y) = k_d(x,y)I_l(x,y)\cos(\theta(x,y)) \tag{1}$$

where $I(x,y)$ is the light intensity at point (x,y), $k_d(x,y)$is the refuse reflection rate at point (x,y) on the surface, $I_l(x,y)$ is the intensity of the reflect light at the point (x,y), $\theta(x,y)$ is the angle between the reflection light and normal vector of surface. Because Lambert's refuse model is a multiplication model, the linear characteristic of the addition model doesn't work well to eliminate the light. Thus it should be transformed into a homomorphism region; here it is taken the logarithm. We have

$$J(x,y) = \log I(x,y) \tag{2}$$

Here $J(x,y)$ is the light intensity at (x,y) in the homomorphism region. We use a plane in the three-dimension space to fit $J(x,y)$, then subtract the fitting plane in the three-dimension space from $J(x,y)$ to eliminate the effect of the sidelight on the face. That is: let the plane in the three-dimension space be , assuming the following minimization function: $z(x,y) = ax + by + c$, assuming the following minimization function:

$$(a,b,c) = \underset{(a,b,c)}{\arg\ \min}\left(\sum_{i=1}^{n}(ax_i + by_i + c - J(x_i,y_i))^2\right) \tag{3}$$

By Lagrange's method, we have

$$A = (X^TX)^{-1}X^TB \tag{4}$$

where

$$A = \begin{pmatrix} a \\ b \\ c \end{pmatrix}\ X = \begin{bmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix}\ B = \begin{pmatrix} J(x_1,y_1) \\ \vdots \\ J(x_n,y_n) \end{pmatrix} \tag{5}$$

After compensated in the homomorphism region, we have

$$J_0(x_i,y_i) = J(x_i,y_i) - z(x_i,y_i) \tag{6}$$

Let

$$I_0(x_i,y_i) = e^{J_0(x_i,y_i)} \tag{7}$$

which is the compensated face image in the homomorphism region. Finally a linear transform on gray level makes each compensated image $I_1(x,y)$ have the fixed mean and deviation. The zoomed face image is performed illumination compensation based on the wavelet and neural network [9].

2.2 Image Wavelet Analysis

Firstly we perform a wavelet analysis on $I_1(x, y)$ to get LLI_2, LHI_2, HLI_2, HHI_2, which present the horizontal lower frequency vertical lower frequency information horizontal lower frequency vertical higher frequency information horizontal higher frequency vertical lower frequency information horizontal higher frequency vertical higher frequency information respectively. Then LLI_2, LHI_2, HLI_2 are analyzed again to get the corresponding horizontal lower frequency vertical lower frequency information LLI_{2LL}, LHI_{2LL}, HLI_{2LL} . We adjust data LLI_{2LL}, LHI_{2LL}, HLI_{2LL} to compensate illumination on face image (LL represents the horizontal lower frequency vertical lower frequency information of the image, LH horizontal lower frequency vertical higher frequency information of the image, HL represents horizontal higher frequency vertical lower frequency information of the image, HH horizontal higher frequency vertical higher frequency information of the image).

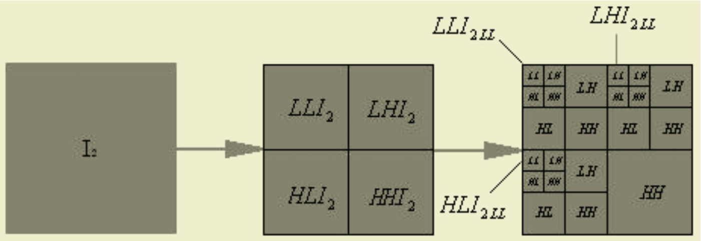


Fig. 1. Wavelet analysis of facial illumination compensation

We have three reasons to do so:
1. The influence of illumination on the face image mainly focuses on the medium lower frequency information of the image.
2. The amount of the date in LLI_{2LL}, LHI_{2LL}, HLI_{2LL} is small, which is helpful to increase the efficiency of BP network and maintain the stability of the network.
3. More importantly, it fully protects the detail information of face image and improves the robustness of illumination compensation.

2.3 The Structure and Training of the Neural Network

We design a three-layer BP neural network, which are input layer, hidden layer and output layer. Let's call InputLayerCount the number of neurons in the input layer, HiddenLayerCount the number of neurons in the hidden layer, OutpurLayerCount the number of neurons in the output layer. The structure of the three layers are the same and the topology structure of each layer is full connect mode.They are used to train LLI_{2LL}, LHI_{2LL}, HLI_{2LL} data respectively. The three BP networks are $BPLLI_{2LL}$, $BPLHI_{2LL}$, $BPHLI_{2LL}$ separately.

The InputLayerCount and OutpurLayerCountof the three BP networks equal the dimensions of data $BPLLI_{2LL}$, $BPLHI_{2LL}$, $BPHLI_{2LL}$(which are decided by the size of the image, the type of the wavelet, and the length of the compact support of the wavelet. In this example, the size of the image is 64×64, the wavelet is Bior3.3, so InputLayerCount=OutpurLayerCount=14×14=196).

The hidden layer can be considered to extract the features of the input pattern via the self-organization of the connection weight value between the input layer and hidden layer, and then transfer them to the output layer. There is no theory about how to choose HiddenLayerCount now. Generally, if the HiddenLayerCount is too big, the network will be too redundant. That means it will take more time to train the network and reduce the training times for the network to be convergent while also reduce the extendibility of the classifier.

If the HiddenLayerCount is too small, it will reduce the time to train the network while it may lead to divergence and reduce the extendibility of the classifier. To ensure the stability of the classifier and expendability of the network, and avoid the problems such as over training of the neural network, we set a relative small HiddenLayerCount originally. After studying several times, if it isn't convergent we will increase the HiddenLayerCount until we get reasonable number of the hide units. According to the experiment and test, we let HiddenLayerCount be 200.

We use a subset [15] of PIE database as the training data, which contains 20 people and each has 21 different kinds of illumination. Some samples of two people in the set under different illumination condition are shown below. In the training process, we choose a sample of the same person under the normal illumination and calculate $StandardLLI_{2LL}$, $StandardLHI_{2LL}$, $StandardHLI_{2LL}$, serving as the goal output of the three BP network $BPLLI_{2LL}$, $BPLHI_{2LL}$, $BPHLI_{2LL}$ separately, which keep constant under different illumination conditions. LLI_{2LL}, LHI_{2LL}, HLI_{2LL} computed from the samples under the different illumination conditions are set to the input of the three BP networks. $StandardLLI_{2LL} - LLI_{2LL}$, $StandardLHI_{2LL} - LHI_{2LL}$, $StandardHLI_{2LL} - HLI_{2LL}$ are the goal output of $BPLLI_{2LL}$, $BPLHI_{2LL}$, $BPHLI_{2LL}$.

Why don't we use $StandardLLI_{2LL}$, $StandardLHI_{2LL}$, $StandardHLI_{2LL}$ to be the goal output of the three BP network $BPLLI_{2LL}$, $BPLHI_{2LL}$, $BPHLI_{2LL}$? There are two reasons: first of all it avoids transforming a non-face image to be a face image. When we compensate a real face image, we add the output of the neural network to the input of the neural network.

Let LLI_{2LL} +(the output of $BPLLI_{2LL}$), LHI_{2LL}+(the output of $BPLHI_{2LL}$), HLI_{2LL} +(the output of $BPHLI_{2LL}$)be the medium result of the compensation instead of the direct output via the neural network. Let the output of $BPLLI_{2LL}$, the output of $BPLHI_{2LL}$, the output of $BPHLI_{2LL}$ be the medium result of the compensation. This is the unique advantage compared with any other face illumination compensation.

Secondly $StandardLLI_{2LL} - LLI_{2LL}$, $StandardLHI_{2LL} - LHI_{2LL}$, $StandardHLI_{2LL} - HLI_{2LL}$ are used to be the training output of the three networks $BPLLI_{2LL}$, $BPLHI_{2LL}$, $BPHLI_{2LL}$. That will reduce the redundancy of the destination space, increase the convergence speed, and improve the

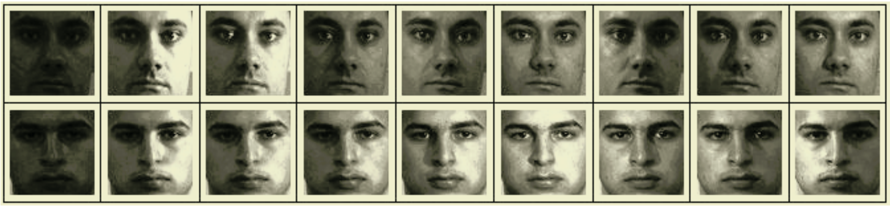


Fig. 2. Samples of two people in the database under different illumination conditions

learning accuracy and network extensibility. So when the three networks become convergent after training, we can perform the face illumination compensation.

3 Proposed Implementation

1. The face image is normalized to 64×64.
2. The normalized image is preprocessed in text way.
3. The pre-processed image is implemented wavelet analysis to get LLI_{2LL}, LHI_{2LL}, HLI_{2LL}.
4. Input LLI_{2LL}, LHI_{2LL}, HLI_{2LL} into the BP network $BPLLI_{2LL}$, $BPLHI_{2LL}$, $BPHLI_{2LL}$ respectively and get the corresponding output $OutBPLLI_{2LL}$, $OutBPLHI_{2LL}$, $OutBPHLI_{2LL}$.
5. Substitute $LLI_{2LL} + OutBPLLI_{2LL}$, $LHI_{2LL} + OutBPLHI_{2LL}$, $HLI_{2LL} + OutBPHLI_{2LL}$for LLI_{2LL}, LHI_{2LL}, HLI_{2LL} separately.
6. We get LLI_2, LHI_2, HLI_2 by using data preserved and wavelet reconstruction method on LLI_{2LL}, LHI_{2LL}, HLI_{2LL} and get image I_2 by using data preserved and wavelet reconstruction method on LLI_2, LHI_2, HLI_2. The image I_2 is the illumination compensated one.

4 Results

The results of experiment indicate our approach solves the illumination compensation problem quite well. It has been applied successfully in our real-time face detection and recognition system (face detection in complex background and eyes fixed 35-40 frames/s). Particularly, our system compensates illusion on the test face in the face detection and doesn't change the non-face images into face image. And the detection rate increase markedly without increasing the false rate.

5 Conclusions

In this paper, we introduced a method of face illumination compensation based on neural network and wavelet. It sufficiently combines multi-resolution analysis of wavelet and the self- adaptation learning and good spread ability of BP neural

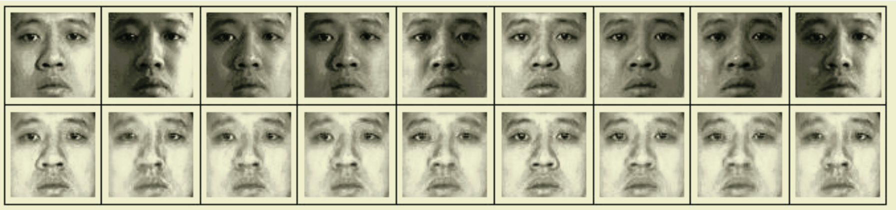


Fig. 3. Some results of the algorithm applied in illumination compensation

network. Extensive study has been performed on how to solve the problem of illumination compensation in face detection and recognition. The experiments results indicate that our algorithm has almost overcome the shortcomings in the methods mentioned in other literature and improved the effect of illumination compensation greatly as well as the face detection and recognition under different illumination. This method has been integrated in our face detection and tracking system successfully.

References

1. Phillips, P.J., Moon, H., et al: The FERET Evaluation Methodology for Face-Recognition Algorithms. IEEE TPAMI, **22** (2000) 1090-1104
2. Phillips, P.J., Grother, P., Micheals, R.J., Blackburn, D.M., Tabassi, E., Bone, J.M.: FRVT (2002) Evaluation Report. http://www.frvt.org/DLs/FRVT_2002_Evaluation_Report.pdf
3. Moses, Y., Ullman, S.: Limitation of Non-Model-Based Recognition Schemes Springer-Verlag, [A]. Proc. ECCV-92, Sandini G., ed [C], (1992) 820-828
4. Warrington, E., Taylor, A.: Two Categorical Stages of Object Recognition. (Perception) **7** (1978) 152-164
5. Sim, T. and Kanade, T.: Combining Models and Exemplars for Face Recognition: An Illuminating Example. CVPR, (2001)
6. Chen, H. F., Belhumeur, P. N., Jacobs, D. W.: In Search of Illumination Invariants. CVPR, (2000)
7. Riklin-Raviv T., Shashua, A.: The Quotient Image: Class Based Recognition and Synthesis Under Varying Illumination Conditions. CVPR, (1999)
8. Basri, R., Jacobs, D. L.: Reflectance and Linear Subspaces. Proc. Eighth IEEE Int'l Conf. Computer Vision, (2001) 383-390
9. Georghiades, A. S., Belhumeur, P. N., Kriegman, D. J.: From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose. IEEE Trans. PAMI, **23** (2001) 643-660
10. Murase, H. Nayar, S.: Visual Learning and Recognition of 3D Object from Appearance. IJCV, **14** (1995) 5-24
11. Shashua, A., Riklin-Raviv, T.: The Quotient Image: Class-Based Re-Rendering and Recognition with Varying Illuminations. IEEE Trans. on PAMI, **23** (2001) 129-139
12. Phillips, P.J., Vardi, Y.: Efficient Illumination Normalization of Facial Images. PRL, **17** (1996) 921-927

13. Zhao, W., Chellappa, R.: Illumination-insensitive Face Recognition Using Symmetric Shape-from-Shading. Proc. Conf. Computer Vision and Pattern. Recog, (2000) 1286-1293
14. Adini, Y., Moses, Y., Ullman, S.: Face Recognition: The Problem of Compensating for Changes in Illumination Direction. IEEE Transactions on Pattern Analysis and Machine Intelligence, (1997) 721-732
15. Terence, S., Simon, B., Maan, B.: The CMU Pose, Illumination, and Expression (PIE) Database of Human Faces. Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, (2001)
16. Peng, Q.S., Bao, H.J., Jin, X.G.: Procedural Elements for Realistic Image Synthesis. Peking: Science Press, (1999)

Global Icons and Local Icons of Images Based Unit-Linking PCNN and Their Application to Robot Navigation*

Xiaodong Gu and Liming Zhang

Department of Electronic Engineering, Fudan University, Shanghai 200433, China
guxiaodong@263.net

Abstract. PCNN-Pulse Coupled Neural Network is a new artificial neural network supported by biological experimental results. Unit-linking PCNN we have developed from PCNN can be efficiently used for image processing. Using Unit-linking PCNN can produce icons of images, namely time signatures of images. This paper describes how to use Unit-linking PCNN to produce global icons, and local icons of images. A global icon of an image based Unit-linking PCNN has the translation, rotation invariance. In some situations, such as object detection, this invariance is an advantage. However, in other situations, such as navigation, Unit-linking PCNN icons should reflect the local changes of scenes and here the invariance of the global icon is not the advantage. Therefore we also introduce the local icons of images based Unit-linking PCNN, which can reflect the local changes of images and have no translation, rotation invariance that global icons have. In the meantime, the robot navigation based global and local icons is shown as an example.

1 Introduction

Eckhorn introduced the linking field network [1] based on the phenomena of synchronous pulse bursts in the cat visual cortex in 1990. After three years, Johnson developed PCNN (Pulse Coupled Neural Network)[2] based on the linking model. PCNN can be applied in many fields, such as image processing, object detection, optimization [3],[4],[5],[6],[7],[8], as other neural networks for example Cellular Neural Network [9]. We introduced Unit-linking PCNN [10] based on PCNN and applied Unit-linking PCNN in image processing, such as image thinning [11]. Using Unit-linking PCNN can generate one-dimensional time series of output pulses in response to two-dimensional images. This one-dimensional time series are also called time signatures of images, or icons of images. An icon corresponding to a whole image is called a global icon, which is invariant against changes due to translation and rotation. In some cases, such as object detection, this invariance is an advantage. However, in other situations, such as navigation, the Unit-linking PCNN icon should

* This research was supported by China Postdoctoral Science Foundation (No.2003034282), Shanghai Postdoctoral Science Foundation, National Natural Science Foundation of China (No.60171036, 30370392), and Key Project of Shanghai Science& Technology Committee (No.045115020).

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 836–841, 2005.

reflect the local changes of the scene and here the invariance of the global icon is not the advantage again. If divide an image into many parts and these parts can be overlapped, every icon corresponding every part is called the local icon. An image divided into n parts corresponds to n local icons and one-to-one correspondence exists between a part and a local icon. Local icons can reflect the local changes of images and have no translation, rotation invariance that global icons have.

In Section 2, Unit-linking PCNN is introduced. In Section 3, global icons and local icons of images based on Unit-linking PCNN are described in detail respectively, and they are used together in robot navigation. The last Section is conclusions.

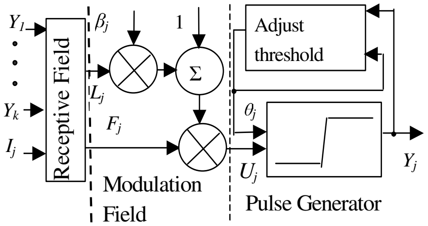


Fig. 1. A Unit-linking PCN

2 Unit-Linking PCNN

$$L_j = \text{Step}(\sum_{k\in N(j)} Y_k) = \begin{cases} 1, & if \quad \sum_{k\in N(j)} Y_k > 0 \\ 0, & else \end{cases}. \tag{1}$$

$$\frac{d\theta_j(t)}{dt} = -\alpha_j^T + V_j^T Y_j(t), \quad \textit{the lower limit of integration is just before the last firing in solution}. \tag{2}$$

$$Y_j = \text{Step}(U_j - \theta_j) = \begin{cases} 1, & if \quad U_j > \theta_j \\ 0, & else \end{cases}. \tag{3}$$

We proposed the Unit-linking PCNN based on the PCNN. A Unit-linking Pulse Coupled Neuron (PCN) consists of three parts: the receptive field, the modulation field, and the pulse generator. Fig.1 illustrates a Unit-linking PCN j. It has two channels. One channel called F channel is feeding input (F_j); the other called L channel is linking input (L_j). $I_j,Y_1,\ldots,Y_k$ are input signals of neuron j. Y_j is the output pulse of neuron j. I_j, an input signal from the external source, only inputs to F channel of j (F_j $=I_j$) .$Y_1,\ldots,Y_k$, which are output pulses emitted by other neurons connected with j , only input to L channel of j (see Eq.(1)).In Eq.(1), $N(j)$ is the neighbor field of neuron j and j does not belong to $N(j)$.In a Unit-linking PCN, we introduce the concept of Unit-linking, namely as to neuron j, when one or more than one neurons fire in its neighbor field, L_j is equal to 1. Compared with traditional PCNs, this reduces parameters and makes the linking inputs of Unit-linking PCNs uniform so that the behaviors of networks consisting of Unit-linking neurons are easy to analyze and control. The uniform L_j is added a constant positive bias firstly. Then it is multiplied by F_j and the bias is taken to be unity ($U_j= F_j(1+\beta_j L_j)$). β_j is the linking strength. The total internal activity U_j is the result of modulation and it inputs to the pulse generator. If U_j is greater than the threshold θ_j, the neuron output Y_j turns into 1 (see Eq.(3)). Then Y_j feedbacks to make θ_j rises over U_j immediately so that Y_j turns into 0. Therefore, when U_j is greater than θ_j, neuron j outputs a pulse. On the other hand, θ_j drops lineally with time increasing. In Eq.(2) V_j^T and α_j^T are the amplitude gain and the time constant of the threshold adjuster . A Unit-linking PCNN consists of Unit-linking neurons and one of those neurons is show in Fig.1.

3 Global Icons and Local Icons Based Unit-Linking PCNN

When Unit-linking PCNN is used to produce global icons and local icons of images, it is a single layer two-dimensional array of laterally linked neurons. One-to-one correspondence exists between image pixels and neurons. The feeding channel of each neuron receives the intensity of the corresponding pixel. Meanwhile, each neuron is connected with neurons by the linking channel in its 8-neighbor field. Each neuron has the same parameters

3.1 Global Icons of Images

In this paper, using Unit-linking PCNN to produce icons of images, for a grey image, the iteration time is 20. Record the sum of neurons that fired at each time in sequence to obtain a one-dimensional vector. This vector consisting of 20 elements is the global icon of the image. Each element is equal to the number of neurons that fired at this level based on intensities and linking waves. The global icon is the pulse train as a function of time. The global icon reflects the image intensity pattern related to the geometry of the image. Obviously the global icon has the translation, rotation invariance because each neuron is identical and has the same connection mode. As to the object in the image, the global icon is insensitive to translation and rotation.

Before describing the algorithm of obtaining the global icon based Unit-linking PCNN, we introduce the symbols that will be used in the algorithm. **F** is a feeding input matrix saving the input image. **L** is a binary linking input matrix, where **L**=Step(**Y** $\otimes$ **K**) , where **Y** is a binary neuron output matrix and **K** is a 3*3 kernel , the center element of **K** is 0, the other elements of **K** are 1.**U** is an internal activity matrix and $\mathbf{\Theta}$ is a threshold matrix. $\mathbf{\Delta}$ is the threshold decay matrix, every element of which is the same constant δ. **Inter** is the matrix, which is used in following algorithm. **F**, **L, U, Y**, **Inter**, $\mathbf{\Theta}$**,** and $\mathbf{\Delta}$ have the same dimensions equal to the image size. **T**, an one-dimensional matrix including 20 elements, saves the sum of neurons that fired at each time. **T**(n) is the sum of neurons fired at time n. '.*' indicates array multiplication and '$\otimes$' indicates two-dimensional convolution. β is the linking strength and each neuron has the same β. *Height* and *Width* are the height and width of the image.

Global icon algorithm based on Unit-linking PCNN is described below.

(1) Original image **F** is normalized between 0 and 1. **L = 0.U = 0. Θ= 1.Y =0.** Give n=0, β=0.2,N=20,δ=1/N=0.05.

(2) **L**= Step(**Y** $\otimes$ **K**).

(3) **Inter= Y**,**U=F.*(1+**β**L**), **Y**=Step(**U** -**Θ**).

(4) If **Y**= **Inter**, go to (5); else, **L**=Step(**Y** $\otimes$ **K**), go back to (3).

(5) Conclude the sum of neurons fired at time *n* based **Y** and save this sum to **T**(n).

(6) **Θ**=**Θ**-**Δ**. Decrease threshold with the increasing of the iteration times.

(7) If **Y**(i,j) =1, **Θ**(i,j) =100.(i=1,…,*Height*; j=1,…*Width*)
Y(i,j)**,** **Θ**(i,j) are elements of **Y, Θ** respectively. When one neuron has fired, increase its threshold to prevent from firing again.

(8) n=n+1. If n<N+1, go back to (2); else end.

The algorithm also can be used for color images by dividing a color image into three channels(R, G, B). For each color channel, we can obtain the corresponding global icon consisting of 20 elements. Simply connecting these three global icons obtains the global icon of the color image consisting of 60 elements.

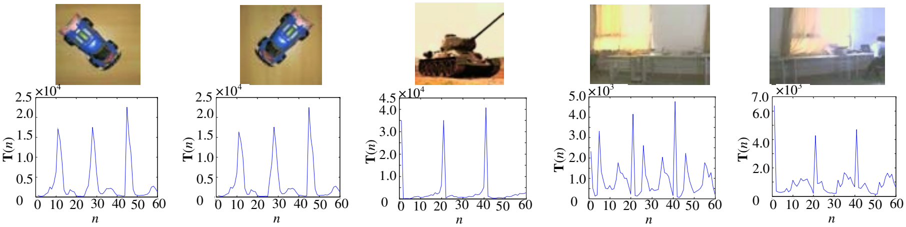


Fig. 2. Five color images in the first row and their global color icons in the second row. One-to-one correspondence exists between images and global color icons

In the first row of Fig.2, the left one is a color car image (256*256), the 2nd one is the result of rotating the left one 90° clockwise, and the 3rd one is a color image of a tank (256*256). The 4th one and 5th one (160*120) are the 10th frame and 50th frame of the lab video with 20 frame/s caught by the robot in the navigation. In Fig.2, there are global color icons under the corresponding color images. In global icons, the horizontal axis (*n*) indicates the iteration time and the vertical axis (T(*n*))indicates the sum of neurons fired at each time. Each color global icon consists of 60 elements. The global icons of the 1st one and the 2nd one are almost identical due to rotation invariance of global icons. In practice, the small rotation error caused by pixel discretization exists between the global icon of the 1st one and that of the 2nd one. The global icons of the1st one, the 3rd one, the 4th one, and the 5th one are very different.

3.2 Local Icons of Images

A global icon of an image based Unit-linking PCNN has the translation, rotation invariance. In some cases, such as object detection, this invariance is an advantage. However, in other cases, such as navigation, the icon should reflect the local changes of the scene and here the invariance of the global icon is not the advantage again.

Therefore in this paper we introduce the local icons of images based Unit-linking PCNN, which can reflect the local changes of images and have no translation, rotation invariance that the global icons have. Dividing an image into many parts, which can be overlapped, every icon corresponding to every part is called the local icon. An image is divided into *n* parts corresponds to *n* local icons and one-one correspondence between a part and a local icon. Local icons can reflect the local changes of images and have no translation, rotation invariance that global icons have.

Here still use the color car images shown in Fig.2 to illustrate the local icons. In Fig.2, the global icons of the left one and the 2nd one are almost identical due to rotation invariance of global icons. Here, divide each image into 4 even parts (see Fig.3) so that each image has 4 local icons. Each local color icon in the left one is very different from the corresponding part of the right one although global color icons of these two images are almost identical. For brevity, only the local icon of part 2 of each image is shown near the corresponding image. In this example, because each part in the same image is not overlapped, for the same image the sum of 4 local icons of 4 parts is almost identical to the global icon. For the image divided into non-overlapped parts, this conclusion is correct.

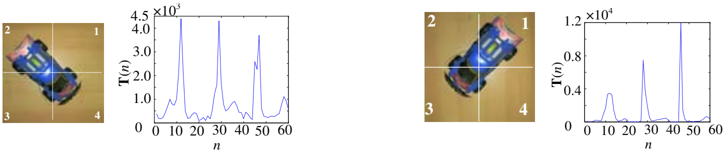


Fig. 3. The two color car images in Fig.2, which almost have the same global color icons, are divided into 4 even parts respectively and the local color icon of part 2 of each image is on the right to the corresponding image

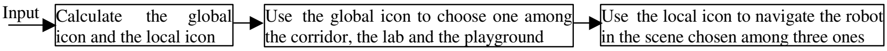

Input
Calculate the global icon and the local icon
Use the global icon to choose one among the corridor, the lab and the playground
Use the local icon to navigate the robot in the scene chosen among three ones

Fig. 4. The block diagram of the robot navigation based global icons and local icons

3.3 Application of Global Icons and Local Icons of Images in Robot Navigation

We have used both global icons and local icons in robot navigation. The input images are color images and the size of each input image is 160*120. Each image is divided into 12 even non-overlapped parts. The size of each part is 40*40.For each image, first calculate 1 global icon and 12 local icons. Fig.4 illustrates how to use global icons and local icons based Unit-linking PCNN to navigate the robot. The global icon and local icons of the same image can calculate at the same time. Then use the global icon to choose one kind of scene from three different kinds of scenes (the corridor, the lab and the playground). After having obtained the certain kind of scene, use local icons to navigate the robot further. In practice, in order to save time, only two color channels are used and the iteration time of each channel reduces from 20 to12 based on experiment. In this situation, the robot can process 2 color images per second based a PIII1.7GHz computer. More research work needs to do to save time further.

4 Conclusions

Using Unit-linking PCNN can produce global icons, and local icons of grey or color images. A global icon has the translation, rotation invariance. This invariance is an advantage in some situations, such as object detection. However, in other situations, such as navigation, the icon of the image should reflect the local changes of thee scene and here the invariance of the global icon is not the advantage. Local icons we proposed in this paper can reflect the local changes of images. As an example, global icons and local icons are together applied in robot navigation. The further research may pay attention to the reduction of execution time, and the combination of global icons and local icons based Unit-linking PCNN in other applications.

References

1. Eckhorn, R., Reitboeck, H.J., Arndt, M., et al.: Feature Linking Via Synchronization among Distributed Assemblies: Simulation of Results from Cat Cortex. Neural Computation, **2** (1990) 293-307
2. Johnson, J.L., Ritter, D.: Observation of Periodic Waves in a Pulse-coupled Neural Network. Opt.Lett., **18** (1993) 1253-1255
3. Johnson, J.L., Padgett, M.L.: PCNN Models and Applications. IEEE Trans. on Neural Networks, **10** (1999) 480-498
4. Kuntimad, G., Ranganath, H.S.: Perfect Image Segmentation Using Pulse Coupled Neural Networks. IEEE Trans. on Neural Networks, **10** (1999) 591-598
5. Ranganath, H.S., Kuntimad, G.: Object Detection Using Pulse Coupled Neural Networks. IEEE Trans. on Neural Networks, **10** (1999) 615-620
6. Gu, X.D., Guo, S.D., Yu, D.H.: New Approach for Noise Reducing of Image Based on PCNN. Journal of Electronics and Information Technology (Chinese), **24** (2002) 1304-1309
7. Caulfield, H.J., Kinser, J.M.: Finding Shortest Path in the Shortest Time Using PCNN's. IEEE Trans. on Neural Networks, **10** (1999) 604-606
8. Johnson, J.L.: Pulse-coupled Neural Nets: Translation, Rotation, Scale, Distortion and Intensity Signal Invariance for Images. Applied Optics, **33** (1994) 6239-6253
9. Zhao J.Y., Wang H.M., Yu D.H.: A New Approach for Edge Detection of Noisy Image based on CNN. International Journal of Circuit Theory and Applications, **31** (2003) 119-131
10. Gu, X.D., Guo, S.D., Yu, D.H.: A New Approach for Automated Image Segmentation Based on Unit-linking PCNN. Proceedings in IEEE International Conference on Machine learning and Cybernetics, Beijing, China (2002) 175-178
11. Gu, X.D., Yu, D.H., Zhang, L.M.: Image Thinning Using Pulse Coupled Neural Network. Pattern Recognition Letters, **25** (2004) 1075-1084

A Neural Model for Extracting Occluding Subjective Surfaces

Keongho Hong and Eunhwa Jeong

Information and Communication Division, Cheonan University
115, Anseodong, Cheonan, Chungnam, 330-794, Republic of Korea
{khhong,ehjeong}@cheonan.ac.kr

Abstract. This paper studied a model that is able to extract occluding surfaces of subjective contour figures based on the mechanism of feature extraction found in a visual system. A common factor in all such subjective contour figures, such as the Kanizsa triangle is having a surface occluding part of a background, i.e. subjective contours are always accompanied by subjective surfaces. In this paper we propose a neural network model that predicts the shape of subjective surfaces. This model employed an important two-stage process of the Induced Stimuli Extraction System (ISES) and Subjective Surfaces Perception System (SSPS). The former system extracted the induced stimuli for the perception of subjective surfaces, and the latter formed the subjective surfaces from the induced stimuli. The proposed model is demonstrated on a variety of Kanizsa-type subjective contour displays. The results of the experiment showed that the proposed model was successful not only in extracting the induced stimuli for the perception of subjective contours, but also in perceiving the subjective surface from the induced stimuli.

1 Introduction

It is common to emphasize the importance of image contours because of their relationship to object boundaries and surface discontinuities in the scene (e.g., Marr, 1982). Often, object boundaries and surface discontinuities exist as luminance changes in the image, but this is not always the case. Although we frequently perceive clear perceptual boundaries between image regions, the physical bases for these percepts might be very slight. Schumann (1904) described the first experiments with stimuli in which contours are perceived without intensity gradients (See Fig.1). This class of figures was virtually forgotten until fifty-five years later when Kanizsa (1955) created a series of new, and more powerful variants of this type of figure which he called "quasi-perceptive margin figures" and which have since been called "subjective contour", "illusory contour", and "anomalous contour" figures.

These kinds of subjective contour figures are shown in Fig.1. The Kanizsa square in (a), the distorted triangle in (b), the foot in (c) and the vertical boundary and circle in (d) is all defined by boundaries that are not made explicit. The Kanizsa-type subjective contours (Fig.1 (a), (b) and (c)) and offset grating contours (Fig.1 (d)) are

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 842–847, 2005.

among the most commonly studied subjective contours. Our experimental model deals with the Kanizsa-type subjective contour.

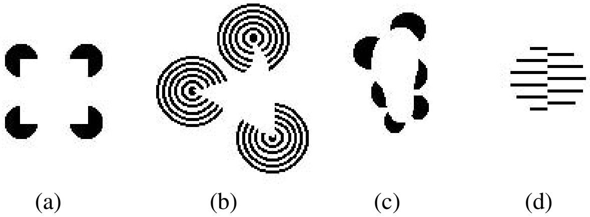


Fig. 1. Some examples of subjective contours. The Kanizsa square in (a), the distorted triangle in (b), the foot in (c), the vertical boundary and circle in (d) are all defined by boundaries that are not made explicit

Many computational models have been proposed to describe the formation of subjective contours, including Ullman (1976), Heitger & von der Heydt (1993), Kellman & Shipley (1995) Williams and Hanson (1994), Williams & Jacobs (1995) etc. Unlike these computational models, we proposed a neural network model for perceiving subjective surfaces including depth sensations based on the mechanism of a mammal's visual pathway.

Our neural network model can be divided into two systems which are described as neural networks with multiple layers: (1) the Induced Stimuli Extraction System (ISES) for the perception of subjective contours including depth information and (2) the Subjective Surfaces Perception System (SSPS). The induced stimuli extraction system (ISES) extracts the induced stimuli needed for the perception of occluded surfaces from subjective contours, except for the inducers (background images), which in turn hide parts of the pattern. The subjective surfaces perception system (SSPS) forms the subjective surfaces, which must always be seen to be above the plane of the inducers, from the induced stimuli. Each system composes neural network architecture with multiple layers.

2 Feature Extraction of Visual System

The mammalian visual system receives input in the form of visible light. Photoreceptors in the retina absorb this light, emitting neural signals on the process, which, in turn, stimulates bipolar and retinal ganglion cells. Information from each retina passes along the optic nerves, though the optic chasm and into the lateral geniculate nucleus (LGN). The cells within the LGN do not appear to exert any profound transformation upon the neural information they receive other than organizing retinal signals into right and left visual field components. From the LGN, information is topographically mapped onto the visual cortex where it is further processed by simple, complex, and hypercomplex cells.

The terms "simple", "complex" etc. refer to the types of stimuli that elicit responses from these from these cells. In a serial model of vision processing, the visual input required to activate a cell within the cortex becomes progressively more complicated further along the visual pathway. The hypothesis made by Hubel & Wiesel was that information was processed by a cascade of cells : LGN to simple cells, simple cells to complex cells, and complex cells to hypercomplex cells. Though there have been some different opinions to Hubel & Wiesel's serial model in the visual pathway, the architecture of the proposed model is still based on the ideas developed by Hubel and Wiesel.

3 Perception of Subjective Surfaces

Image Acquisition (layer U_0) The color classification in subjective contour figures is not of great significance. Thus, for simplification, we remove the process of color classification, and simply convert a color image to a binary one.

Contrast Extraction by LGN (layer U_1) The system first detects low-level feature, such as contrast in the binary image (layer U_0). This work performs by spatial filtering. A set of two-dimensional DOG (Difference Of two Gaussians) functions with an adequate space constant is used for the spatial filters. The function is defined by:

$$DOG(x,y) = \frac{1}{2\pi\sigma_e^2} e^{-r^2/2\sigma_e^2} - \frac{1}{2\pi\sigma_i^2} e^{-r^2/2\sigma_i^2} \tag{1}$$

where r represents distance from the origin, σ_e represents the space constant of excitatory region and σ_i represents that of inhibitory region. This filter corresponds to an on-center and off-surround receptive field found in mammal's retina [5]. The ratio of space constants σ_i / σ_e is set to 1.6 for the filter. This ratio gives a close approximation to the idea Laplacian operator [6].

3.1 Induced Stimuli Extraction System (ISES)

The ISES extracts the induced stimuli needed for the perception of surfaces from subjective contours, except for the inducers, which in turn hide parts of the pattern.

Simple Visual Feature Extraction (layer U_2) Simple cells are excellent at detecting the presence of *simple* visual features, such as lines and edges of *a particular orientation*. In consideration of the orientation selectivity property of simple cells, a set of asymmetrical two-dimensional DOG filters for *eight preferred orientations* is used. The filter with vertical preferred orientation is defined by

$$d_2(x,y) = \left(e^{-x^2/2\sigma_e^2} - \frac{\sigma_e}{\sigma_i} e^{-x^2/2\sigma_i^2} \right) e^{-y^2/2\sigma_{en}^2} \tag{2}$$

This filter corresponds to a simple cell receptive field found in mammal's visual system[4].

Feature Restoration (layer U_3) As simple cells, complex cells respond to stimuli such as lines and edges of a particular orientation. However, the exact location of the stimulus is of no concern to a complex cell, as long as it is within that cell's receptive field. A set of two-dimensional Gaussian filters for eight orientations is used. The filter with vertical preferred orientation is defined by

$$d_3(x,y) = e^{-x^2/2\sigma_x^2} \cdot e^{-y^2/2\sigma_{en}^2} \tag{3}$$

This filter corresponds to a complex cell receptive field found in mammal's visual system[4].

Induced Stimuli Extraction (layer U_4) The types of visual features that appear to elicit response from hypercomplex cells are light-dark stimuli containing corners, curves and broken lines. The hypothetical arrangement of complex cells can implement an *end-stopped* hypercomplex receptive field. As a combination of several complex cells, the filter is defined by

$$d_4(x,y) = \prod_{\phi=0}^{7} d_3(x,y,\phi) \tag{4}$$

Image Restoration (layer U_5, U_6) The filter of layer U_5 recovers the weaken or reduced stimuli. A set of two dimentional Gaussian filters is used. For removing unnecessory noises, the system performed image operation with the input image. The output $u_6(x,y)$ of layer U_6, can be expressed as following logical expression;

$$u_6(x,y) = [u_5(x,y) \wedge u_1(x,y)] \tag{5}$$

where $\wedge$ represents logical AND operation.

3.2 Subjective Surfaces Perception System (SSPS)

The SSPS forms the subjective surfaces, which must always be seen to be above the induced stimuli.

Response Extraction Between the Induced Stimuli (layer U_7) The filter is detecting responses between the induced stimuli of the ISES. A set of asymmetrical two-dimensional three Gaussian filters for *eight preferred orientations* is used. The filter with vertical preferred orientation is defined by

$$g(\sigma,x) = e^{-x^2/2\sigma^2} \tag{6}$$

$$d_7(x,y) = \left(g(\sigma_e, x-l) - g(\sigma_i, x) + g(\sigma_e, x+l)\right) \cdot g(\sigma_{en}, y) \tag{7}$$

This filter corresponds to a simple cell receptive field.

Response Restoration (layer U_8) A set of two-dimensional Gaussian filters for eight orientations is used. The filter for a complex cell receptive field with vertical preferred orientation is defined by equation (3).

Subjective Surface Extraction (layer U_9) The process forms subjective surface from the extracted stimuli using image operation and feedback process. The output $u_9(x,y)$ of layer U_9, can be expressed as following;

$$d_9(x,y) = \prod_{\alpha=0}^{7} 1[\iint_{A_9} d_3(\xi,\eta,\alpha)\cdot u_8(x+\xi, y+\eta,\alpha)d\xi d\eta - \theta_9] \tag{8}$$

$$u_9(x,y) = [d_9(x,y) \vee u_6(x,y)] \tag{9}$$

where $\vee$ represents logical OR operation.

The output $u_9(x,y)$ of layer U_9 feedbacks to input image of SSPS(layer U_7) until removing the gap between the induced stimuli and forming surface.

Smoothing Process (layer U_{10}) This is a smoothing process for image improvement. A set of two-dimensional Gaussian filter is used.

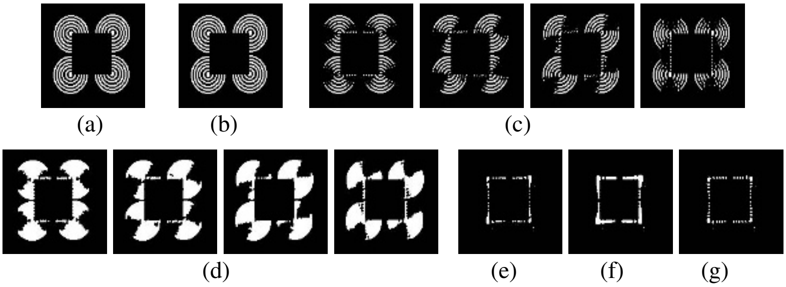


Fig. 2. Induced stimuli image of ISES. (a) binary image (b) image after contrast extraction by DOG filter; (c) some example images after simple visual feature extraction; (d) some example images after feature restoration; (e) image after induced stimuli extraction; (f) image after blurring process; (g) output image

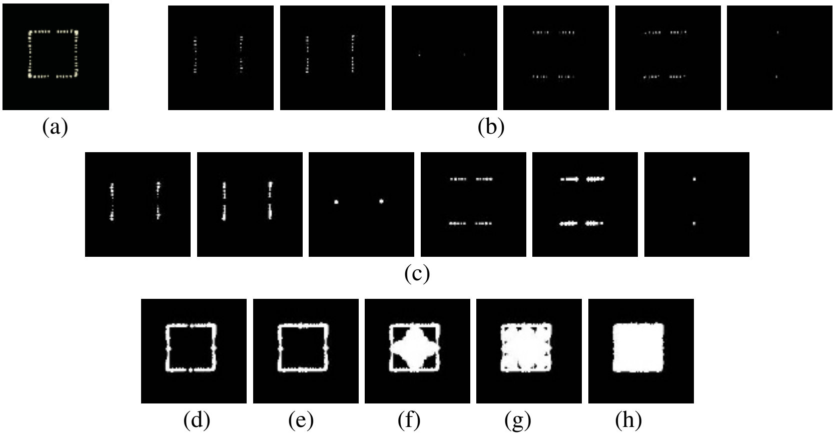


Fig. 3. Square surface perception of SSPS (a) input image (b) images after each response extraction between the induced stimuli; (c) images after each response restoration; (d) image after first iteration for surface extraction; (e) image after second iteration; (f) image after third iteration; (g) image after fourth iteration; (h) output image

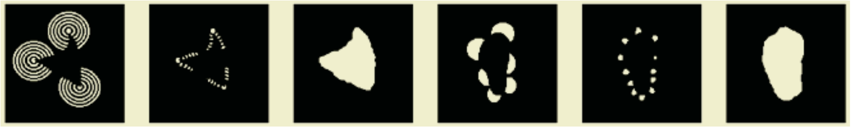


Fig. 4. The Result Images

4 Experimental Results

In order to show the performance of the model, experiments have been carried out using various Kanizsa-type subjective contour figures. Examples of the experiments are shown in Fig.2, Fig.3, and Fig.4. These results show that the performance of the model is sufficiently general.

5 Conclusion

A novel approach for the perception of occluded surface from Kanizsa-type subjective contour figures is proposed using the mechanism of feature extraction in visual system. Our model consists of two-stage coherent processing systems with multiple layers: the ISES and the SSPS. By the processing steps of each system, The former system extracts the induced stimuli for the perception of subjective surfaces, and the latter forms the subjective surfaces from the induced stimuli. The proposed model is demonstrated on a variety of Kanizsa-type subjective contour displays. The results of the experiment showed that the proposed neural network model was successful not only in extracting the induced stimuli for the perception of subjective contours, but also in perceiving the subjective surface from the induced stimuli.

References

1. Shumann, F.: Einige Beobachtungen Uber Die Zusammenfassung Von Gesichtseindrucken Zu Einheiten. Physiologische Studien, **1** (1904) 1-32
2. Kanizsa, G.: Organization in Vision. Praeger, New York (1979)
3. Franco Purghe, Staniey Coren.: Amodal Completion, Depth Straitification, and Illusory Figures: A Test of Kanizsa's Explanation. Perception, **21** (1995) 325-335
4. Marr, Vision, D.: A Computational Investigation into the Human Representation and Processing of Visual Information. W. H. Freedom & Company (1982)
5. Hubel, D. H., Wiesel, T. N.: Receptive Fields Binocular Interaction and Functional Architecture in the Cat's Visual Cortex. J Physiology, (1962) 106-154
6. Marr, D. and Hildreth, E.: A Theory of Edge Detection. Proc. R. Soc. Lond. B207 (1980) 187-217
7. Livingstone, M., Hubel, D.: Segregation of Form, Color, Movement, and Depth: Anatomy, Physiology and Perception. AAAS (Science), **240** (1988) 740-749
8. Ullman, S.: Filling in the Gaps: The Shape of Subjective Contours and a Model for Their Generation. Biological Cybernetics, **25** (1976) 1-6
9. Heitger, F., von der Heydt, R.: A Computational Model of Neural Contour Processing: Figure-Ground Segregation and Illusory Contours. Proceedings of the IEEE (1993) 32-40

Hopfield Neural Network Image Matching Based on Hausdorff Distance and Chaos Optimizing*

Zhenghao Shi[1,2], Yaning Feng[1], Linhua Zhang[3], and Shitan Huang[2]

[1] School of Computer Science, Xi’an University of Technology
Xi’an, Shaanxi 710048, China
[2] Xi’an Institute of Micro-electronics, Xi’an, Shaanxi 710054, China
[3] School of Information Science, Chongqing University, Chongqing 400044, China

Abstract. Due to its capability of high-speed information processing and uncertainty information processing, Feature point based Hopfield Neural Network image matching method has attracted considerable attention in recent years. However, there often exists much difference between two images, especially under the influences of distortion factors, thus the result of image matching is affected greatly. In addition, Hopfield Neural Network is often trapped in local minima, which gives an optimization solution with an unacceptable high cost. To overcome the defects mentioned above, in this paper, Hausdorff distance is used to measure the degree of the similarity of two images. Chaos is used to optimize the search process of Hopfield Neural Network, and a new energy formulation for general invariant matching is derived. Experimental results demonstrate the efficiency and the effectiveness of the proposed method.

1 Introduction

Image matching approaches based on neural networks have seen an explosion of interest over the last few years [1]. Many methods have been proposed [2],[3]. However, all these works aimed at finding the corresponding matching points. When amount of the feature points exceeds a certain extent or dimension of parameter transforming space is too high, computing time of these methods will be increased in a germination way. In addition, there often exists much difference between two images, such as object translating, rotation, scale variety, lighting variety and scene change, extracting points can’t be one to one correspondence. So there is some difficult applying existed image matching methods mentioned above in general situations.

To overcome drawbacks mentioned above and to avoid Hopfield Neural Network trapping into a local minimum, Hausdorff distance [4] is used to measure the degree of the similarity of two images., and chaos theory[5] is used to optimize the computing process of Hopfield Neural Network, and a new energy formulation for general invariant matching is derived. Experimental results for real images demonstrate the efficiency and the effectiveness of the proposed method.

* Supporting fund: Spaceflight Advanced Research fund (413160203).

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 848–853, 2005.

2 Measuring the Similarity of Images by Hausdorff Distance

Hausdorff distance (HD) is a metric between two point sets. Given two point sets $A = \{a_1, a_2, ..., a_p\}$ and $B = \{b_1, b_2, ..., b_q\}$, the Hausdorff distance between A and B is defined as [4]

$$H(A,B) = \max(h(A,B), h(B,A)), \tag{1}$$

$$h(A,B) = \max_{a \in A} \min_{b \in B} \| a - b \|, \tag{2}$$

$$h(B,A) = \max_{b \in B} \min_{a \in A} \| b - a \|. \tag{3}$$

Where h (A, B) and h (B, A) represent the directed distance between two sets A and B. $\|\cdot\|$ denotes some norm of points of A and B. The function h(A,B) identifies the point A which is the farthest one from any point B and measures the distance from *a* to its nearest neighbor in B. The Hausdorff distance H (A, B) measures the degree of mismatch between two point sets A and B.

This measure is asymmetric, since it dose not consider how well each of the points in B is fit by A. Matching can thus be performed against a large image that contains the real time image as a subset..

Assuming size of the real time image feature set R is M, and the one of the reference image feature set B is N, $M < N$, H(R, B) denotes the Hausdorff distance between the real time image feature set R and the reference image feature set B. Obviously, if the real time image corresponds to a small part of the reference image, then the Hausdorff distance between the real time image feature set and its corresponding sub-image feature set of the reference image should be unique. This best matching relationship can be depicted with a M×N matrix { $V_{i,j}$ }. Where $V_{i,j}$ ranges in [0,1], and the row serial number i is corresponding to serial number of the real time image feature point, the column serial number j is corresponding to serial number of the reference image feature point. If distance between the reference image feature *i and* the real time image feature *j is equal to the Hausdorff distance between* the real time image feature set and its corresponding sub-image feature set of the reference image, then $V_{i,j}$ =1,else $V_{i,j}$ =0. Obviously, the best matching relationship matrix { $V_{i,j}$ } should meet the following conditions:

$$\sum_{j=1}^{N} V_{ij} = 1, i = 1,2,...,M . \tag{4}$$

Viz. number of "1" in each row of the matrix {$V_{i,j}$ } should be only one, this denotes that to any point of the real time image feature point set, there is not more than one point in the reference image feature point set whose distance to the real time image feature point set is equal to the Hausdorff distance between the two corresponding image feature points set.

$$\sum_{\mathrm{i}=1}^{\mathrm{M}} V_{ij} \neq 0, j = 1,2,...,N \,. \tag{5}$$

This shows that there is not less than one element is “1” in each column of the matrix $\{V_{i,j}\}$. It denotes that to any point of the reference image feature point set, if it is an isomorphic part of the real time image, then there is not less than one point in the real time image feature point set whose distance to the reference image feature point set is equal to the Hausdorff distance between the two corresponding image feature points set.

$$\sum_{i=1}^{M}\sum_{j=1}^{N} V_{\mathrm{ij}} \neq 0, i = 1,2,...,M, j = 1,2,...,N \,. \tag{6}$$

This shows that there is not less than one in the reference image feature point set whose distance to the real time image feature point set is equal to the Hausdorff distance between the two corresponding image feature points set.

When making the design model mentioned above correspond to Hopfield neural network, each element of matrix $\{V_{i,j}\}$ was thought as a neuron, state of a neuron shows the matching degree. Interconnections among these neurons are mapped from Coefficients of row restriction, column restriction and element sum restriction of the design model. Assuming that x and y respectively denotes any point of the real time image feature point set, i and j respectively denotes any point of the reference image feature point set. For there is not less than one point in the real time image feature point set whose distance to the reference image feature point set is equal to the Hausdorff distance between the two corresponding image feature points set, so there appears more than one “1” in the output matrix of the neural network, viz. there is no column restriction in energy function of the neural networks. Based on analysis mentioned above, Hopfield neural network energy function corresponding to image matching based on Hausdorff distance is defined as

$$\begin{aligned} E = \frac{\alpha_1}{2}\sum_x(\sum_i V_{xi} - 1)^2 + \frac{\alpha_2}{2}(\sum_x\sum_i V_{xi})^2 \\ + \frac{\alpha_3}{2}\sum_x\sum_i | dis(x,\ \mathrm{i}) - H(R,B)\, | V_{xi} \,. \end{aligned} \tag{7}$$

Where $\frac{\alpha_1}{2}, \frac{\alpha_2}{2}, \frac{\alpha_3}{2}$ respectively denotes a weight coefficient of each restriction; V_{xi} denotes the matching degree that point x in the real time image feature point set is matched with point i in the reference image feature point set, which is corresponding to output state of a neuron; $d\mathrm{is}(x,\ \mathrm{i})$ denotes distance between point x of the real time image feature set and point i of the reference image feature set; $H(R,B)$ denotes the Hausdorff distance between the real time image feature point set and the reference image feature point set.

3 Optimizing Hopfield Neural Networks Image Matching by Chaos

Chaos is a motion state with randomicity directly produced by a determinacy equation, which can non-repeatedly go through all states within some range, so it is widely used to optimizing an algorithmic search process to avoid it trapping at a local minimum· Aihara and Chen etc. introduced chaos ideas into Hopfield neural networks model and presented the following Chaotic Neural Network (CNN) model [5]:

$$V_i(t) = \frac{1}{1+e^{-\frac{U_i(t)}{\varepsilon}}}, \tag{8}$$

$$\begin{aligned} U_i(t+1) = kU_i(t) + \alpha\left[\sum_{j=1, j\neq i}^{n} W_{ij}V_j(t) + I_i\right] \\ + Z(t)g[U_i(t) - U_i(t-1), i = 1,2,...,n], \end{aligned} \tag{9}$$

$$Z(t+1) = (1-\beta)Z(t). \tag{10}$$

Where Z(t) is a bifurcate parameter, which controls the intensity of the self feedback term; β (0 ≤ β ≤ 1) is a attenuation parameter, which dependents upon Z(t).

The system state varies step by step with the decrease of Z(t). In the initial search stage, if Z(t) is enough larger, the neural network system will be indicative of chaotic characteristic, and fully use the abundant dynamic characteristic of chaos to implement polymorphism ergod. For polymorphism ergod done according to the chaotic track and not limited to an object function, so it is very robust in avoiding the neural network system to trap in a local minimum.

When Z(t) is too small to act to evolvement of the system, the whole system will degrade into a single Hopfield neural networks, then the system will do more search according to the Hopfield neural networks gradient descent search mechanism in a smaller rage obtained by the pilot study, and convergence to a global satisfying solution soon.

It is obvious that introduction of Z(t) makes the neural network system's chaotic search hold randomicity and ergodic concurrently, which ensures the algorithm not only can escape from a local minimum, but also has high search efficiency.

Based on analysis above, image matching algorithm proposed in this paper adopted this Hopfield neural network model.

We take the derivative of the (7) with respect to V_{xi}

$$\frac{\partial E}{\partial V_{\mathrm{xi}}} = \alpha_1(\sum_{k=1}^{N} V_{xi} - 1) + \alpha_2 \sum_{j=1}^{M} V_{xi} + \frac{\alpha_3}{2}\sum_{x}\sum_{i} | dis(x,\ \mathrm{i}) - H(R,B)|. \tag{11}$$

from (9) and (11), the following equation of motion can be obtained:

$$
\begin{aligned}
U_{\mathrm{ik}}(t+1) = kU_{ik}(t) + \alpha(\alpha_1(\sum_{k=1}^{N} V_{xi} - 1) + \alpha_2 \sum_{j=1}^{M} V_{xi} \\
+ \frac{\alpha_3}{2} \sum_{x} \sum_{i} \left| dis(x,\ \mathrm{i}) - H(R,B) \right|) + Z(t) \sum_{i=1, i \neq j}^{n} W_{xi} V_{xj}(t) + I_0 .
\end{aligned} \tag{12}
$$

This just is the motion equation of Hopfield neural network image matching algorithm based on Hausdorff distance and chaos.

4 Experiment Results

Experimental images are shown as fig.1, size of the reference image (showed as fig. 1) is 120×120, and that of the real time image is 40×40.

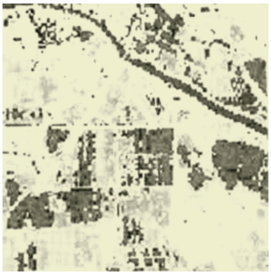


Fig. 1. The experimental image

Experiment 1: The real time image is directly randomly selected from the reference image.

Experiment 2: Adding Gauss noise into the reference image first, then rotating it with angle 3°.and then based on this, randomly cut out a 40×40 area from it as a real time image.

Interesting Points is taken as image matching feature points. Under the same condition, image matching to the two sets images is respectively done by IPBHNNIM (Interesting Point Based Hopfield Neural Network Image Matching Algorithm) and HDTCNN (Hopfield Neural Network Image Matching Algorithm based on Hausdorff Distance and Chaos). The experimental results are shown as Figure 2 and Table1.

From Fig. 2 we can see that the convergence speed of the conventional HNN output neuron is slower, and yet the one of HDTCN is observably faster, output neurons of HDTCNN soon switched to the gradient descent search process of HNN after experiencing a double period inverse biforked continuous chaotic divaricating process. Statistical results in Table 1 also authenticated this.

From Table 1 we also can see that in the case of that there is much noise in matching images, the matching probability of HNN is 92%, yet the one of HDTCNN is 96%. Experimental results mentioned above show that HDTCNN image matching algorithm proposed in this paper is more robustness than the conventional one.

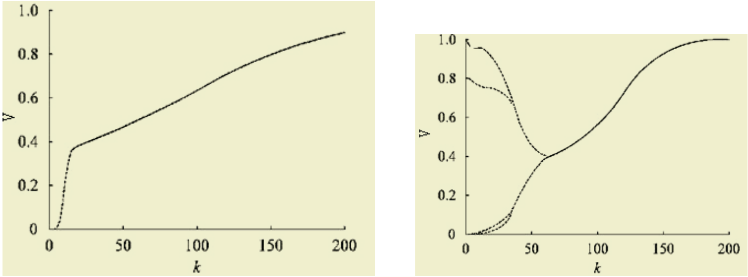


Fig. 2. Output curve of HNN and HDTCNN output neurons

Table 1. Comparison of matching probability and match time

	IPBHNNIM		HDTCNN	
	Match probability	Average time-costing (sec)	Match probability	Average time-costing (sec)
Experiment 1	100%	200	100%	150
Experiment 2	92%	200	96%	150

5 Conclusions

We employed Hausdorff distance to measure the degree of the similarity between two images, and used Chaos to optimize the search process of Hopfield Neural Network, derived a new energy formulation for Hopfield Neural Network based image matching method, where the problem of image matching is treated with the minimization of an energy function. Experimental results for real images also demonstrate the efficiency and the effectiveness of the proposed method.

References

1. Shi, Z., Huang, S., Feng, Y.: Artificial Neural Network Image Matching. Microelectronics and Computer, **20** (2003) 18-21
2. Nasrabadi, N.M., Li, W.: Object Recognition by a Hopfield Neural Network. IEEE Trans. Systems, Man, and, Cybernetics, **21** (1991) 1523-1535
3. Li, W., Lee, T.: Hopfield Neural Networks for Affine Invariant Matching. IEEE Transactions on Neural Networks, **12** (2001) 1400-1410
4. Huttenlocher, D.P., Kalanderman, G.A.: Comparing Image Using the Hausdorff distance. IEEE Trans. Pat. Anal. Machine Intel, **15** (1993) 850-863
5. Chen, L., Aihara, K.: Chaotic Simulated Annealing by a Neural Network Nodal with Transient Chaos. Neural Networks, **8** (1995) 915-930

Neural Network Based Fairing of Digitized Curves and Its Application

Jianhua Hou, Zongying Ou, and Mingen Guo

Key Laboratory for Precision and Non-traditional Machining Technology of Ministry
of Education, Dalian University of Technology, Dalian, Liaoning 116024, China
{uniweb,mingenguo}@126.com, ouzyg@dlut.edu.cn

Abstract. A fairing method for digitized curves, based on neural network optimization, is presented in this paper. The energy function and constraints of a digitized curve are given first, with which the fairing process is converted into a quadratic optimization problem. A neural network is then used to solve the optimization problem. Fairness adjustment of points' position is realized by this optimization. An application to binary image magnification is illustrated, which shows better result than other methods.

1 Introduction

Curve fairing is an important problem in CAD/CAM, also it's of high application value in other areas. For fairing of digitized curves, many algorithms have been proposed. Optimization method is an important class among these methods [1].

Aimed at the fairing process of digitized curves, a neural network based optimization method is presented in this paper. For a given digitized curve, the energy function is constructed directly from the points sequence on the curve, and then it's expressed as a quadratic function. Considering the actual requirements in digitized curve fairing, constraints of points' position in fairing process are proposed. So the fairing process of a digitized curve is converted into a quadratic optimization problem with bound constraints.

For solving this optimization problem, neural network is used. The neural network optimization method is amenable to parallel implementation, unlike the more conventional approaches to the bound-constrained quadratic optimization summarized by More and Toraldo [2], which are time consuming. Fairness adjustment of points' position is realized by this optimization.

The paper is organized as follows. In Section 2, energy of curve and constraints in fairing process are introduced. In Section3, an optimization method based on neural network is described. In Section 4, an application to binary image magnification is illustrated, examples are also given in this section. The conclusion in Section 5 reviews the method of this paper.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 854–859, 2005.

2 Energy of Curve and Constraints in Fairing Process

For a digitized curve, heavy quantized noises exist in curve data, which is a main factor affects the fairness of the curve. The purpose of curve fairing process is just to eliminate these noises. Here, energy optimization is utilized to adjust the points' position on a digitized curve to perform the fairing.

2.1 Energy of Digitized Curve

Denote the points on a digitized curve sequentially as $\overrightarrow{P_i}$, $i = 0,1,\cdots,n$. Then, the energy function of the curve can be written as [1]:

$$E = \sum_{i=1}^{n-1} \frac{1}{l_i + l_{i+1}} \left\| e_{i+1} - e_i \right\|^2 . \tag{1}$$

where $e_i = (\overrightarrow{Pi} - \overrightarrow{P_{i-1}})/l_i$ is the unit vector of arc $\overrightarrow{P_{i-1}P_i}$, $l_i = \left\| \overrightarrow{P_i} - \overrightarrow{P_{i-1}} \right\|$ is the length of arc $\overrightarrow{P_{i-1}P_i}$ .In equation (1), $\left\| e_{i+1} - e_i \right\|$ represents the direction change from arc $\overrightarrow{P_{i-1}P_i}$ to arc $\overrightarrow{P_iP_{i+1}}$,and ($l_i + l_{i+1}$) is the sum of length of the two arcs. Energy E can be used to indicate the fairness of a curve. When E is large, the fairness of the curve is poor, and when E is small, the fairness of the curve is good. So, in fairing process, the principle of adjustment of the points' position is to minimize E.

For simplicity, let l_i be their initial value, and keep unchanged in the following optimization. As boundary conditions, let the end points fixed on their original positions, adjustment operation only performed to $\overrightarrow{P_i}$, $i = 1,2,\cdots,n-1$.

Denote the points in fairing process as $\overrightarrow{Q_i}$, $i = 1,2,\cdots,n-1$, initial position of $\overrightarrow{Q_i}$ is $\overrightarrow{P_i}$, $i = 1,2,\cdots,n-1$, the coordinate of point $\overrightarrow{Q_i}$ is (x_{2i-1} , x_{2i}), then the coordinates of all the points $\overrightarrow{Q_i}$ constitute the vector $x = (x_1, x_2, \cdots, x_{2n-3}, x_{2n-2})^T$. In equation (1), express e_i with coordinates of related points, then equation (1) can be rewritten as:

$$E = \sum_{i=1}^{n-1} [(p_i^T x - b_{ix})^2 + (q_i^T x - b_{iy})^2] . \tag{2}$$

The first part in equation (2) is the X-coordinate component, the second part is the Y-coordinate component. p_i and q_i are column vectors with the size of (2n-2). Most of the elements in p_i and q_i are zero, only few element are non-zero. b_{ix} and b_{iy} are non-zero only for $i = 1$ and n-1.

Denote $D = (p_1, q_1, \cdots, p_{n-1}, q_{n-1})^T$, $b = (b_{1x}, b_{1y}, \cdots, b_{(n-1)x}, b_{(n-1)y})^T$. Then, we have:

$$E(x) = (Dx - b)^T (Dx - b) = x^T Hx - 2d^T x + g \tag{3}$$

where $H = D^T D$, $d = D^T b$, $g = b^T b$ .Matrix H is symmetric and positive semidefinite , the size of H is (2n-2)×(2n-2).

2.2 Constraints in Fairing Process

In curve fairing process, the range of points' movement should be constrained. Here, the position of adjusted point $\overrightarrow{Q_i}$ is limited to be in the unit square centered at $\overrightarrow{P_i}$ (within 1 pixel). Denote the coordinate of point $\overrightarrow{P_i}$ as (P_{ix}, P_{iy}) , then we have:

$$P_{ix} + 0.5 \ge x_{2i-1} \ge P_{ix} - 0.5 \, , P_{iy} + 0.5 \ge x_{2i} \ge P_{iy} - 0.5 \, , i = 1, \cdots, n-1 \tag{4}$$

Let $ub = (P_{1x} + 0.5, P_{1y} + 0.5, \cdots, P_{(n-1)x} + 0.5, P_{(n-1)y} + 0.5)^T$, then $ub$is the upper bound of vector x ; let $lb = (P_{1x} - 0.5, P_{1y} - 0.5, \cdots, P_{(n-1)x} - 0.5, P_{(n-1)y} - 0.5)^T$, then lb is the lower bound of vector x . The inequalities (4) can be expressed as:

$$ub \ge x \ge lb \tag{5}$$

Considering both (3) and (5), the adjustment of points' position can be expressed as the optimization problem below:

$$\min\{E = x^T Hx - 2d^T x + g : ub \ge x \ge lb\} \tag{6}$$

By (6), curve fairing process is converted into a quadratic optimization problem.

3 Neural Network Optimization

In order to ensure that the constraints are always satisfied , let:

$$x = h(u) = Bf(u) \tag{7}$$

where $f(u)$ is defined as follows:

$$f_i(u) = f_i(u_i) = \begin{cases} s_i, u_i < s_i \\ u_i, u_i \in [s_i, t_i] \\ t_i, u_i > t_i \end{cases} .$$

u_i is permitted to vary without constraint, and B is a positive diagonal matrix which serves as a preconditioner, $s = (s_1, \cdots, s_{n-1})^T$ and $t = (t_1, \cdots, t_{n-1})^T$ are the constraints

ub and lb on x mapped onto corresponding values of u , $s = B^{-1}lb$ and $t = B^{-1}ub$ .A typical element $f_i(u_i)$ is illustrated in Fig.1. Replacing x with $h(u)$, the energy function in (3) can be written as:

$$E(u) = h(u)^T Hh(u) - 2d^T h(u) + g \ . \tag{8}$$

Using (8), the bound-constrained optimization problem in (6) reduces to the unconstrained minimization of $E(u)$ over u.

For solving the optimization problem in (8), a single-layered recurrent neural network is used, the state vector u is governed by the following differential equation [3]:

$$\frac{du}{dt} = d - Cf(u) - Au \ . \tag{9}$$

where $f(u)$ is the network output, C is the lateral feedback matrix with zero diagonal elements, and A is a positive diagonal matrix representing the passive decay rate of the state vector. One node of the network is illustrated in Fig.2.

To map the constraint quadratic problem onto the neural network, set:

$$A = diag(HB) \, , C = HB - A \ . \tag{10}$$

where diag() selects the diagonal elements of its matrix argument. The desired output $x = Bf(u)$ is obtained from the network output.

Bouzerdoum [3] proved that the neural network in (9) has a unique equilibrium point u* which is mapped by h(u) on to x*, the unique constrained minimum of $E(x)$. Also he proved that the energy function $E(u)$ is a global Liapunov function for the neural network described by (9), and the network is globally convergent. Under appropriate conditions on matrices A and C, the neural network is exponentially asymptotically stable, which can converge exponentially to the optimal solution.

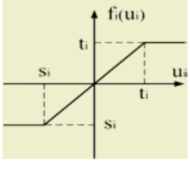


Fig. 1. Function $f_i(u_i)$

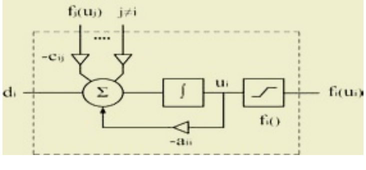


Fig. 2. Neural network node

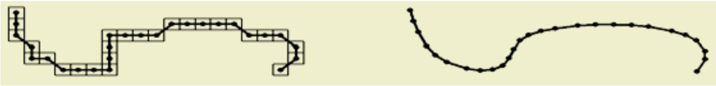
Data simulation result reveals that the matrix H is ill-conditioned. It's known that if a matrix has a large condition number, then considerable round-off errors accumulate when attempting to solve the problem. For this optimization problem, the error amplification factor is bounded by the condition numbers of matrices HB and A. The susceptibility of the system to errors can be considerably reduced by choosing the preconditioner B so that the condition number of HB is smaller than that of H. In [3], a simple method is given to set matrix B. Denote the ith diagonal element of B by β_i, and the ith diagonal element of H by q_{ii} ,then:

$$\beta_i = \kappa / q_{ii}\, , \kappa > 0\, . \tag{11}$$

where κ is a factor which can affect the speed of convergence of the neural network, we can simply set $\kappa = 1$.

4 Application to Image Magnification

Based on the optimization process mentioned above, a fairness curve can be obtained with adjusted points. Fig.3 shows an example. Fig.3 (a) is the original digitized curve, squares in this figure constitute the constraint band of the algorithm. Fig.3 (b) shows the result after fairing process using the method in this paper. Clearly the fairness of the curve has been improved significantly.



(a) (b)

Fig. 3. An example of digitized curve fairing

The following example introduces the application of this method in binary image magnification.

The process of image magnification is to interpolate new pixels from the original image. Several methods, including nearest neighbor interpolation, bi-linear interpolation etc., are commonly used. In the case of nearest neighbor interpolation, the magnified image contains mosaic. When bi-linear interpolation is used, the smoothing effect may degrade the high frequency component, and edges become blurred [4].

Here, an edge-directed algorithm is presented based on the method in this paper. First, edges in the image are abstracted. Assume the background is of value 0 in the binary image, then the edge points can be easily determined by the following principle: the pixel is of value 1,and at least one neighbor point is of value 0. After edge detection, the method in this paper is used to fair the digitized edge curves. For each edge curve after fairing process, a 3-order B-spline interpolation is executed to get a continuous curve passing through the adjusted points [5]. By these steps, fairness and continuous edge curves are obtained from these digitized edge points. This is a process of edge curve restoration. When a binary image is to be magnified, a re-digitize operation is performed based on the continuous edge curves obtained above, and so the magnified image can be gotten.

Fig. 4 shows an example of binary image magnification.Fig.4 (a) is the original image, which is a Chinese character ‘ren’; Fig.4 (b), (c) and (d) are magnified images, the magnification factor is 4. Fig.4 (b) shows the result using nearest neighbor algorithm, edge of the result image is zigzag; Fig.4 (c) shows the result using bi-linear interpolation algorithm, edge of the result image is blurred; Fig.4 (d) shows the result using the method in this paper, edge of the result image keep still sharp and smooth.

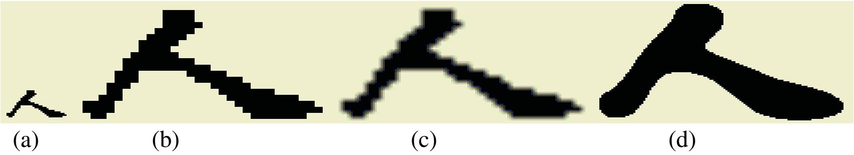


Fig. 4. Examples of binary image magnification

From this example, it can be seen that using the method in this paper, the result is better than that of other algorithms. In fact, because the fairness and continuous edge curves have been restored, so no matter how many times to magnify the binary images, high quality results can always be gotten by using this method.

5 Conclusion

A digitized curve fairing method with neural network optimization is presented in this paper. By using neural network, this method is amenable to parallel implementation, unlike the conventional approaches. One application example is given in binary image magnification, which shows better result compared with other algorithms. Also, the method described in this paper can be widely used in other areas.

References

1. Zhu, X., et al.: Modeling Technology of Free Form Curves and Surfaces. Beijing: Science Press 2000 (in Chinese)
2. More, I.J., Toraldo, G.: On the solution of Large Quadratic Programming Problems with Bound Constraints. SIAM J. Optimization, **1** (1991) 93-113
3. Bouzerdoum, A.: Neural Network for Quadratic Optimization with Bound Constraints. IEEE Trans. On Neural Networks, **4** (1993) 293-303
4. Castleman, K.R.: Digital Image Processing. Englewood Cliffs, NJ: Prentice-Hall International Inc (1996)
5. Shi, F.: Computer Aided Geometric Design and Non-Uniform Rational B-spline. Beijing, Beijing University of Aeronautics & Astronautics Press (1994)

A Digital Image Encryption Scheme Based on the Hybrid of Cellular Neural Network and Logistic Map

Wei Zhang[1,2], Jun Peng[3], Huaqian Yang[1,2], and Pengcheng Wei[1]

[1] Department of Computer and Modern Education Technology
Chongqing Education College, Chongqing 400067, China
[2] Department of Computer Science and Engineering
Chongqing University, Chongqing 400044, China
[3] Department of Electronic Information Engineering
Chongqing University of Science and Technology, Chongqing 400050, China

Abstract. In this paper, a digital image encryption scheme based on the hybrid chaotic system is proposed. A three-order Cellular Neural Network (CNN) with complex dynamical behavior and Logistic chaotic map are employed in the encryption scheme. The output sequences of two chaotic systems are combined effectively by using a special method, and a fast image encryption is realized by means of the bit-wise XOR operations. The results of the security analyses indicate that this encryption scheme not only has a large key space but also has a very sensitivity with respect to the encryption key. The encryption scheme proposed in this paper possesses perfect diffusion and confusion properties and it can resist the know-plaintext attacks and chosen-plaintext attacks.

1 Introduction

Along with the rapid development of Internet and universal application of multimedia technology, lots of valuable image information need be transmitted with security. Although conventional cryptography algorithms such as DES can be applied to image cryptography, they are easily attacked because of short encryption key, and public encryption algorithms can provide long encryption key, they aren't adapt to bulky image encryption due to their slow encryption speed [9-13]. Hence, it is necessary to research a new image encryption approach with strong security.

The researches have found that Cellular Neural Network (CNN) has a complex Dynamic behavior [1],[2],[3]. It is extensively applied in fields of pattern recognition, image processing and the solving partial differential equation. CNN is easily realized by using VLSI circuit because of its regular structure and each cell only coupled with adjacent cells. As we all know, chaotic system is sensitive to initial values and system parameters; it is suitable for information encryption due to good features such as broad band, noise-like, accurate regeneration and difficult prediction. In this paper, by combing a three-order CNN chaotic network with Logistic chaotic map, a fast and safely image encryption system is designed and analyzed.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 860–867, 2005.

2 Image Encryption Scheme

2.1 Chaotic System Selection

Chua and Yang first proposed the Cellular Neural Network (CNN) in 1988[1]. Here, we select a three-order CNN dynamic model with full connection:

$$\frac{d_{x_j}}{dt} = -x_j + a_j y_j + \sum_{k=1,k\neq j}^{3} a_{jk} y_j + \sum_{k=1}^{3} s_{jk} x_k + i_j\,,\ j = 1,2,3\ . \tag{1}$$

where x_j is a state variable, y_j is a cell output. y_j is related to x_j and defined as

$$y_j = 0.5(|x_j + 1|) - |x_j - 1|\,,\ j = 1,2,3\ . \tag{2}$$

Let three cell-template parameters are [4]:

$$a_{12} = a_{a3} = a_2 = a_{23} = a_{32} = a_3 = a_{21} = a_{31} = 0\,;$$

$$S_{13} = S_{31} = S_{22} = 0\,; i_1 = i_2 = i_3 = 0\,; S_{33} = 1\,; S_{21} = S_{23} = 1\,;$$

Then, model (1) is abbreviated as

$$\begin{aligned} \frac{dx_1}{dt} &= -x_1 + a_1 y_1 + S_{11} x_1 + S_{12} x_2 \\ \frac{dx_2}{dt} &= -x_2 + x_1 + x_3 \qquad . \\ \frac{dx_3}{dt} &= -x_3 + S_{32} x_2 + x_3 \end{aligned} \tag{3}$$

In Fig. 1, we can obtain chaotic attractor with different property by assigning different values to parameters a_1, S_{11}, S_{12} and S_{32}. These chaotic attractors can be generated in a large scale, the complexity of output sequence in subsequent encryption system is enhanced, and the systematic security is evidently enhanced as well.

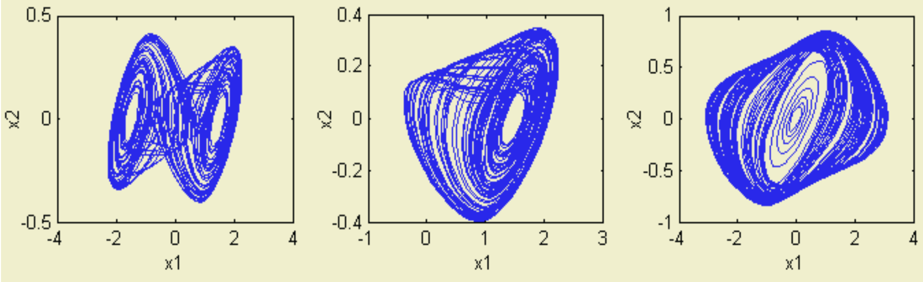


Fig. 1. Different chaotic attractors of 3-order CNN model. (a) a1 = 3.86, s11 = -1.55, s12 = 8.98, s32 = -14.25; (b) a1 = 3.85, s11 = -1.55, s12 = 8.76, s32 = -14.35; (c) a1 = -4.198, s11 = 2.365, s12 = 7.45, s32 = -10.98

In chaotic system, another classical Logistic chaotic map is

$$\theta_{i+1} = \mu(1-\theta_i)\,,\quad -1 < \theta_i < 1\ . \tag{4}$$

where μ is systematic parameter. Where $\mu = 4$, system is in a chaotic status and its orbits are travel and non-periodic , so we only discuss it in this paper.

2.2 Description of Image Encryption Scheme

Fig.2 shows an image encryption scheme based on the combination of two chaotic systems, chaotic system I is a three-order CNN, chaotic system II is a Logistic chaotic Map. Two systems produce chaotic sequences C_i^1 and C_i^2 at time i respectively. The rules generated by sequences are defined as:

$$C_i^1 = floor(K(x_{i,j}^2 + x_{2,j}^2 + x_{3,j}^2)^{1/2} + \lambda) \bmod 256 \ . \tag{5}$$

$$C_i^2 = floor((K\theta_i + \lambda)^2)\, mod\, 256 \ . \tag{6}$$

Here $x_{1,i}$, $x_{2,i}$ and $x_{3,i}$ in (5) are values of three states variable of CNN at time i , θ_i in (6) is the value of Logistic system at time i , K and λ are control parameters which are used to magnify the sensitive dependence on initial condition in chaotic system.

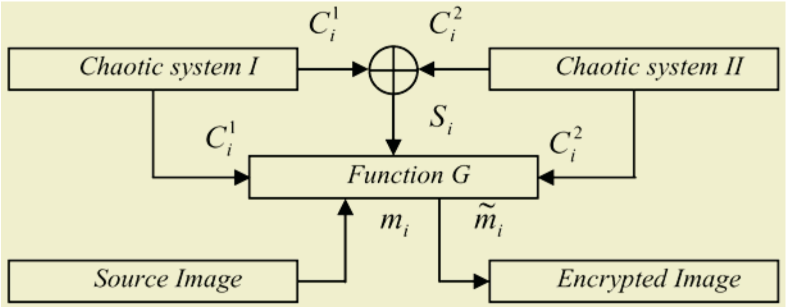


Fig. 2. Image encryption scheme block

In Fig. 2, $S_i = C_i^1 \oplus C_i^2$, m_i and $\tilde{m}_i$ denote the i pixel of original image and encrypted image respectively (numbering image pixels according to the sequences of TV scan). Function G is applied in encryption or decryption, it is described as follows:

1) Encryption m_i: $\tilde{m}_i = C_i^1 \oplus ((C_i^2 + m_i)\, mod\ 256)$, where S_i is even
$\tilde{m}_i = C_i^2 \oplus ((C_i^1 + m_i)\, mod\ 256)$ where S_i is odd

2) Decryption $\tilde{m}_i$: $m_i = C_i^1 \oplus ((\tilde{m}_i - C_i^2)\, mod\ 256)$, where S_i is even
$m_i = C_i^2 \oplus ((\tilde{m}_i - C_i^1)\, mod\ 256)$, where S_i is odd

The encryption key is expressed by a array with 10 elements, it is, Key $= (x_{1,0}, x_{2,0}, x_{3,0}, a_1, S_{11}, S_{12}, S_{32}, \theta_0, K, \lambda)$ Where $x_{1,0}$, $x_{2,0}$ and $x_{3,0}$ are the initial value of states variables and $a_1, S_{11}, S_{12}, S_{32}$ are system parameters of CNN model (3), θ_0 is the initial value of Logistic system, K and λ are control parameters. To improve the system initial sensitivity, K is not suitable for too small.

3 Security Analysis

3.1 Sensitivity to Key

Let Key_S=(0.20,0.20,0.20,3.86, -1.55,8.98, -14.25,0.3871,3000,127), experimental results of resolution 128×128 woman.bmp are shown in Fig. 3. The results indicate that this encryption scheme is sensitive to encryption key, different encrypted images are caused by slight variances, and correct decryption can be done by fully match with the original encryption key.

We know that the encryption key has 10 parameters. Let initial values of two chaotic systems are positive number which are less than 1, and their precision is 4, so the four system parameters of CNN model (3) are. The value of parameter K ranges between 1 and 106, the value of λ ranges from 0 to 255, then the space of encryption key is about 1038.9($\approx$2129), here, the entropy of encryption system is 129. Because of possessing enough large space of encryption key, the encryption system can resist brute attack effectively.

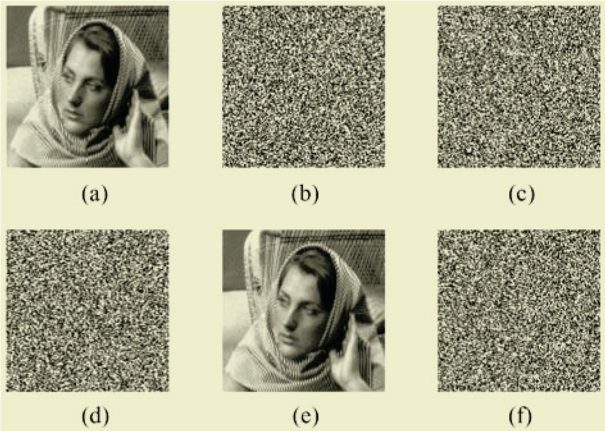


Fig. 3. Key sensitive test. (a)original image; (b),(c)and(d) Encrypted image using slight difference from the encryption key and key parameter is K =3001, λ =128, $x_{2,0}$ =20.001(the others parameter is the same to key_S); (e) Decrypted image using correct key ; (f) Decrypted image using slight difference from the encryption key

3.2 Diffusion Property

Diffusion means that redundancy of plain-text is dispersed in cipher-text to hide the statistical structure of plain-text. Statistical analysis has been performed on the proposed image encryption algorithm, demonstrating its superior confusion and diffusion properties which strongly resist statistical attacks. We examine the correlation between two vertically adjacent pixels, horizontally adjacent pixels, and two diagonally adjacent pixels, respectively. Relevant coefficient is defined as follow

$$r_{xy} = \mathrm{cov}(x, y) \Big/ \sqrt{D(x)D(y)} \quad . \tag{7}$$

Where is variance, is variance, x and y denote the gray value of image. In numerical computation, the discrete forms were used [5].

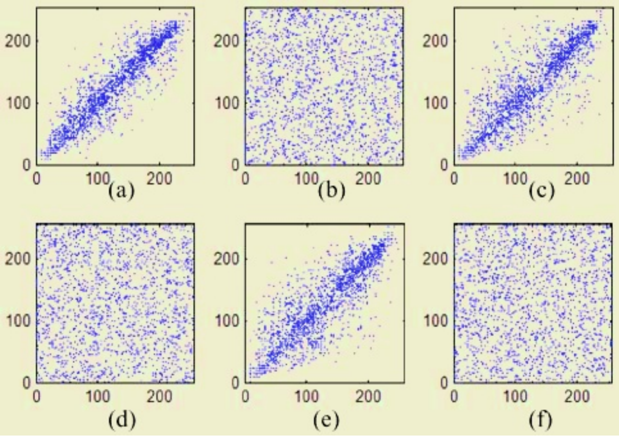


Fig. 4. Pixel gray value on location(x,y). (a),(c)and(e) indicate two vertically adjacent pixels gray value ,horizontally adjacent pixels gray value, and two diagonally adjacent pixels gray value on location(x,y) in plain-image respectively. (b),(d) and(f) indicate two vertically adjacent pixels gray value, horizontally adjacent pixels gray value, and two diagonally adjacent pixels gray value on location(x,y) in cipher-image respectively. Relevant coefficient of (a-f) are 9365, 0.0286, 0.8654, 0.0203, 0.8536, 0.0197

Firstly, randomly select 1000 pairs of pixels from an image, then calculate in term of formula (7),($N = 1000$). Signs r_{xy}^{h} , r_{xy}^{v} and r_{xy}^{d} denote relevant coefficient of between horizontal, vertical and diagonal adjacent pixels. In plain-image, r_{xy}^{h} is 0.9365, r_{xy}^{v} is 0.8654 and r_{xy}^{d} is 0.8536. In chipper-image, r_{xy}^{h} is 0.90286, r_{xy}^{v} is 0.0203 and r_{xy}^{d} is 0.0197. Because an plain-image has visual meaning, gray value of pixel varies slowly. Hence, relevant coefficient of between pixels is larger than cipher-image's in three adjacent cases. Whereas the cipher-image resembles noise, that is consistent with the law of statistics.

The distribution status of 1000 pair's adjacent pixels on (x,y)-(x+1,y) gray plane, which are chosen randomly, is shown in Fig.4, that is a help to understand the relevant coefficient in three adjacent cases. For example, (a) of Fig.4, random pairs in original image form a long and narrow strip on (x,y)-(x+1,y) gray plane, however, these pixel pairs in the cipher-image behave a random distributive status, which almost fill the whole (x,y)-(x+1,y) gray plane. From the point of view of cryptography, this kind of random contribution, which is one of characters a good encryption algorithm should possess, fully shows that the proposed encryption system possesses a favorable diffusion performance of plain-text.

3.3 Confusion Property

Confusion means that the relation of between encryption key and cipher-text is masked, the statistic relation between secret encryption and cipher-text become complexity as possible, and attacker can not educe the encryption key from cipher-text. The image Camerman.bmp of size 256×256 and a full black image (namely, value of all pixels is 0) are encrypted using the encryption key key_S mentioned in section 3.1. The histogram of plain-images and cipher-images are drawn in Fig. 5, respectively.

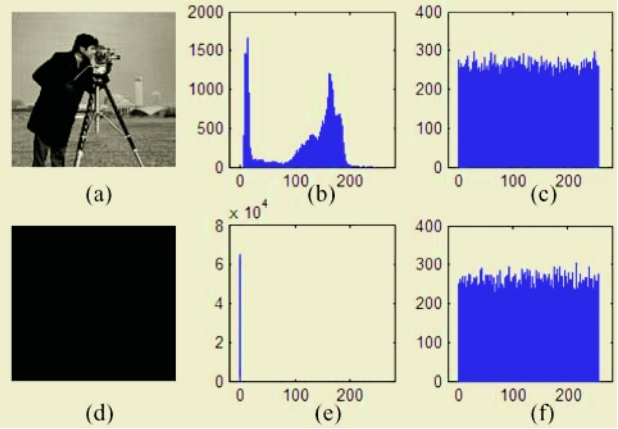


Fig. 5. The histogram of plain-images and cipher-image. (a)an image named Cameraman.bmp; (d) full black image; (b)and (e) The histogram of two plain-image (c)and(f) The corresponding histogram

We find that, although the histogram of two plain-images has their distributive characters respectively, the histogram of two ciphered images are fairly uniform. It is very difficult to educe secret key from cipher-text when attacker try to attack by using the known-plaintext attacks or chosen-plaintext attacks [18].

4 Conclusions

In this paper, some characters possessed by chaotic system are utilized[14], for example, sensitivity to initial value and systematic parameter, complexity and uncertainty of chaotic sequence and so forth. The new scheme employs a three-order full connection Cellular Neural Network and output sequences of Logistic map to realize image encryption. The results of the security analyses also indicate that the encryption scheme proposed in this paper not only has a large key space but also has a very sensitive to the encryption key, and it can resist the known-plaintext attacks and chosen-plaintext attacks[15].

Shannon proposed two basic principle of cryptogram design in his classic paper [6], namely diffusion and confusion. However, mixed property of orbits of chaos can correspond to the diffusion property of cryptography, random-like property and sensitivity to systematic parameter of chaotic signal can correspond to the confusion property of Cryptography [7]. The results of the sensitivity experiment and the statistic contrast analyses in section 3 indicate that system possess very diffusion and confusion by employing chaotic mechanism in encryption system.

It is a significant attempt to combine Cellular Neural Network and Logistic map in image encryption. It would be interesting to continually research the application based on Cellular Neural.

Acknowledgements

The work described in this paper was supported by a grant from the National Natural Science Foundation of China (No.60271019), the Doctorate Foundation of the Ministry of Education of China (No.20020611007), Chongqing Education Committee (No.011401), Chongqing Science and Technology Committee (No.2001-6878).

References

1. Chua, L.O., Yang, T.: Cellular Neural Network Theory. IEEE Trans. Circuits System, **35** (1988) 1257-1272
2. Chua, L.O, Roska, T.: The CNN Paradigm. IEEE Trans. CAS-I, 40 (1993) 47-156
3. Civalleri, P.P., Gilli, M.: On Dynamic Behavior of Two-cell Cellular Neural Networks. Int. J. Circ, **21** (1993) 451-471
4. He, Z.Y., Zhang, Y.F., Lu, H.T.: The Dynamic Character of Cellular Neural Network with Application to Secure Communication. Journal of China Institute of Communications, 20 (1999) 59-67
5. Chen, G.R., Mao, Y.B., C.K. Chui: A Symmetric Image Encryption Scheme Based on 3D Chaotic Cat Maps. Chaos, Solitons and Fractals, **21** (2004) 749-761
6. Shannon, C.E.: Communication Theory of Secrecy System. The Bell System Technical Journal, **28** (1949) 656-715
7. Kocarev, L.: Chaos-based Cryptography: A Brief Overview. IEEE Trans. on CAS-I, **1** (2001) 6-21
8. Diffie, W., Hellman, M.E.: New Directions in Cryptography. IEEE Trans. on Information Theory, **6** (1976) 644-654
9. Chang, C.C., Hwang, M.S., Chen, T.S.: A New Encryption Algorithm for Image Crypt-Systems. J System Software, **58** (2001) 83-91
10. Cheng, H., Li, X.B.: Partial Encryption of Compressed Images and Videos. IEEE Trans. Signal Processing, **48** (2000) 2439-2451

11. Fridrich, J.: Symmetric Ciphers Based on Two-dimensional Chaotic Maps. Int J. Bifurcat Chaos, **8** (1998) 1259-1284
12. Li, S.J., Zheng, X., Mou, X., Cai, Y.: Chaotic Encryption Scheme for Real-time Digital Video. Proc SPIE on Electronic Imaging, San Jose, CA, USA (2002) 4666-4667
13. Scharinger, J.: Fast Encryption of Image Data Using Chaotic Kolmogorov Flows. J Electron Imaging, **7** (1998) 318-325
14. Kocarev, L., Jakimovski G.: Chaos and Cryptography: From Chaotic Maps to Encryption Algorithms. IEEE Trans. Circuit System, **48** (2001) 163-169
15. Mao, Y.B., Chen, G.: Chaos-based Image Encryption. Handbook of Computational Geometry for Pattern Recognition, Computer Vision, Neurocomputing and Robotics. New York, Springer-Verlag, in press (2004)
16. Schneier, B.: Applied Cryptography Protocols, Algorithms, and Source Code in C. 2nd Ed. New York Wiley (1995)

Image Encryption Scheme Based on Chaotic Neural System

Shaojiang Deng, Linhua Zhang, and Di Xiao

Department of Computer Science and Engineering
Chongqing University, Chongqing 400044, China

Abstract. Encryption of image is different from that of texts due to some intrinsic features of images .The time-delay chaotic system can generate complicated dynamical behavior. This is because it has many positive Lapunov exponents. Hence, it is called as super-chaotic system. In this paper, Liao's chaotic neural system, cat map and general cat map are introduced and analyzed respectively. Then, a new image encryption scheme by employing the Liao's chaotic system to generate parameters of general cat map is designed. Experimental tests have also demonstrated the high security and fast encryption speed of the proposed scheme.

1 Introduction

Today, more and more information that include text, audio, image and other multimedia, has been transmitted over the internet. Images have been widely used in our daily life. So, the security of image is very important. Due to some intrinsic features of images, such as bulk data capacity and high correlation among pixels, traditional encryption algorithms are not suitable for practical image encryption. We know that the aim of traditional encryption algorithms is to shuffle the plain image ,it make ciphers look like random. The arrange operator is confirmed in advance and has not any relation with key. This is obvious bug, so some iterative product encryption is attacked easily by differential encryption analysis. But array algorithm based on key and regarding key as parameter that can uniquely make sure the property of array. Due to the exceptionally desirable properties of mixing and sensitivity to initial conditions and parameters of chaotic maps, so, chaos-based encryption has given a new and efficient way to deal with the intractable problem of fast and highly secure image encryption. Now there exist many image encryption methods such as: Arnold map, etc. There exist many image encryption algorithms based on chaotic system(such as Logistic, Chebeshev, Lorenz, etc.). It is well-known that low dimension chaotic system has only one positive Lyapunov exponent. Furthermore, the chaotic signal is decrypted easily. To improve the security property, the time-delay chaotic neural system with super-chaotic property and many positive Lapunov exponents is applied to image encryption. The time-delay chaotic neural system is so complicated that it is difficult and infeasible to attack.

According to the basic principle of cryptology [3], a cryptosystem should be sensitive to the key, i.e., the cipher-text should have close correlation with the key. There

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 868–872, 2005.

are two ways to accomplish this requirement: one is to mix the key thoroughly into the plain-text by means of the encryption process; another is to use a good (ideally, truly random) key generation mechanism. In [5], the authors first extended the cat map to “general cat map”, then the general cat map is used to encrypted image. In encryption algorithms of the general cat map, the keys(*a* and *b*) are get from keyboard directly. This paper proposes a new method that generate parameters(*a* and *b*) by using Liao’s chaotic neural system. The proposed method have better security than that of the general cat map.

2 Liao’s Chaotic Neural System

Liao’s chaotic neural system is:

$$\dot{x}(t) = -x(t) + af[x(t) - bx(t-\tau) + c], t > 0 \quad (1)$$

where *f*(*x*) is a nonlinear function. the time delay $\tau > 0$, *a* and *b* are constants. Suppose that the initial conditions of system (1) satisfy:

$$x(h) = x_{in}(h), h \in [-\tau, 0]$$

where $x_{in}(h)$ is continuous-value function on $[-\tau, 0]$.

Liao et al. [1] investigated the linear stability and local stability of bifurcating periodic solution in the neural network model (1), conditions of stability are determined. Chaotic behavior of the system with

$$f(x) = \sum_{i=1}^{2} \alpha_i [\arctan(x + k_i) - \arctan(x - k_i)],$$

where a_i and k_i are constants , and $a_1 = 2$ $k_1 = 1$ $a_2 = -1.5$ $k_2 = 4/3$, are also found. A series of dynamical behavior for system (1) occur by fixing *a* and increasing *b*, interested reader may refer to [1] (see Figure 1). In [1], the authors has obtained that the largest Lyapunov exponent is positive when *b*>1.868.

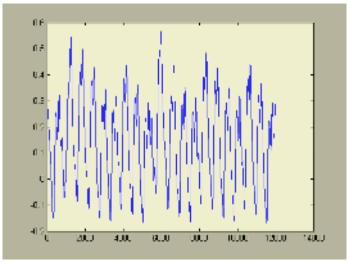


Fig. 1. *a*=3, *b*=8, waveform diagram for system (1)

3 Image Encryption

3.1 Cat Map and General Cat Map

Chaotic encryption algorithms have shown some exceptionally good properties in many concerned aspects regarding security, complexity, speed, computing power and

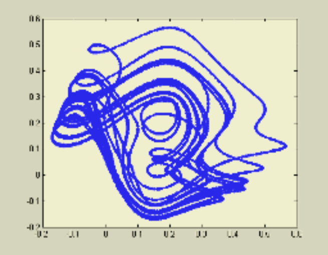


Fig. 2. *a*=3, *b*=8, phase plane diagram for system (1)

computational overhead etc, such as Arnold image encryption is good algorithm. In [5], the authors first extended the cat map to "general cat map", then the general cat map is used to encrypted image.

The classical Arnold cat map is a two-dimensional invertible chaotic map [7] described by

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \bmod 1 \cdot \tag{2}$$

The map is area-preserving since the determinant of its linear transformation matrix is equal to 1. The Lyapunov characteristic exponents of the map are the eigenvalues σ_1 and σ_2 of the matrix in (3), given by

$$\sigma_1 = \frac{1}{2}(3+\sqrt{5}) > 1, \quad \sigma_2 = \frac{1}{2}(3-\sqrt{5}) < 1 \quad . \tag{3}$$

The above 2D cat map is now generalized by introducing two control parameters, *a* and *b*, as follows:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & a \\ b & ab-1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \bmod 1 \quad . \tag{4}$$

One can easily verify that det **A**=1,which means that the discrete version of the general map is a 1-1 map, and that its mixing property and the sensitivity to initial condition and parameters are kept unchanged. In (4), the parameters (*a* and *b*) are key of the general cat map. So how to choose *a* and *b* is very important.

3.2 Key Scheming

According to the basic principle of cryptology [3], a cryptosystem should be sensitive to the key, i.e., the cipher-text should have close correlation with the key. There are two ways to accomplish this requirement: one is to mix the key thoroughly into the plain-text through the encryption process; another is to use a good (ideally, truly random) key generation mechanism.

The key (denote K_m) directly used in the proposed encryption scheme have three parameters (*a*, *b* and *n* of the general cat map, where *a*, *b*, *n* are integer, *n* is encryption iterative times). While the user's input key (denote K_u) is a string of characters, which can be taken as a sequence of bits. Thus, there is a transform K_u to K_m, during which a diffusion mechanism is introduce, so as to protect the key from opponent's attacks.

In the proposed scheme, Liao's chaotic neural system is employed in key scheming, which is modeled by (1), When b>1.868, the system is chaotic. Simulation shows that the system orbit is extremely sensitive to the parameter b, therefore b is used to control the generation of the cipher key. Suppose user input a binary sequence of 64 bits. The binary sequence is divided into four segments, denoted as k_a, k_b, k_n, k_m, respectively, each with 16-bit long. So, them are used to generate the control of general cat map.

To generate a and b of the general cat map, the following are fist used to compute the control b of Liao's system:

$$\mathrm{b}=K_a \times k_b + 1.868$$

Where $K_a = \sum_{i=0}^{15} k_a(i) \times 2^{-i}$,in which $k_a(i)$ is the ith bit in sequence k_a.

Initial values $x(1)$ of Liao's system are derived from k_m by using the following formulas:

$$x(1)= | K_m - 0.5 |$$

Where $K_m = \sum_{i=0}^{15} k_m(i) \times 2^{-i}$,in which $k_m(i)$ is the ith bit in sequence k_m.

Then, in the next step, parameters are set as a=3 and the other parameters obtain above are used to compute Liao's system, respectively, yielding three time values when t=11000, t=12000 and t=13000(denoted as $x_{11000}, x_{12000}, x_{13000}$). Next then, the following formulas are used to obtain the final parameters values of a , b and n for general cat map:

$$\begin{aligned} \mathrm{a} &= x_{11000} \times \mathrm{N}, \\ \mathrm{b} &= x_{12000} \times \mathrm{N}, \\ \mathrm{n} &= x_{13000} \times k_n . \end{aligned} \tag{5}$$

where N is length of original image.

The Fig.3-4 shows the original image and encryption image.

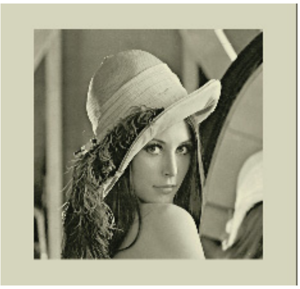


Fig. 3. Original image

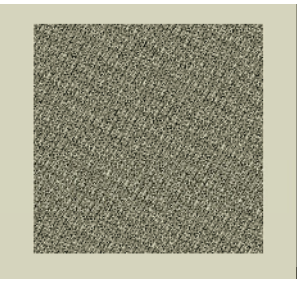


Fig. 4. Encryption image

4 Conclusion

It is well-known that the time-delay chaotic neural system have many good properties, because it can generate complicated dynamical behaviors, have super-chaotic propriety and have many positive Lapunov exponents. In this paper, the general cat map is employed to shuffle the position of image pixels and another chaotic map is used to confuse the relationship between cipher-image and plain-image. The general cat map's parameters are derived from the time-delay chaotic neural system. Experimental tests have also demonstrated the high security and fast speed of the new image encryption scheme.

Acknowledgements

The work described in this paper was supported by the National Natural Science Foundation of China (No. 60271019), the Doctorate Foundation Grants from the Ministry of Education of China (No. 20020611007), the Post-doctoral Science Foundation of China and the Natural Science Foundation of Chongqing.

References

1. Liao, X.F., Wong, K.W., Leung C.S., et al.: Hopf bifurcation and Chaos in a Single Delayed Neuron Equation with Non-monotonic Activation Function. Chaos, Solitions and Fractals, **12** (2001) 1535-1547
2. Mao, .B, Chen, G.R., et al: A Novel Fast Image Encryption Scheme Based on 3D Chaotic Baker Maps. International Journal of Bifurcation and Chaos, **14** (2004) 3613-3624
3. Schneier B.: Applied Cryptography: Protocols, Algorithms, and Source Code in C. 2nd ed. Wiley; New York (1995)
4. Li, S.J., Zheng, X.: Cryptanalysis of a Chaotic Image Encryption Method. In: IEEE Int Symposium Circuits and Systems, Scottsdale, USA (2002)
5. Ma, Z., Qiu S.: An Image Cryptosystem Based on General Cat Map. Journal of China Institute of Communication , **24** (2003), 51-57
6. Qi, D., Zou, J., et al.: A Novel Transformation and Its Application In Image Formation Hide. Science in China (Series E), **30** (2003)
7. http://mathworld.woffram.com/ArnoldsCatMap.html

Effectiveness of Different Target Coding Schemes on Networks in Financial Engineering

Kidong Lee[1], Junghee Park[2], and Sangjae Lee[3]

[1] Department of Business, University of Incheon, Incheon, South Korea
kdlee@incheon.ac.kr

[2] School of Information Technology, Jaeneung College, Incheon, South Korea
jpark@mail.jnc.ac.kr

[3] Department of E-business, College of Business, Sejong University, South Korea
Sangjae@sejong.ac.kr

Abstract. The effectiveness of backpropagation (BP) network relies mainly on a proper architectural design as well as finding appropriate parameter values by training the network at the same time. In this experiment, we test the effectiveness of three different target coding schemes including Lowe and Webb's method of BP using unmatched business samples in the context of financial engineering. The results of the study show that Lowe and Webb's reciprocally weighted target-coding scheme may not be the best choice among the three target coding schemes in neural network applied. Rather, in neural net experiment, no knowledge about prior proportion on sample shows the promising result.

1 Introduction

Previous studies in the artificial neural network (ANN) suggest that these connection based technologies are promising in the recognition of ever-changing business patterns. Among many types of neural networks, Back Propagation (BP) networks have been the most frequently used algorithms in the field of financial engineering, specially in the prediction of business failure [1], [9],[12],[14].

The focus of this study is to provide a way to test the effectiveness of three target-coding schemes including Lowe and Webb [7] applied for unbalanced financial data sets in the bankruptcy prediction. Many of bankruptcy prediction studies focus only on matched cases ignoring the fact that bankrupt firms in the real world are indeed small portions of the entire population. Even when they launch an unmatched study, researchers often forget to implement target coding scheme so that in the past research show a limited contribution on the effect of target coding scheme to neural network classifiers. This study is to fill the gap of the past research and is to provide a more appropriate way to find network parameters, it is hoped, leading to better classifiers.

2 Prior Literature

Neural networks were not used as a bankruptcy classification technology until the early 90's. Odom and Sharda [8] were the first researchers to investigate the feasibil-

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 873–878, 2005.

ity of neural networks in firm failure prediction. After this first neural experiment, a significant volume of neural network research followed [1],[2],[3],[5],[6],[9],[10],[12],[14]. A few selected major neural nets on financial application are shown in Table 1.

Table 1. Neural network applications applied in financial engineering

Author(s)	Data	Methodology	Focus of the study
Odom & Sharda	129 firms	BP	First network application in Bankruptcy prediction
Tam & Kiang	59 pairs	BP, linear ID3 logistic, *k*NN	Misclassification cost are included in bankruptcy cost
Wilson & Sharda	Resampling Techniques	BP, statistical	Monte Carlo resampling techniques applied
Jo and Han	Korean firms	DA, case-based BP	Case-based reasoning applied
O'Leary	15 prior studies	Meta-study	Comparative studies of previous bankruptcy studies
Alam et al.	100 firms	Fuzzy clustering Self-organizing	Fuzzy algorithm and self-organizing map applied

3 Research Design and Methodology

3.1 Dataset and Variables

The data sample for this study consists of Korean firms that have filed for bankruptcy during the period of 1995 - 1998, selected from a pool of the listed companies on the Korean Stock Market stored in a database in the Korea Investors Service (KIS), Inc. In this data set, financial institutions such as commercial banks or investment bankers are excluded. The whole population set of Korean firms traded in the Korean stock market is used. At the end of 1998, the total number of listed firms traded in the Korean stock market was about seven hundred, which does not include financial companies. Searching for a failed firm gives in 113 non-financial failed firms in the listed companies in the Korean stock market during the period. Then, the availability of the financial ratios of the failed firms used for this study further reduced the final bankrupt sample size to 84.

After examining the availability of financial ratios used in this study, the number of non-bankrupt firms comes down to 460. Therefore, the total number of objects used for this study is 544, consisting of 84 failed firms and 460 non-failed firms. The ratio of the failed firms to the non-failed firms in this unmatched sample is 1 to 5.5 (or 15.4% of the Korean listed firms went bankrupt). Each firm is described by Altman's five variables such as working capital/total assets, retained earnings/total assets, earnings before interest and taxes/total assets, market value of equity/book value of total debt, and sales/total assets since the prediction capabilities of these ratios were well documented in the previous literature [1],[3],[8],[14].

Bias due to changing data set composition could have a detrimental impact on determining neural network architecture and its parameters. The cross-validation technique enables us to use a whole data set so that, it is hoped, any bias effect would be minimized [12],[14]. Four-fold cross-validation technique is introduced to investigate the classification performance of neural networks in terms of sampling variation. Training will be conducted on any two of the four subsets while the remaining two sets are used for validation and testing, respectively. Due to this four-fold cross-validation, the outputs of this study are resulted in four different subsets.

3.2 Three Coding Methods Comparison

CODING METHOD 1:
Uniform weight coding for target vector / no information on prior group proportion.
In this case, the target vector uses uniform weighting, that is, 0 for a non-bankrupt firm and 1 for a bankrupt firm. Also no information on the prior group proportion is available. Whether an object is classified into bankrupt or non-bankrupt, we need a threshold decision or a cut-off value. For this particular case, we use the following rule that if a majority of output value belongs to a particular output node, that input object is classified to that output group. Notice that this classification decision is the exactly same as the .5/.5 threshold decision used in most matched bankruptcy study. A logistic function is used in all hidden nodes in this scheme.

CODING METHOD 2:
Uniform weight coding for target vector / group proportion available.
Again, the target vectors are used the uniform coding scheme, coded as 0 for a non-bankrupt firm and 1 for a bankrupt firm. However, in this particular case, it is assumed that the information on the prior group proportion is available. Thus, the threshold of how to classify an object into one of the two groups can incorporate the prior group proportion information. That is, in this particular unmatched problem in hand, we have informed on the prior proportion that the bankrupt firms is 15.44% and the non-bankrupt firms 84.56% of the whole population. Since the proportion of the bankrupt group is 15.44%, if the output value of the output node 1 is greater than or equal to 0.1544, it belongs to a bankrupt group. The above statement is the exactly same as the following statement. If the output value of the output node 2 is less than .8456, that an object is classified as a bankrupt one. Otherwise, an object is classified a non-bankrupt. Again, a logistic function is used in the hidden nodes and output nodes in this scheme.

CODING METHOD 3:
Reciprocally-weighted target coding [7]
In Lowe and Webb's weighted target coding scheme, the target vector is inversely proportional to the square root of the proportions of samples in each class in the training set. That is,

$$t^p_k = (P/n_k)^{1/2} \text{ if } x^p \in k \text{ and 0 otherwise.} \quad (1)$$

Where P is the total number of object patterns, n_k is the number of patterns in each group, $t^p{}_k$ is the target value of class k for pattern p, and x^p is the input value for pattern p.

We use this particular coding scheme because in the unmatched bankruptcy case, the largest class dominates the classification membership mainly because of usage of the square error function for a cost function [1],[7]. Output vectors produced by using this target-coding scheme supposedly prevent a neural network from overcompensating for unmatched class proportion. Unlike the two other unweighted coding schemes that both use a logistic function for the hidden and output nodes, We use a logistic function at the hidden nodes and a linear function instead at the output node for this reciprocally-weighted target coding method as suggested by Lowe & Webb [7].

4 Experimental Results

In this financial engineering study, we explore the three methods described: Coding Methods 1, 2 and 3. Especially, we focus on the following two issues: first, the sensitivity of prior group proportion on bankruptcy decision and second, the efficacy of Lowe and Webb's coding scheme [7] compared with the two other methods.

Table 2 shows the performance effects of BP networks when applied to four subsets of the unmatched bankruptcy prediction sample. Each of the three methods employs different cut-off thresholds to classify a bankrupt cluster from a non-bankrupt cluster, given the knowledge about the prior proportion of the population and the weighting schemes for target vectors as explained earlier.

Coding method 1 and 2 give the performance results of the unweighted target-coding schemes of the BP networks. Specifically, Coding method 1 gives the classification and prediction rates when a .5/.5 threshold to separate a bankrupt from a non-bankrupt is used. Coding method 2 gives the performance outcomes when one uses the prior proportion as a threshold, that is, a .1544/.8456. Whereas these first two methods show the performance results of the unweighted cases, Coding method 3 provides the outcomes of the reciprocally weighted target-coding scheme suggested in the Lowe and Webb [7]. Especially, the maximum gain decision criterion, which provides the maximum classification rate, is used.

The best model selection is the process of applying the two criteria: first, choose the highest prediction performance of test sets from each subset and if there were a tie among the model prediction accuracy, the parsimonious model, the one with the least number of hidden nodes, was selected.

As shown in Table 2, where the .5/.5 cut-off threshold with unweighted target coding strategy (Coding method 1) is used, the prediction rates of the best models identified for each subset are {86.76%, 86.76%, 88.97%, 91.18%}. The .1544/.8456 cut-off threshold with unweighted target coding strategy (Coding method 2) produces the prediction rates as {83.82%, 86.03%, 86.76%, 88.24%}. When the Lowe and Webb's model (Coding method 3) is used, the prediction rates for the best models identified are {76.47%, 85.29%, 80.15%, 82.35%}.

Table 2. Summary of performance comparison of three target coding methods

Target coding	Subset 1	Subset2	Subset 3	Subset 4
Coding method 1	86.76%	86.76%	88.97%	91.18%
Coding method 2	83.82%	86.03%	86.76%	88.24%
Coding method 3	76.47%	85.29%	80.15%	82.35%

As shown, performance in terms of prediction rates, the .5/.5 cut-off with unweighted target coding strategy used is clearly better than the two other schemes used. Theoretically, the Lowe and Webb's target coding scheme would correct the tendency of overcompensating to the largest group proportion in training. And thus it is expected that the Lowe and Webb coding scheme should provide not only the highest performance rate but also fix the problem of overcompensating to the largest group in an unmatched sample proportion. But this result clearly shows that is not the case. In addition, the best method is the .5/.5 threshold strategy even though the prediction rate differences between the .5/.5 strategy and the .1544/.8456 strategy seems to be minimal. One may think from these experimental results that in the two-group classification problem, knowing the exact prior group proportion may not be necessary. It may imply that the threshold decision is not sensitive to the prior group proportion.

5 Summary and Conclusions

Several things were tested in this study. Among them, the four-fold cross-validation scheme has been applied to the data set to reduce the effect of sampling variability. Also, the three different strategies for threshold decisions of the BP networks were tested: (1) .5/.5 threshold, (2) .1544/.8456 threshold, and (3) the Lowe and Webb's method [7].

The findings of the financial engineering study may be summarized in two important things. First, threshold decision may not be very sensitive to the prior group proportion. In reality, threshold decision (or cutoff decision) largely depends on the degree of separation of data. Second, the Lowe and Webb's reciprocally weighted target coding scheme, compared with two other coding schemes, did not work well for this particular real-world data set, despite its sound theoretical base. Thus, we tentatively believe that the result of our research also confirm that the result of neural network experiment is fundamentally data-driven. In the same logic, we definitely need to test more different types of data, besides ours. However, at this juncture, from our research, we may say that researchers may not have to pay too much attention to adjust parameters of neural network, target coding and/or cutoff in this type of neural network research.

References

1. Alam, P, Booth, D, Lee, K, Thordarson, T.: The Use of Fuzzy Clustering Algorithm and Self-Organizing Neural Networks for Identifying Potentially Failing Banks: An Experimental Study. Expert Systems With Applications, **18** (2000) 185-199

2. Back, B., Laitinen, T., and Sere, K.: Neural Networks and Genetic Algorithms for Bankruptcy Predictions. Expert Systems With Applications, **11** (1996) 407-413
3. Boritz, J. E. and Kennedy, D. B.: Effectiveness of Neural Network Types for Prediction of Business Failure. Expert Systems With Applications, **9** (1995) 503-512
4. Jo, H.Y., and Han, I.G.: Integration of Case-Based Forecasting, Neural Network, and Discriminant Analysis for Bankruptcy Prediction. Expert Systems With Applications, **11** (1996) 415-422
5. Lacher, R.C., Coats, P.K., Sharma, S.C., and Fant, L.F.: A Neural Network for Classifying the Financial Health of a Firm. European Journal of Operations Research, **85** (1995) 53-56
6. Leshno, M. and Spector, Y.: Neural Network Prediction Analysis: The Bankruptcy Case, Neurocomputing, **10** (1996) 125-147
7. Lowe, D. and Webb, A.: Optimized Feature Extraction and the Bayes Decision in Feed-Forward Classifier Networks, IEEE Transactions on Pattern Analysis and Machine Intelligence, **13** (1991)
8. Odom, M. and Sharda, R.: A Neural Network Model for Bankruptcy Prediction. Proceedings of the IEEE International Conference on Neural networks (1990) 163-168
9. O'Leary, D.E.: Using Neural Networks to Predict Corporate Failure. International Journal of Intelligent Systems in Accounting. Finance & Management, **7** (1998) 187-197
10. Salchenberger, L.M., Cinar, E.M., and Lash, N.A.: Neural Networks: A New Tool for Predicting Thrift Failures. Decision Sciences, **23** (1992) 899-916
11. Sharda, R. and Wilson, R.L.: Neural Network Experiments in Business-Failure Forecasting: Predictive Performance Measurement Issues. International Journal of Computational Intelligence and Organization, **1** (1996) 107-117
12. Tam, K.Y. and Kiang, M.Y.: Managerial Applications of Neural Networks: The Case of Bank Failure Predictions, Management Science, **38** (1992) 926-947
13. Wilson, R.L. and Sharda, R.: Bankruptcy Prediction Using Neural Networks. Decision Support Systems, **11** (1994) 545-557
14. Zhang, G., Hu, M.Y., Patuwo, B.E., and Indro, D.C.: Artificial Neural Networks in Bankruptcy Prediction: General Framework and Cross-Validation Analysis. European Journal of Operational Research, **116** (1999) 16-32

Select the Size of Training Set for Financial Forecasting with Neural Networks

Wei Huang[1,2], Yoshiteru Nakamori[1], Shouyang Wang[2], and Hui Zhang[1]

[1] School of Knowledge Science, Japan Advanced Institute of Science and Technology
Asahidai 1-1, Tatsunokuchi, Ishikawa, 923-1292, Japan
{w-huang,nakamori,zhang-h}@jaist.ac.jp

[2] Institute of Systems Science, Academy of Mathematics and Systems Sciences
Chinese Academy of Sciences, Beijing, 100080, China
swang@iss.ac.cn

Abstract. The performance of financial forecasting with neural networks dependents on the particular training set. We design mean-change-point test to divide the original dataset into different training sets. The experiment results show that the larger training set does not necessarily produce better forecasting performance. Although the original datasets are different, the change-points to produce the optimal training sets are close to each other. We can select the suitable size of training set for financial forecasting with neural networks based on the mean-change-point test.

1 Introduction

Neural networks can suffer high variance in the financial forecasting performance, because they are data-driven and model-free. The financial researchers have used various sizes of training sets to train neural network, ranging from one year to sixteen years [2], [4], [5], [6], [9]. However, once the researchers have obtained their training data, they typically use all of the data in building the neural network forecasting model, with no attempt at comparing the effect of data quantity on the performance of the forecasting models. Recently, Walczak examined the effects of different sizes of training sample sets on forecasting foreign exchange rates. The presented results indicate that for financial time series two years of training data is frequently all that is required to produce appropriate forecasting accuracy [8].

Financial time series are affected by many highly correlated economic, political and even psychological factors. These factors interact in a very complex fashion and cause the varying behavior of financial time series. Changes in time series often occur gradually so that there is a certain amount of fuzziness in the change point [3]. Therefore, we can conjecture that the movement of financial time series has a series of change points, which divide data into several groups that take heterogeneous features from each other. Our contributions are to design mean-change-point test to detect change points in financial time series and produce training sets based on the change points. The remainder of this paper is organized as follows. Section 2 describes the

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 879–884, 2005.

mean-change-point test. Section 3 give the process to produce different training set based on mean-change-point test. In Section 4, we design an experiment to compare the prediction performance of the neural networks trained with the different training sets from mean-change-point test. Finally, conclusions are given in Section 5.

2 Mean-Change-Point Test

We assume that a change-point divides time series into two segments that have as much different means as possible. The mean-change-point test are described as follows:

First, for the n observations x_1, x_2, …, x_n, we calculate statistic F as follows:

$$F = \sum_{t=1}^{n} (x_t - \overline{x})^2 \ . \tag{1}$$

where $\overline{x} = \frac{\sum_{t=1}^{n} x_t}{n}$.

Then at each place j in the series, we calculate

$$F_j = \sum_{t=1}^{j} (x_t - x_{j1})^2 + \sum_{t=j+1}^{n} (x_t - x_{j2})^2 \ . \tag{2}$$

where $x_{j1} = \frac{\sum_{t=1}^{j} x_t}{j}$, $x_{j2} = \frac{\sum_{t=j+1}^{n} x_t}{n-j}$.

Next, for each point in the sequence, we calculate $F - F_j$; and then set

$$c = \arg_{1<j<N} \max(F - F_j) \ . \tag{3}$$

where c is the estimated change point in the sequence when the maximum value of $F - F_j$ occurs.

3 Produce Training Sets by Change-Point Detection

For the N observations y_1, y_2, …, y_N, we first set the minimal size of training set $T(T < N)$ and the initial start-point $s_1(1 < s_1 < N - T)$. The training sets are produced as follows:

Step 1. Let $i=1$.

Step 2. The i th training set contains the observations with the serial number ranging from s_i to N, namely y_{s_i}, y_{s_i+1},..., y_N.

Step 3. Detect the change point c_i ($s_i \le c_i \le N$) in y_{s_i}, y_{s_i+1},..., y_N by the mean-change-point test.

Step 4. If $N-c_i$ is less than T, exit. Otherwise, let $s_{i+1}=c_i+1$ and $i=i+1$.Go to Step 2.

4 An Experiment

In our experiment, neural networks are developed to predict the daily price of the U.S. dollar(USD) against the Japanese yen(JPY). We compare the prediction performance of neural networks trained with the training sets from mean-change-point test.

4.1 Neural Network Model

In order to reduce the degrees of freedom in the developed neural network models and to maintain consistency with previous research efforts, we focus on 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. Based on an empirical analysis of sampling interval for short-term exchange rate forecasting with NNs [1], we employ 4 days as sampling interval of daily exchange rates. That is, the inputs to the NNs forecasting model are p_t, p_{t-4} ... , $p_{t-4*(m-1)}$ and the output is p_{t+n}, when we make a prediction n days ahead using the neural network with m input nodes. For simplicity, we let m=4, n = 3 in the following experimental analysis.

Generally speaking, too many nodes in the hidden layer produce a network that memorizes the input data and lacks the ability to generalize. It has been shown that the in-sample fit and the out-of-sample forecasting ability of NNs are not very sensitive to the number of hidden nodes [9]. Parsimony is a principle for designing NNs. Hence, the number of hidden nodes is equal to the number of input nodes.

4.2 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 values x_k and predicted values $\hat{x}_k$, the NMSE can be defined as follows:

$$\text{NMSE} = \frac{\sum_{k\in P}(x_k - \hat{x}_k)^2}{\sum_{k\in P}(x_k - \bar{x}_k)^2} \ . \tag{4}$$

where $\bar{x}_k$ is the mean of actual values.

4.3 Data Preparation

From Pacific Exchange Rate Service provided by Professor Werner Antweiler, University of British Columbia, Canada, we obtain 2230 daily observations covering the period from Jan 4, 1993 to Oct 23, 2001. The data have corresponding serial numbers, ranging from 1 to 2230, namely N =2230. In our study, there are four input nodes, four hidden nodes and one output node in the neural network. So the number of connections contained in the neural network is (4 + 1) × 4 + (4 + 1) × 1 = 25. A lower bound on the number of training examples required to train a back-propagation network is four times the number of weighted connections contained in the neural network [7]. Therefore, the minimum training set should contain 100 observations, namely $T = 100$.

To compare the change-points detected from the different original time series, we let initial start-point s_1 =1, 258, 514, 770, 1025, corresponding to the five datasets, of which the start year is 1993, 1994, 1995, 1996, 1997, respectively. We apply the mean-change-point test on the five datasets to detect change-points and produce the different training sets shown in Table 1-5. The performance of the neural networks trained with the different training sets are compared by using the test set containing the daily exchange rates from Oct 24, 2001 to Dec 28, 2001.

4.4 Results

Take Table 1 as an example. As the start-point of the training set increases, the value of NMSE first decreases and then increase. The neural networks perform best in the 3rd training set. It implies that the larger training set do not necessarily produce better forecasting performance. The similar pattern can be observed in Table 2-5. Neural networks are puzzled, when we introduce data into the model that is not representative of the current time series behavior. On the other hand, neural networks cannot get enough information to investigate the underlying relationship in financial time series if the size of training set is too small. The suitable size of training set provides the balance between getting enough information and grasping the underlying law of the current time series behavior.

The start-points of training sets for the best prediction performance are 1666, 1668, 1665, 1665, 1665 in Table 1-5. That is to say, we can produce almost the same optimal training sets based on the mean-change-point test, although the original datasets are different. Hence, we can select the suitable size of training set for financial forecasting with neural networks based on the mean-change-point test.

Table 1. The prediction results of neural networks trained with the training sets from mean-change-point test(start year=1993)

Training set i	Start-point s_i	Change-point c_i	Training set size	NMSE
1	1	947	2230	0.2081
2	948	1665	1283	0.1831
3	1666	2022	565	0.1548
4	2023	2071	208	0.2079
5	2072	2182	159	0.2560

Table 2. The prediction results of neural networks trained with the training sets from mean-change-point test(start year=1994)

Training set i	Start-point s_i	Change-point c_i	Training set size	NMSE
1	258	872	1973	0.2349
2	873	1667	1358	0.2002
3	1668	2022	563	0.1461
4	2023	2071	208	0.2079
5	2072	2182	159	0.2560

Table 3. The prediction results of neural networks trained with the training sets from mean-change-point test(start year=1995)

Training set i	Start-point s_i	Change-point c_i	Training set size	NMSE
1	514	769	1717	0.2259
2	770	1001	1461	0.1968
3	1002	1664	1229	0.1685
4	1665	2022	566	0.1579
5	2023	2071	208	0.2079
6	2072	2182	159	0.2560

Table 4. The prediction results of neural networks trained with the training sets from mean-change-point test(start year=1996)

Training set i	Start-point s_i	Change-point c_i	Training set size	NMSE
1	770	1001	1461	0.1968
2	1002	1664	1229	0.1685
3	1665	2022	566	0.1579
4	2023	2071	208	0.2079
5	2072	2182	159	0.2560

Table 5. The prediction results of neural networks trained with the training sets from mean-change-point test(start year=1997)

Training set i	Start-point s_i	Change-point c_i	Training set size	NMSE
1	1025	1664	1206	0.1941
2	1665	2022	566	0.1579
3	2023	2071	208	0.2079
4	2072	2182	159	0.2560

5 Conclusions

To select the suitable training set for financial forecasting with neural networks, we design mean-change-point test to divide the original dataset into different training sets. We conduct an experiment on the exchange rate(USD/JPY) forecasting and compare the prediction performance of the neural networks trained with the different training sets. The experiment results show that the prediction performances of neural networks firstly improve and then decrease, as the size of the training set decrease. Although the original datasets are different, the change-points to produce the optimal training sets are close to each other. We can select the suitable size of training set for financial forecasting with neural networks based on the mean-change-point test.

References

1. Huang, W., Lai, K.K., Nakamori, Y. & Wang, S.Y.: An empirical Analysis of Sampling Interval for Exchange Rate Forecasting With Neural Networks. Journal of Systems Science and Complexity, **16** (2003) 165-176
2. Jamal, A.M., & Sundar, C.: Modeling Exchange Rates with Neural Networks. Journal of Applied Business Research, **14** (1998) 1-5
3. Kumar, K. & Wu, B.L.: Detection of Change Points in Time Series Analysis with Fuzzy Statistics. International Journal of Systems Science, **32** (2001) 1185-1192
4. Refenes, A.N.: Constructive Learning and Its Application to Currency Exchange Rate Forecasting. In R. Trippi and E. Turban (Ed.), Neural Networks in Finance and Investing, New York: Irwin (1993) 465-493
5. Steurer, E.: Nonlinear modelling of the DEM/USD Exchange Rate. In A. Refenes (Ed.), Neural Networks in the Capital Markets, New York: Wiley (1993) 199-211
6. Tenti, P.: Forecasting Foreign Exchange Rates Using Recurrent Neural Networks. Applied Artificial Intelligence, **10** (1996) 567-581
7. Walczak, S. & Cerpa, N.: Heuristic principles for the Design of Artificial Neural Networks. Information and Software Technology, **41** (1999) 107-117
8. Walczak, S.: An Empirical Analysis of Data Requirements for Financial Forecasting with Neural Networks. Journal of Management Information System, **17** (2001) 203-222
9. Zhang, G., & Hu, M.Y.: Neural Network Forecasting of the British Pound/Us Dollar Exchange Rate. Omega, International Journal of Management Science, **26** (1998) 495-506

Estimating the Yield Curve Using Calibrated Radial Basis Function Networks

Gyusik Han, Daewon Lee, and Jaewook Lee

Department of Industrial and Management Engineering
Pohang University of Science and Technology
Pohang, Kyungbuk 790-784, Korea
{swallow,woosuhan,jaewookl}@postech.ac.kr

Abstract. Nonparametric approaches of estimating the yield curve have been widely used as alternative approaches that supplement parametric approaches. In this paper, we propose a novel yield curve estimating algorithm based on radial basis function networks, which is a nonparametric approach. The proposed method is devised to improve accuracy and smoothness of the fitted curve. Numerical experiments are conducted for 57 U.S. Treasury securities with different maturities and demonstrate a significant performance improvement to reduce test error compared to other existing algorithms.

1 Introduction

Along with stocks and loans, bonds are the main asset class with which the government or a private enterprise can raise money. The term structure of interest rates plays a key role in the reasonable bond-pricing. Because it is based on an interest rate model, the model need be calibrated before using it. The fundamental object to calibrate against is the yield curve of bonds. The fitting techniques of yield curves to yield data or maturity data are divided into two categories, parametric approaches and nonparametric approaches. A popular parametric approach is a family of Nelson-Siegel curves [2]. Despite its simple structure, many parametric approach, however, revealed poor performance in yield curve fitting. As alternative approaches, nonparametric approaches have recently been widely used. Especially, cubic B-splines, which are the kernel functions for estimating a bond yield, are popularly chosen among various nonparametric approaches [3], [5], [13], [14].

In this paper, we propose a novel fitting method for the yield curve. The proposed method comprises two phases. The first phase finds an initial yield curve using a radial basis function network (RBFN) whose inputs are maturities of bonds and outputs are their yields. In the second phase, the initial yield curve predicted by the RBFN in the first phase is calibrated to minimize bond-pricing error. The performance of the proposed algorithm is verified concerned with improving the curve-fitting accuracy and regularization by applying it to notes and bonds issued by U.S. Treasury Department.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 885–890, 2005.

2 Existing Yield Curve Fitting Algorithms

To verify the merits of our proposed nonparametric and computational method for fitting the yield curve, the performance of this method needs to be measured against widely used alternative models. In this paper, a smoothed spline model is used for this purpose. It assumes that the forward rate curve, δ, is a linear combination of basis functions. Cubic B-splines, a linear nonparametric interpolation method, are considered as basis. Let $\{\mathrm{P}_i\}_{1\leq i\leq N}$ be the ith bond price at settlement date. δ is chosen to minimize the following objective function

$$\sum_{i=1}^{N}\{P_i - \hat{P}_i(\delta)\}^2 + \theta\int_0^{\tau_K}\left[\delta''(t)\right]^2 dt \tag{1}$$

over the space of all cubic B-splines with knot points(time) $\tau_0 < \tau_0 < \cdots < \tau_K$. $\hat{P}_i(\delta)$ is the ith theoretical bond price from the estimated yield curve [14].

Another approach to be considered is Nelson and Siegel curves [1], [2]. The forward rate curve modeled by Nelson and Siegel is

$$\delta(\tau) = \beta_0 + (\beta_1 + \beta_2\tau)e^{-k\tau} \tag{2}$$

where $\beta_0,\ \beta_1,\ \beta_2,$ and k are parameters to be estimated in the following way:

$$(\beta_0^*,\ \beta_1^*,\ \beta_2^*,\ k^*) = \arg\min_{(\beta_0,\ \beta_1,\ \beta_2,\ k)}\left\{\varUpsilon \mid \varUpsilon = \sum_{i=1}^{N}[P_i - \hat{P}_i(\delta)]^2\right\}$$

3 The Proposed Method

3.1 Phase I: RBFN-Based Initial Curve-Fitting Phase

In the first phase, we train a RBFN to estimate the spot rate curve with a given bond information on prices and maturities. A (generalized) radial basis function network (RBFN) involves searching for a suboptimal solution in a lower-dimensional space that approximates the interpolation solution where the approximated solution $F^*(\mathbf{w})$ can be expressed as follows:

$$\begin{aligned} F^*(\mathbf{w};\tau_i,\mathbf{T},\mathbf{L}) &= \sum_{j=1}^{l} w_j\boldsymbol{\varPhi}_j(\tau_i) \\ &= \sum_{j=1}^{l} w_j\phi\left(\frac{\|\tau_i - t_j\|}{\lambda_j}\right) \end{aligned} \tag{3}$$

where the set of RBF centers, $\mathbf{T} = \{t_j | j = 1,\ldots,l\}$, and the set of scale parameters, $\mathbf{L} = \{\lambda_j | j = 1,\ldots,l\}$, are given [7]. Hence the training procedure of the GRBFNs is composed of selecting centers of the hidden neurons and estimating the weights that connect the hidden and the output layers. Once centers have been fixed, the network weights will be directly estimated by using the least squares algorithm.

To apply RBFNs to estimate the spot rate, we first calculate the spot rate (r_i) with respect to maturity (τ_i) on the basis of bootstrapping through bond information. Then an initial yield curve is fitted as minimizing the criterion function

$$J(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N} \|r_i - F^*(\mathbf{w}; \tau_i, \mathbf{T}, \mathbf{L})\|^2. \tag{4}$$

The network weights can be directly estimated by using the pseudo-inverse,

$$\mathbf{w} = (\mathbf{\Phi}^T\mathbf{\Phi} + \lambda\mathbf{\Phi}_0)^{-1}\mathbf{\Phi}^T\mathbf{R} \tag{5}$$

where $\mathbf{R}$ is the vector of the spot rates obtained from the bootstrapping method, $\mathbf{\Phi} = [\phi(\tau_i, t_j)]_{i=1,\dots,N,\ j=1,\dots,K}$, $\mathbf{\Phi}_0 = [\phi(\tau_i, t_j)]_{i,j=1,\dots,K}$ and λ is a regularization parameter of the generalized RBFN [9].

The estimated spot rates, obtained from the trained RBFN, can be used to predict bond prices. Because the ultimate aim of the yield curve fitting is to construct a bond pricing model, in the second phase, we have to additionally optimize the weights of the trained RBFN to minimize bond pricing error.

3.2 Phase II: Trust Region-Based Optimal Yield Curve Search

The second phase solves an unconstrained nonlinear programming to minimize the bond pricing error as follows:

$$\min_{\mathbf{w}} E(\mathbf{w}; \mathbf{R}, \mathbf{T}, \mathbf{L}) = \sum_{i=1}^{N} \Big\{P_i(r(\boldsymbol{\tau})) - \hat{P}_i(F^*(\boldsymbol{\tau}; \mathbf{w}, \mathbf{T}, \mathbf{L}))\Big\}^2 \tag{6}$$

where P_i is the ith bond price for the true spot rate $r(\boldsymbol{\tau})$ at time $\boldsymbol{\tau}$ and $\hat{P}_i$ is the ith bond price for the spot rate $F^*(\mathbf{w})$ obtained from the initially trained RBFN in the first phase. To minimize Eq. (6) the second phase employs a trust region algorithm as follows. For a given weight vector $\mathbf{w}(n)$, the quadratic approximation $\hat{E}$ is defined by the first two terms of the Taylor approximation to E at $\mathbf{w}(n)$;

$$\hat{E}(\mathbf{s}) = E(\mathbf{w}(n)) + \mathbf{g}(n)^T\mathbf{s} + \frac{1}{2}\mathbf{s}^T\mathbf{H}(n)\mathbf{s} \tag{7}$$

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

$$\min_{\mathbf{s}} \hat{E}(\mathbf{s}) \quad \text{subject to} \quad \|\mathbf{s}\|_2 \le \Delta_n \tag{8}$$

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

$$\rho_n = \frac{E(\mathbf{w}(n)) - E(\mathbf{w}(n) + \mathbf{s}(n))}{\hat{E}(\mathbf{0}) - \hat{E}(\mathbf{s}(n))}, \tag{9}$$

Δ_n is adjusted between iterations as follows:

$$\Delta_{n+1} = \begin{cases} \|\mathbf{s}(n)\|_2/4 & \text{if } \rho_n < 0.25 \\ 2\Delta_n & \text{if } \rho_n > 0.75 \text{ and } \Delta_n = \|\mathbf{s}(n)\|_2 \\ \Delta_n & \text{otherwise} \end{cases} \tag{10}$$

The decision to accept the step is then given by

$$\mathbf{w}(n+1) = \begin{cases} \mathbf{w}(n) + \mathbf{s}(n) & \text{if } \rho_n \geq 0 \\ \mathbf{w}(n) & \text{otherwise} \end{cases} \tag{11}$$

which means that the current weight vector is updated to be $\mathbf{w}(n) + \mathbf{s}(n)$ if $E(\mathbf{w}(n) + \mathbf{s}(n)) < E(\mathbf{w}(n))$; Otherwise, it remains unchanged and the trust region parameter Δ_n is shrunk and the trial step computation is repeated [8], [10], [11].

The proposed method has several advantages. At first, since RBFN has a universal property of approximating an arbitrary curve, it can provide a good fitting for the true yield curve. In addition, the use of a trust region-based search procedure makes much better estimation for the curve fitted by RBFN.

4 Experimental Results

In this section, we verify how well the proposed method works compared with three existing fitting algorithms of the yield curve which include

- Cubic B-splines (Fisher's method; FM): [14]
- Nelson-Siegel curve (NS): [1], [2]
- MLP method (MM) : [4], [6], [8], [9]

The data used in the empirical analysis are 57 U.S. Treasury securities settled on November 3, 1997. U.S Treasury securities are semiannual-coupon bonds issued in order for U.S. Treasury Department to finance government borrowing needs at the lowest cost over time. The 57 securities have two quoted prices, bid price quotes and asked price quotes. We regard bid price quotes as real clean prices of the securities and use them in the empirical analysis. We used the spot rates bootstrapped from 50 of the 57 securities for training models, and the remaining 7 bonds for testing them. The kernel function for the proposed method is the Gaussian kernel function. The MLP method employed a multilayer-perceptron network instead of RBFN in *Phase I*.

Simulation results are shown in Table 1 and Fig 1. The fitting performance measures are the training error and the test error for bond pricing. Not surprisingly, NS, a parametric approach, provides the poorest performance because error is relatively severe. In effect, NS is used when good accuracy is not required. The proposed method shows a better performance compared to the others in terms of accuracy.

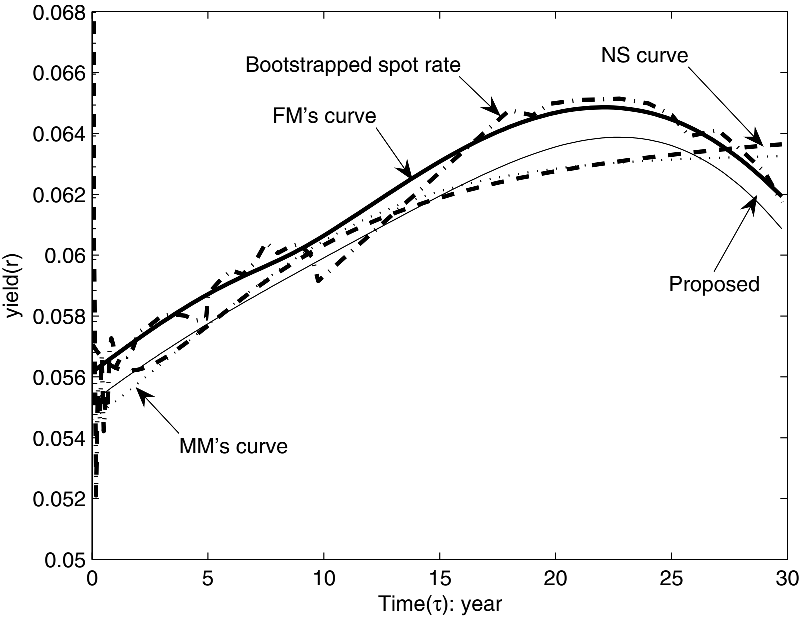


Fig. 1. Comparison of results for the fitting of the yield curve

Table 1. Simulation Result using four yield-curve fitting techniques. (b.p.(basis point)=0.01%)

Algorithm	Training Error	Test Error
NS	1438 b.p.	97 b.p.
FM	253 b.p.	73 b.p.
MM	1219 b.p.	80 b.p.
Proposed	287 b.p.	56 b.p.

5 Conclusion

In this paper, we've proposed a novel yield curve fitting algorithm to mitigate pricing errors and improve a generalization performance for estimating the yield curve. The proposed method consists of a RBFN-based initial curve-fitting phase and a trust region-based optimal yield curve search phase. In order to show the effectiveness of the proposed algorithm, we've conducted experiments with 57 U.S. Treasury securities. The experimental results demonstrate that the new algorithm not only successfully improves generalization but also has better pricing accuracy compared with the existing algorithms for the fitting of the yield curve. An application of the methods to other assets' yield curve estimation remains to be further investigated.

Acknowledgement

This work was supported by the Korea Research Foundation under grant number KRF-2004-041-D00785.

References

1. James, J., Webber, N.: Interest Rate Modeling. John Wiley & Sons Ltd, London (2000)
2. Nelson, C.R., Siegel, A.F.: Parsimonious Modeling of Yield Curves. The Journal of Business, **60** (1987) 473-489
3. de Boor, C.: A Practical Guide to Splines. Springer-Verlag, Berlin Heidelberg New York (1978)
4. Lee, D.-W., Choi, H.-J., Lee J.: A Regularized Line Search Tunneling for Efficient Neural Network Learning. Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg New York **3173** (2004) 239-243
5. Dierckx, P.: Curve and Surface with Splines. Oxford Science Publications, New York (1995)
6. Choi, H.-J., Lee, H.-S., Han, G.-S., Lee, J.: Efficient Option Pricing via a Globally Regularized Neural Network. Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg New York **3174** (2004) 988-993
7. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistcal Learning: Data Mining, Inference, and Prediction. Springer-Verlag, Berlin Heidelberg New York (2001)
8. Lee, J.: Attractor-Based Trust-Region Algorithm for Efficient Training of Multilayer Perceptrons. Electronics Letters, **39** (2003) 71-72
9. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice-Hall, New York (1999)
10. Lee, J., Chiang, H.-D.: A Dynamical Trajectory-Based Methodology for Systematically Computing Multiple Optimal Solutions of General Nonlinear Programming Problems. IEEE Transactions on Automatic Control, **49** (2004) 888-899
11. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer-Verlag, Berlin Heidelberg New York (1999)
12. Bliss, R.R.: Testing Term Structure Estimation Methods. Advances in Futures and Options Research, **9** (1997) 197-231
13. McCulloch, J.H.: The Tax Adjusted Yield Curve. Journal of Finance, **30** (1975) 811-830
14. Fisher, M., Nychka, D., Zervos D.: Fitting the Term Structure of Interest Rates with Smoothing Splines. Working Paper 95-1, Finance and Economics Discussion Series, Federal Reserve Board, Washington (1995)

Fast ICA for Online Cashflow Analysis*

Shangming Yang and Zhang Yi

Computational Intelligence Laboratory, School of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, Sichuan 610054, China
minn003@163.com, zhangyi@uestc.edu.cn

Abstract. Independent component analysis is an important tool for separating blind sources and also a well-known method of finding latent structure in data. With the observed data and some of their properties, one can find original sources and structures of these data. This can be used in data mining to discover useful information from the online data. In this paper, based on the Fast ICA algorithm, a model for chain stores' sales is set up. Using this model, we can analyze the distributions of the sold products from the online cashflow in each store of the chain; the cashflow is from the mixture of different products in a store. This algorithm will be very attractive for a commercial enterprise to provide information for making their future sales plan.

1 Introduction

Independent component analysis (ICA) [1, 2, 3, 4] model is utilized mainly in blind sources separation of unknown signals. It is a statistical and neural network method that explores a set of multidimensional observations as a combination of unknown variables. These unknown variables are called independent components which are assumed to be statistically independent of each other. The basic ICA model is

$$\mathbf{x} = \mathbf{f}(\mathbf{v}, \mathbf{s}) \tag{1}$$

where $\mathbf{x} = (x_1, ..., x_m)$ is the observed vector and f is a general unknown function with parameters $\mathbf{v}$ which are statistically independent latent variables listed in the vector $\mathbf{s} = (s_1, ..., s_n)$. Usually we use a special case of (1) in the research or practical application, which is a linear vector function:

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{2}$$

where $\mathbf{A}$ is an unknown $m \times n$ mixing matrix. In formulae (1) and (2), variables $\mathbf{x}$ and $\mathbf{s}$ are considered as random vectors. The mixing matrix $\mathbf{A}$ is constant for all observations, but in the practical application, it's not available, the only data we have is the observed vector $\mathbf{x}$. All vectors are column vectors. The

* This work was supported by National Science Foundation of China under Grant 60471055 and Specialized Research Fund for the Doctoral Program of Higher Education under Grant 20040614017.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 891–896, 2005.

linear ICA model (2) requires the following assumptions [1]: at most one of the independent components s_j can be Gaussian variable, and the matrix A must be of full column rank with $n = m$. Only if these two conditions are satisfied, independent components s_j in the linear model (2) can be estimated up to a permutation of their order and a scaling of their values.

ICA was originally introduced to deal with problems that are related to the blind source separation in cocktail-party problem [6]. Imagine that you are in a party where two people are speaking simultaneously. You have two microphones, which are held in different locations. This is the situation in formula (2) when $m = n = 2$. The microphones give you two recorded time signals, which are denoted by $x_1(t)$ and $x_2(t)$, with x_1 and x_2 the amplitudes, and t the time index. Each of these recorded signals is a weighted sum of the speech signals emitted by the two speakers, which we denote by $s_1(t)$ and $s_2(t)$. It would be very useful if you could now estimate the two original speech signals $s_1(t)$ and $s_2(t)$ by using only the recorded signals $x_1(t)$ and $x_2(t)$. it's clearly, this algorithm allows us to separate the two original source signals $s_1(t)$ and $s_2(t)$ from their mixtures $x_1(t)$ and $x_2(t)$.

In [7], ICA was applied to a financial problem: the cashflow of many different stores in the same retail chain, was used to analyze the fundamental factors common to all stores that affect the cashflow data. Thus, the cashflow effect of the factors specific to any particular store, i.e. a couple of different fundamental factors hidden in the original data could be clearly analyzed.

This paper concentrates on the data mining viewpoint of ICA. It tries to estimate the original data sources by the observed cashflow. Differing from the method in Oja's [7] , we will use the cashflow to estimate the distribution of the sold products at a given peiod. The results will be very helpful for sales management if we can obtain the correct estimations. In data mining, we know that different types of products may not be always independent, if they are strongly related, using association rules mining, one can find the relationship, then these products should be in the same class.

2 Association Rule Mining Concepts

In association rules mining [5], a set of items is referred as itemset. An itemset that contains k items, is a k-itemset. Suppose $I = \{i_1, i_2, ..., i_m\}$ is a set of items, D is a set of database transactions, where each transaction T is a set of items such that $T \subseteq I$. Let A be set of items. A transaction set T is said to contain A if and only if $A \subseteq T$. An association rule is an implication of the form $A \Rightarrow B$, $A \subset I$, $B \subset I$ and $A \cap B = \phi$. It has the following two significant properties:

$$support : (A => B) = P(AUB)$$

The probability of itemset D contains both itemset A and B

$$confidence : (A => B) = P(A/B)$$

The probability of itemset D contains A also contains B.

Association rules mining is the rule to generate support and confidence which are greater than or equal to their minimum support and confidence. For independent component analysis, the constraint is that the variables which represent the latent data are statistically independent, if some of the variables in the components are strong related, using ICA to separate the blind sources may lead to wrong conclusions. We use association rule mining to find those related variables that can be considered as the same class, and we therefore need to reduce the dimensionality.

3 ICA for Online Cashflow Analysis

For an enterprise that has hundreds of retail chain stores, their database systems are in a local or global network system. For the headquarter, the daily sales of all stores are always online available. Therefore, the cashflow for each store is observable. However, the detail sales for each type of product are hard to obtain, traditionally they need complex statistical work to analyze. It is important for a company to know which type of product has increasing sales, which one has decreasing sales and which one has stable sales in a period, the company needs this information to make purchasing plans for short and medium term future sales. Based on the following basic idea, we can build up a model for the data analysis to solve the problems:

a. For any two types of products, we assume product *a* and product *b* are independent, which means if a customer buys product *a* then she or he will not be necessary to buy *b* except that some products have to be sold in pairs. According to the association rules in data mining, if the support is not big enough, some products may have some relationship in their sales, but they are just some special cases. In general, they are statistically independent.

b. In some huge business area, for a given period of time, we assume that the total sale for products in a store is given, which only depends on the distribution of the customers; the sales will be different in different stores.

c. Assume there are enough stores for the observation. For example, if we have 100 types of different products to investigate, we must have 100 stores for the observations of the cashflow, so that we can separate the total sales to different classes of products. If there are too many classes for analysis, we can choose some products for online analysis, the only problem is, at a given period of time, the total sales of these products are online available.

From the description in *a*, *b* and *c*, we have the following sales model:

$$\begin{array}{l} x_1 = f_{11}s_1 + f_{12}s_2 + ... + f_{1n}s_n \\ x_2 = f_{21}s_1 + f_{22}s_2 + ... + f_{2n}s_n \\ ... \\ x_n = f_{n1}s_1 + f_{n2}s_2 + ... + f_{nn}s_n \end{array} \tag{3}$$

where $x_1, x_2, ..., x_n$ are n different observations of cashflow data, $s_1, s_2, ...s_n$ are the total sales for each of the n types of different independent products in the whole company, f_{ij} are the sale factors for different stores. For example, if there are four different types of products A, B, C and D for sale in a company and there are four stores in this company, in a given period of time, product A has \$1000 ($s_1 = 1000$) sale and in which store 1 has 10%, store 2 has 35%, store 3 has 30% and store 4 has 25%, these four percentages are f_{11}, f_{21}, f_{31} and f_{41} respectively, in the same way for the other f_{ij}. We consider this an ICA model because for a given total sale of product A, it is distributed to four different stores and for any customer, the actions a customer buys product A, B, C or D are independent, which means s_1, s_2, s_3, s_4 are independent. In ICA, f_{ij} and s_i are unknown variables. Using the online observations of x_i, we can estimate the distributions of s_i at a given time period.

FastICA is a faster method for maximizing negentropy, the resulting FastICA algorithm finds a direction vector $\mathbf{w}$, such that the projection $\mathbf{w}^T\mathbf{z}$ maximizing nongaussianity. Using Fast ICA to separate those independent components, usually it has two steps [8]:

a. Using eigenvalue decomposition of the covariance matrix for prewhitening data we have $E(\mathbf{x}\mathbf{x}^T) = \mathbf{E}\mathbf{D}\mathbf{E}^T$, $\mathbf{V} = \mathbf{E}\mathbf{D}^{(-1/2)}\mathbf{E}\mathbf{T}$ and $\mathbf{z} = \mathbf{V}\mathbf{x}$ $\mathbf{D}^{(-1/2)} = diag(d_1^{(-1/2)}, d_2^{(-1/2)}$,..., $d_n^{(-1/2)})$, $\mathbf{E}$ is the orthogonal matrix of eigenvectors of $\mathbf{E}(\mathbf{x}\mathbf{x}^T)$ and $\mathbf{D}$ is the diagonal matrix of its eigenvalues

b. Using Fast ICA to separate the variables, we have $\mathbf{y}$=$\mathbf{W}\mathbf{z}$, here $\mathbf{y}$ is the vector of the eventual separated variable, that is, the estimation of the original source vector $\mathbf{s}$. We denote any one column in $\mathbf{W}$ to be a vector $\mathbf{w}$, the final $\mathbf{w}$ for the estimation is generated from the following iterations

$$\mathbf{w} \leftarrow E(\mathbf{z}g(\mathbf{w}^T\mathbf{z})) - E(g'(\mathbf{w}^T\mathbf{z})\mathbf{w}) \tag{4}$$

where $g(x)$ may be chosen from the following three formulae: $g_1 = tanh(a_1 u)$, $g_2 = u \times exp(-a_2 u^2/2)$ and $g_3(u) = u^3$.

4 Simulations

To test the proposed model, we use a random generator to generate two group artificial data for the experiments. In order to test the model, we set up a group of percentages for f_{ij} so that we can decide the efficiency by comparing the original data and the estimates. In practice, the values for f_{ij} are not important, they are actually unknown variables in the analysis, the only thing we care about is whether the conditions for model are satisfied. From (3) and the randomly generated data we have, we obtain the mixed data, using Fast ICA, we can separate the mixed data. Fig. 1 is the distributions of two group of observed data in the simulations, Fig. 2 and Fig. 3 are the comparisons of the corresponding results, the left figures are the original data distributions and the right figures are their corresponding Fast ICA estimates. Comparing the distributions in Fig. 2 and in Fig. 3, we can see that the original data distributions are almost the same as the estimates, except in very few points the highest and the lowest points are delayed or brought forward. Non-negative ICA [9] may be also a good algorithm for this estimation but the convergence of the algorithm is not proved.

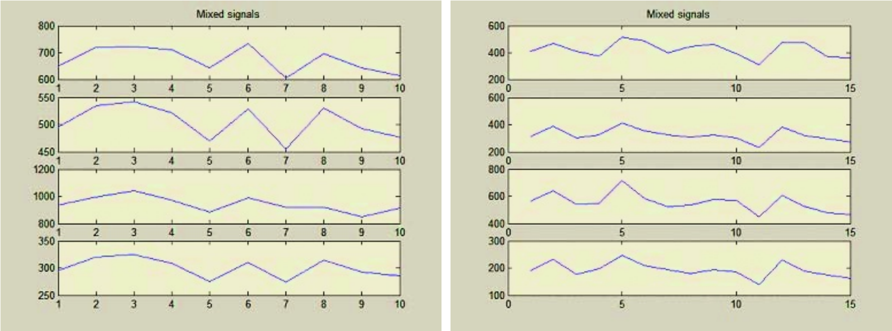


Fig. 1. Distributions of observed data group 1 and data group 2

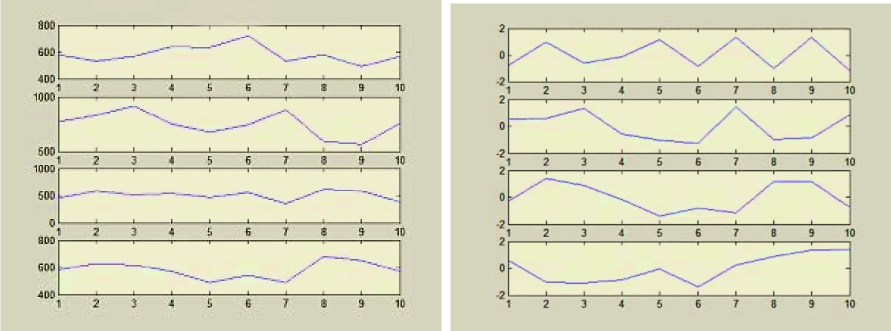


Fig. 2. Distributions of original data group 1 and separated data group 1

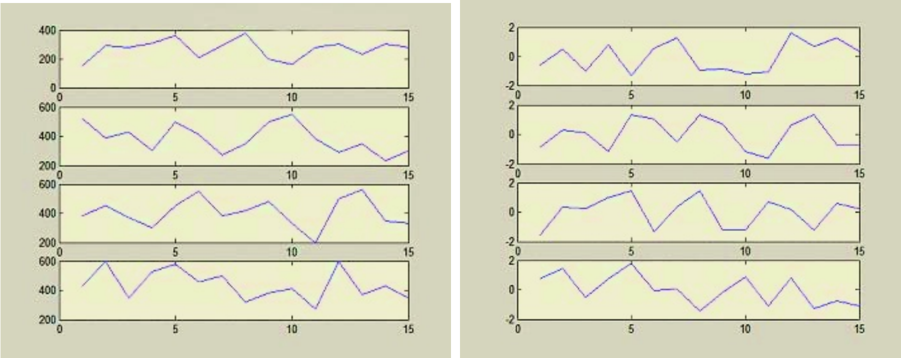


Fig. 3. Distributions of original data group 2 and separated data group 2

5 Conclusions

The basic ICA network consists of three parts, which are whitening, separation, and basis vector estimation layers. In this paper, this network is used to estimate the online sales distribution, which is significantly helpful for the sales management. From the distribution of the cashflow, we only know that in a given period, the sales are increasing, decreasing or unchanging, but we don't know which types of products have those changes. The results can be estimated by using the proposed algorithm. Simulations show that the this model is very effective. For the given data groups, each original distribution and its estimated distribution are extremely similar. The existing problem is from the estimates, the exact amount of sales are not available, this needs the further investigation.

References

1. Comon, P.: Independent Component Analysis a New Concept? Signal Processing, **36** (1994) 287, C314
2. Jutten, C., Herault, J.: Blind Separation of Sources, part I: An Adaptive Algorithm Based on Neuromimetic Architecture. Signal Processing, **24** (1991) 1, C10
3. Hyvarinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. Wiley Interscience (2001)
4. Bingham, E.: Advances in Independent Component Analysis with Applications to Data Mining. Dissertations in Computer and Information Science, Helsinki University of Technology, Espoo, Finland (2003)
5. Han, J., Kamber, M.: Data Mining – Concepts and Techniques. Higher Education Press, Beijing, China (2001)
6. Hyvarinen, A., Oja, E.: Independent component analysis: algorithms and applications, Neural Networks **13** (2000) 411, C430
7. Kiviluoto, K., Oja, E.: Independent Component Analysis for Parallel Financial Time Series. Proceedings of the International Conference on Neural Information Processing (ICONIP '98), Tokyo, Japan, **2** (1998) 895, C898)
8. Hyvarinen, A.: Fast and Robust Fixed-Point Algorithms for Independent Component Analysis. IEEE Transactions on Neural Networks, **10**, (1999)
9. Plumbly, M., Oja, E.: A "Nonnegative PCA" Algorithm for Independent Component Analysis. IEEE Transactions on Neural Networks, **15**, (2004)

Impacts of Internet Stock News on Stock Markets Based on Neural Networks

Xun Liang[1,2]

[1] Institute of Computer Science and Technology, Peking University, Beijing 100817, China
liangxun@icst.pku.edu.cn
[2] Department of Management Science, Stanford University, Palo Alto, CA 94305, USA

Abstract. The research of impacts between the Internet stock news (ISN) and the stock price movements emerges with the era of the popular usage of the Internet. In this paper, first, the ISN is empirically related to the stock returns based on the neural networks. Second, the relationship between the significant increases of the ISN volume and the significant changes of stock prices is addressed, also implemented by neural networks. Experiments demonstrate the probable relations, hinted by the neural networks. The results are helpful in probing the microstructure of the stock markets.

1 Introduction

A stock market is a very complex system. Investors frequently rely on the stock news to trade [2]. The relations between the stock information and the stock prices have been an interesting subject and studied many years [1],[8,][9].

In the era of Internet, the webs quickly become one of the most important media to convey the stock information. A huge amount of news is published on the Internet each day. In recently years, more and more scholars show increasing interests to study the impacts of Internet stock news (ISN) on the stock markets [1],[9].

The paper explores the impacts of ISN and stock markets from two new aspects.

First, we discuss the impacts on the stock returns by a given ISN. As the impacts of stock information on stock markets have been studied extensively in the past, the novel point in our study is that the Internet privileges are taken into account. It was announced by the W3C (World Wide Web Consortium) that in the next generations of webs, XML, instead of HTML, will be used in the webpages. Based on this protocol, the web browsers like the Microsoft's Internet Explorer should not only process the style and content of information as currently, but also convey the meanings through XML tags. For example, <IntensityImpacted>0.5</IntensityImpacted> shows that the intensity impacted by the ISN on the markets is calibrated as 0.5. We assume that in the next generation of webpages, all the ISN published by the web servers of large Internet media will be incorporated in some tags like in the above showing the subjective or predictive intensity of the ISN, the range of investors that the news impacts, etc, although the news in the tags may be designed invisibly to the normal webpage readers, but visibly to the computer programs.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 897–903, 2005.

For convenience, we call the information in the XML tags the header information or simply the headers. The headers allow computers for quicker processing. Noticing that the important stock news in the Internet duplicates the important stock news on the television and newspaper, it is optimistic that the important ISN may also be strongly correlated with the stock markets. Consequently, the study of ISN with respect to the stock prices is worthwhile also. Needless to say, the neural networks are a good option to mine the relationship between the headers and stock returns.

Second, the Internet also offers us possibility to find the volume of its stock news each day. It is the tedious work to enumerate the stock news on newspaper and television each day. As a result, in the past the research on the relations between the volumes of ISN and the stock prices remains almost blank. With the help of computers, the enumeration of ISN is as easy as any other work done by computers. There is not so much literature currently on this topic. The work of [9] shows that the overnight stock message posting volume is predictive to the change of stock returns, and hence demonstrates the impacts of Internet stock messages posted on web per unit time on the stock markets.

Largely, the study in this paper is in the area of financial time series analyses where plenty of results have been archived, including those based on neural networks. The novel part in this paper is that all the stock information is from the Internet. In addition, the model of neural networks is not simply used. Instead, the financial news on the Internet is pre-processed first. As a result, the neural networks can work with their advantages in the applications.

The remaining of this paper is as follows. Section 2 explores possible mappings from the ISN to stock returns. Section 3 illustrates mappings from the volumes of ISN to stock prices. Section 4 concludes the paper.

2 Impacts of ISN on Stock Returns

On the server side in the next generation of webpage protocols, when the stock news is uploaded to the web server, the stock news contributor should subjectively provide 5 reader-invisible headers according to her expertise and prediction to the impacts of the stock news that she is uploading, DurationImpacted, IntensityImpacted, CirculationRange, StockMarketStatus, and IndustryStatus. For the given news, the 5 headers are scaled as follows. The duration that the news impacts is scaled into the interval [0,1], the intensity of the news into [-1,1] and the range of investors that the news impacts into [0,1], respectively. Also, the status like the bear or bull market and the status in the corresponding industry are marked with numbers in [-1,1]. The above news is put into the header of each news, represented by 5 XML tag fields. It is admitted that the headers are subjective and so to some extent the results would depend on how they are rated manually. However, since we only want to estimate the positive or negative directions of the stock movement, it is sufficient for our purpose if the headers could approximately indicate the positive or negative sentiment of the news. In addition to the headers, the other regular information like time, title, author and text, is uploaded in the same way as before.

On the client side, a computer harvester tirelessly collects the stock news. Of course, the harvester should collect from many financial news web servers like stated in the above. Our data were harvested from the principal stock webs finance. zgjrw.com, www.sic-cfi.com, www.cfi.net.cn, www.cnlist.com, www.chinabond.com. cn, www.cnfund.cn/jjcs.asp, finance.sina.com.cn, www.sse.com.cn, www.szse.cn, www.hexun.com, www.cs.com.cn, quote.stock.163.com/stock/index, www.pa18.com, www.cnfxj.com, www.boyaxun.com, and www.stock2000.net. The stock news from the above websites is in no way complete since the Internet contains oceans of stock news. However, since the important news duplicates each other, the websites listed in the above are adequate sufficient for conducting experiments.

Totally, 236 pieces of news were collected in our study. The data are divided into 2 sets. The training pattern set consists of 182 pieces of news and the testing pattern set consists of 54 pieces of news. The stock returns during the corresponding period are also collected for observations.

We only consider the moving directions of the stock prices.

The data pre-processing should be done before used in training neural networks.

First, consider the following two numbers, the return on the day when the news occurs r_0 and the return in the next day after the news occurs r_1. Which represents the impacts by the ISN? We compare r_0 and r_1, and adapt the one with bigger absolute value. For convenience, we write it as r. If $r = r_0$, according to the assumption of semi-strong form of efficient market hypothesis in the financial theory [7], the ISN may influence the markets even before it is announced. If $r = r_1$, it explains that the impacts of the news may have some psychology delay in investors. After all, the largest-impact-occurring time is still unknown, unpredictive and mysterious to scholars, and deserves our further study. Second, we transfer the return r to s = sign(r), namely we only use the directions of stock price movements. Again, the reason is that we can probably announce the news is the good news or bad news, but we cannot predict precisely the intensity of its impacts on the markets. Taking into account that the stock markets are not solely influenced by the news in the Internet or even by the stock news on the newspaper and television, our efforts to elaborate at the moment is invalid.

The mapping between <DurationImpacted>, <IntensityImpacted>, <CirculationRange>, <StockMarketStatus>, <IndustryStatus> and stock returns is realized by a 5-H-1 feedforward neural network. The training starts from a network with H=30. If the training is trapped into local minima, H is increased by one and the new network with the expanded architecture is retrained. After the network has been trained, the superfluous hidden neurons are removed based on the orthogonal projection approach [4][5]. Finally, we reach the network with H=43. Among the 54 testing data, the trained neural network correctly predicts 33 signs of returns, accounting for 61% of the total. In 24 cases with relatively larger intensities, the correct rate is 17/24 = 71%. In other 3 cases with very large intensities, the correct rate is 3/3 = 100%.

It is concluded (1) the signs of stock returns, or the directions of stock price movements, on the day or the following day of the corresponding ISN, tend to be predictive based on the neural networks, and (2) the news with larger intensity is more likely to predict the market movement than the average.

3 Mapping Between Significant Increases of ISN and Significant Changes of Stock Prices

In this section, we try to relate daily volumes of ISN to stock prices using neural networks. As there are not many papers on the volumes of ISN in the financial literatures, there are even fewer results based on neural networks.

Without loss of generality, we study company *i*, *i* = 1, …, *I*, where *I* is the number of companies we study on the stock markets. First, we pre-process the volumes of ISN. The volume of ISN on day *k*, denoted by $v_i(k)$, is defined as the number of pieces of ISN on the sub-website of company *i*, *k* = -*K*, … , -1, 0, where *K* is the observation period, e.g., *K* = 60. Clearly, $0 \le x_i(k)$. The mean of $v_i(k)$ for company *i* is $\mu_i = \frac{\sum_{k=-K}^{-1} v_i(k)}{K}$, and the standard deviation is $\sigma_i = \sqrt{\frac{\sum_{k=-K}^{-1} (v_i(k) - \mu_i)^2}{K-1}}$, *i* = 1, … , *I*.

The usage of values in *K* days incorporates more news into the measure and hence should result in better predictions.

From the markets, it is observed that when the ISN for a company changes little, the stock price for the same company is relatively placid; and when the ISN changes significantly, the stock price is also significantly oscillatory. Consequently, it seems to us that it is easier to establish the relationship between the significant increases of on the daily volumes of ISN and the significant changes of the stock prices than the one between the ISN and the stock prices directly.

If v_i on day *k* satisfies $\mu_i + 2\sigma_i \le v_i$, we call that a significant increase on the volumes of ISN occurs for company *i* on day *k*. The magnitude of the volume of ISN is $\theta_i = \max\left\{\frac{v_i - \mu_i}{\sigma_i}, 0\right\}$. θ_i includes more past information than the simple $\frac{v_i(k+1)}{v_i(k)}$.

Let $p_i(k)$ be the stock price for company *i* on day *k*, $\Delta p_i(k) = p_i(k) - p_i(k-1)$ the stock price change on day *k* for company *i*, *k* = -*K*, … , -1, 0. The mean is $\lambda_i = \frac{\sum_{k=-K+1}^{-1} |\Delta p_i(k)|}{K-1}$, and the standard deviation is $\tau_i = \sqrt{\frac{\sum_{k=-K+1}^{-1} (|\Delta p_i(k)| - \lambda_i)^2}{K-2}}$, *i* = 1, … , *I*. If the absolute value of stock price change $\Delta p_i(k)$ on day *k* satisfies $\lambda_i + 2\tau_i \le |\Delta p_i(k)|$, we call that a significant change of the stock price occurs for company *i* on day *k*.

The input pattern vector is $X_i(k) = (\theta_i(k+1), \theta_i(k), \theta_i(k-1), \ldots, \theta_i(k-L+1), |p_i(k)|, |p_i(k-1)|, \ldots, |p_i(k-L+1)|)^\tau \in [0,1]^{(2L+1)\times 1}$. There is only one element in the output pattern $|p_i(k+1)| \in [0,1]^{1\times1}$. Using the older data as the input patterns is a common practice in neural network applications [6]. The parameter L in the input patterns is selected experimentally. If L is too small, it may not convey enough older news; if L is too large, it leads to unnecessarily large number of input ports and makes the network superfluous. We have tried L from 5 to 10, meaning that data of past 5 trading days to 10 trading days are used in input ports. It appears to us that $L = 9$ gives the best results.

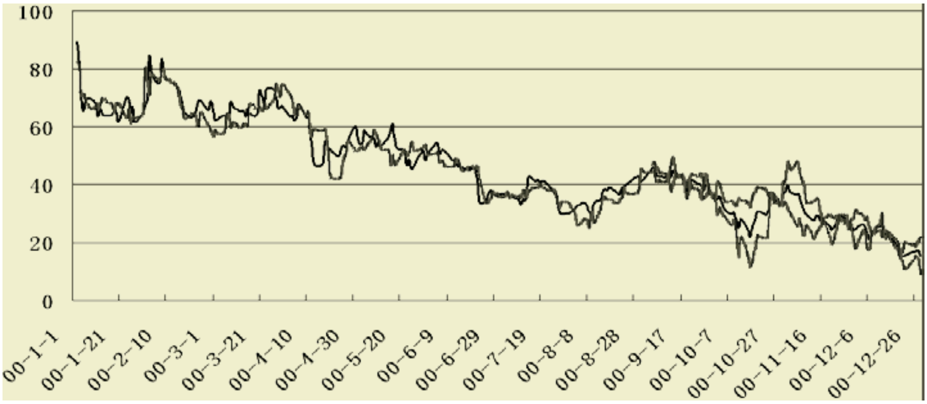


Fig. 1. The stock price movements for the firm AMZN. The dark line is the actual stock price movement. There are two segments for the shallow lines. From 2000-1-1 to 2000-8-31, the data are training patterns. The shallow line shows the price movement given by the trained neural network. From 2000-8-31 to 2000-12-29, the data are testing patterns. There are two shallow lines. Since the neural network only predicts the absolute value of the stock price change $|p_i(k+1)|$ on the testing day $k+1$, the price that the neural network hints is the prices $p_i(k) \pm |p_i(k+1)|$. As a result, the two shallow lines are symmetric to the real stock price movement. The prices are in US dollars

The ISN was harvested from the websites including Yahoo.com, Ragingbull.com, and Smartmoney.com. Our harvesters have collected all the ISN for all the companies on the NYSE and NASDAQ markets each day and saved in the database. In the experiments, the overnight ISN is divided into two parts by 5:00pm EST according to the time posted on web. The volumes of ISN posted on the webs in the nights and the non-trading days are added to the next trading day.

From Yahoo's website, for each company i, we download the .csv files containing the closing prices for the year 2000. In our simulation, the number of companies is $I = 400$. The original training patterns are the data from 2000-01- 03 to 2000-08-31.

After the neural network is trained, we predict $|p_i(k+1)|$ day by day. For example, on the testing day 2000-09-01 (for simplicity we write k+1=2000-09-01), we know $\theta_i(k)$, $\theta_i(k-1)$, … , $\theta_i(k-L+1)$, $|p_i(k)|$, $|p_i(k-1)|$, …, $|p_i(k-L+1)|)^\tau$ and also $\theta_i(k+1)$ on the testing day. Hence $X_i(k)$ is known and $|p_i'(k+1)|$ is computed by the trained neural network. In the next trading day, let $k=k+1$ (and the next trading day k+1=2000-09-

05). No matter the network predicts correctly or not, the real $|p_i(k+1)|$ as opposed to the predicted one is 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 neural network is assigned the work to learn the mapping. If it cannot learn the patterns, a new hidden neuron is added. When the mapping is learned, some hidden neurons are removed based on the orthogonal projection approach [4],[5]. Finally, the neural network with $H = 14$ is obtained.

The following public companies from NYSE and NASDAQ are used in experiments, AFCI, AMZN (see Fig.1), AT, ATK, BRCM, CC, CCUR, CNET, EBAY, HLTH, INFY, ISON, JDSU, LMIN, LVLT, MLS, MOLX, ORCL, PCLN, PSFT, QCOM, QUOT, RCOM, SIRI, SNDK, TXN, UCOMA, WLP. It appears to us that there exists some relationship between the volumes of ISN and stock price movements. The results are consistent with the work that the overnight stock message posting volume is predictive to the change of stock returns [9]. The experiments demonstrates that the training errors are less than 12% for all the public companies listed above, and the testing errors are less than 28%.

4 Concluding Remarks and Future Research

It is found that the contents of the ISN are related to the stock returns based on the XML protocols. Also, the volume of the ISN is associated with the stock price movements. From both aspects, our study illustrates the relationship between the two.

The results in this paper are preliminary and more studies should be done. For example, the probabilities should be introduced and the correlation coefficients between the ISN and the stock price could be computed accordingly. The Bayesian probability network would refine the result in this paper.

The ISN is becoming a new indicator for stock price movements. For the stock market professionals, it is beneficial for them to know the ISN as a new index to observe the undercurrents of stock markets. As a typical eFinance multidisciplinary subject, the work based on the powerful neural networks is beneficial to explore the micro-structure of the stock price movements from a new angle.

References

1. Das, S. R., Chen, M. Y.: Sentiment Extraction from Small Talk on The Web. Technical Report, Santa Clara University (2004)
2. Grundy, F., Bruce, D.: Trade and the Revelation of News Through Prices. Review of Financial Studies, **2** (1989) 495-526
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.: A Study of Removing Hidden Neurons in Cascade-Correlation Neural Networks. Proc of Int Joint Conf on Neural Networks Budapest ,**2** (2004) 1015-1020

5. Liang, X, Xia, S.: Methods of Training and Constructing Multilayer Perceptrons with Arbitrary Pattern Sets. Int Journal of Neural Systems, **6** (1995) 233-247
6. Narendra, S., Parthasarathy, K.: Gradient Methods for the Optimization of Dynamical Systems Containing Neural Networks. IEEE Trans on Neural Networks ,**2** (1991) 252-262
7. Keane, S. M.: Efficient Market Hypothesis and the Implications for Financial Reporting. Van Nostrand Reinhold New York (1983)
8. Titman, S.: Security Analysis and Trading when Some Investors Receive News Before Others. Journal of Finance, **49** (1994) 1665-1698
9. Wysocki, P. D.: Cheap Talk on the Web -Determinants of Postings on Stock Message Boards. Technical Report, University of Michigan (1999)

Coherent Risk Measure Using Feedfoward Neural Networks

Hyoseok Lee[1], Jaewook Lee[1], Younggui Yoon[2], and Sooyoung Kim[1]

Department of Industrial and Management Engineering
Pohang University of Science and Technology
Pohang, Kyungbuk 790-784, Korea
{ronalee,jaewookl,sookim}@postech.ac.kr
[2] Department of Physics, Chung-Ang University, Seoul, 156-756, Korea
yyoon@cau.ac.kr

Abstract. Coherent risk measures have recently emerged as alternative measures that overcome the limitation of Value-at-Risk (VaR). In this paper, we propose a new method to estimate coherent risk measure using feedforward neural networks and an evaluation criterion to assess the accuracy of a model. Empirical results are conducted for KOSPI index daily returns from July 1997 to October 2004 and demonstrate that the proposed method is superior to the other existing methods in forecasting the conditional expectation of losses beyond the VaR.

1 Introduction

The market risk of a portfolio refers to the possibility of financial loss due to the joint movement of systematic economic variables such as interest and exchange rates. To quantify the market risk is important to regulators in assessing solvency and to risk managers in allocating scarce capital. To allocate capital, assess solvency, and measure the profitability of different business units, managers and regulators quantify the magnitude and the likelihood of possible portfolio value changes for various forecast horizons. This process is often referred to as "measuring market risk", which is a subset of risk management functions [2], [8].

Since 1990s, Value-at-Risk (VaR) has been used as the most useful index to measure market risk [5]. As markets is becoming more volatile, and sources of market risk is proliferating, a need for institutions to develop more sophisticated VaR measures had increased. However, VaR has been criticized for the limitation that it cannot give any information on the loss greater than the measured VaR values nor correctly aggregate risk across portfolio [3]. As a substitute for VaR, Artzner, Delbaen, Eber, Heath(ADEH, 1999) proposed a new class of coherent risk measure by specifying some properties for a reasonable risk measure to have. Conditional-Value-at-Risk (CVaR), which satisfies the properties of a coherent risk measure, has recently emerged as alternative measure [1], [10].

In this paper, we propose a method to estimate VaR and CVaR using a neural network approach to enhance predictive power by directly estimating volatility and utilizing it to obtain more accurate estimate for VaR and CVaR.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 904–909, 2005.

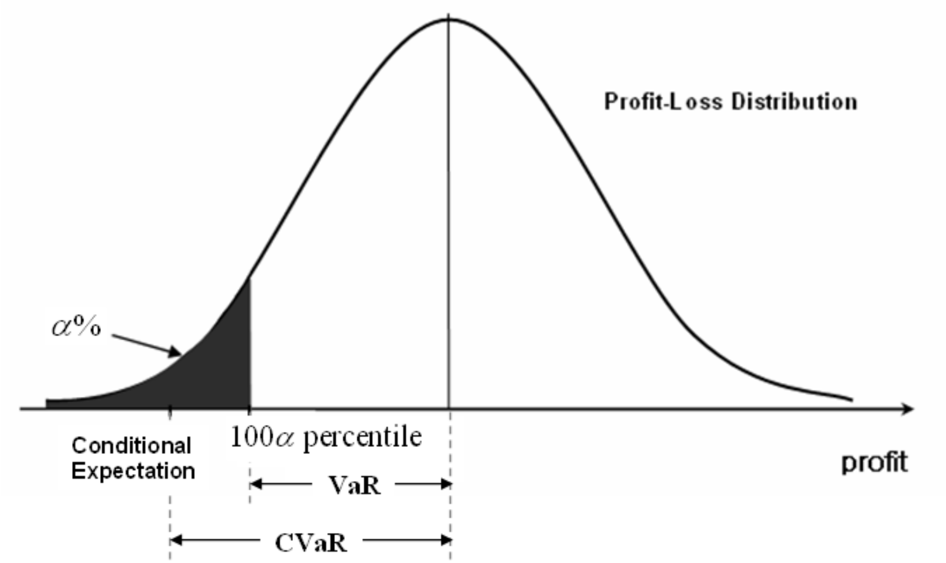


Fig. 1. VaR and CVaR

The performance of the proposed method is verified concerned with improving VaR and CVaR estimation by applying it to daily returns of KOSPI index from July 1997 to October 2004.

2 VaR and CVaR

A return series is computed via the first difference of the log-transformed prices

$$r_t = \log(p_t) - \log(p_{t-1}), \tag{1}$$

where p_t is the value of stocks at time t and r_t is the profitability in log scale at time t. The measurement of market risk has focused on one particular metric called VaR, which is defined as a on-sided confidence interval on the portfolio losses:

$$\mathrm{Prob}[r_t > -\mathrm{VaR}] = 1 - \alpha \tag{2}$$

The VaR_α of the loss is the value

$$\mathrm{VaR}_\alpha = \min\{\mathrm{VaR}|\mathrm{Prob}[r_t > -\mathrm{VaR}] \geq \alpha\} \tag{3}$$

where the parameter α is the confidence level. The interpretation is that, over a large number of trading days, the value of a bank's portfolio will decline by no more than VaR $\alpha\%$ of the time. If we use the Delta-Normal method, VaR with α confidence interval can be written

$$\mathrm{VaR}_\alpha = \alpha \mathrm{W} \sigma \tag{4}$$

where W is the wealth, and σ is the standard deviation of return distribution.

The CVaR_α of the loss associated with a decision x is the value

$$\mathrm{CVaR}_\alpha = E[-X| - X \geq \mathrm{VaR}_\alpha(X)] \tag{5}$$

Therefore, CVaR is understood as conditional expectation value of loss when the loss is above VaR. Since both VaR and CVaR are based on the forecast on the portfolio value distribution, the method to estimate a volatility plays an important role.

3 Estimation of Volatility

3.1 Previous Works

Although the value r_t^2, the square of profitability, is an unbiased estimator of volatility, GARCH and EWMA are widely used to obtain more accurate volatility and to reflect the past data, which are described by

$$\begin{aligned} &\mathrm{GARCH} : \sigma_t^2 = K + G_1 \cdot \sigma_{t-1}^2 + A_1 \cdot \epsilon_{t-1}^2 \\ &\mathrm{EWMA} : \sigma_t^2 = \lambda\sigma_{t-1}^2 + (1-\lambda)r_t^2 \end{aligned} \tag{6}$$

The basic idea of EWMA is setting smoothing constant to be smaller than 1, e.g., 0.94 or 0.97, giving more weight in the values near the present than in the past. The merit of EWMA is that the calculation is easy. However, there is arbitrariness in the choice of trials because the method does not provide theoretical background in determining the smoothing constant. The GARCH model is widely used in analyzing financial time series because the model can easily represent the centralization of volatility which is observed in financial market frequently and it reduces the number of trials to estimate compared to the ARCH model.

3.2 Proposed Approach Using Neural Networks

The basic idea of the proposed method is to train a neural network through the past volatility data to forecast future volatility, thereby obtaining a VaR or CVaR. The following formula represents our proposed method to estimate volatility.

$$\sigma_{t+1}^2 \cong r_{t+1}^2 = f(r_t^2, r_{t-1}^2, \cdots, r_{t-s}^2, \theta) + \epsilon_t \tag{7}$$

The neural network structure is trained with the square of profitability of past s-days as its input and the square of profitability as its output. Note that the proposed method to estimate volatility estimates future volatility reflecting the change in profitability until today, while other established methods such as GARCH and EWMA make use of past and present profitability data to estimate present volatility. Moreover, this point is distinguished from other neural network approach that employs mixture density model to obtain VaR [9].

4 Performance Evaluation

Recently, Basle Committee announced a new recommendation that financial institutions calculate a BIS required capital based on VaR, obtained internally from models, and that sanctions are invoked afterwards based on verifying the appropriateness of the VaR model. Basle Committee regulates that the following is applied equally to the *post priori* verification

- Models are verified based on the number of failures
- The confidence level is 99 % equally.
- *Post priori* verification period is based on a quarter and recent data of 250 business days corresponding to 12 months is used.

Green zones, yellow zones, and red zones are determined based on the number of failures exceeding VaR internally calculated from models according to those common criteria. Basle institution calculates the required capital according to the following formula selecting the greater of the VaR of the past 60 days and VaR of yesterday. Larger stabilizing exponents, k, lead to larger required capitals because the VaR of the past 60 days is adopted in most cases.

$$\text{Required Capital} = \max\left(k\frac{1}{60}\sum_{i=1}^{60} 10\text{day}VaR_{t-i}, 10\text{day}VaR_{t-1}\right) \tag{8}$$

$$k = 3 + \alpha, 0 \geq \alpha \geq 1. \tag{9}$$

In this paper, we assess the accuracy of the model by examining the failure rate of the model based on the Basle regulators. The failure rate is the proportion of times the VaR figure is exceeded in a given sample. At a given moment, a VaR number is specified when a certain confidence level c for a total of T observations is given and N is defined as the number of exceptions. One statistic for this test is the Bernouilli trials test, which is used for testing sequences of success and failure.

In addition to this evaluation criterion based on failure rate or number of failures for VaR, in this paper, we evaluate the performance of a method by statistically measuring the differences between expected loss and actual loss for CVaR, which is a matter of primary concern. For example, CVaR_1, the average loss exceeding VaR obtained in the first quarter, is an expected loss in next quarters. Therefore, $\bar{X}_1$, the average loss exceeding VaR in 1^{th} quarter, should be the expected loss approximately. We propose a t-test method, verifying the utility of CVaR_i according to the loss-based VaR. The actual loss is evaluated using p-value of the t-test. Null hypothesis and T-statistics is following.

$$\text{H}_0 : \sum_{i=1}^{n} \left(\bar{\text{X}}_i - \text{CVaR}_i\right) \tag{10}$$

$$T = \frac{\sum_{i=1}^{n}\left(\frac{\bar{\text{X}}_i - \text{CVaR}_i}{n}\right)}{s/\sqrt{n}} \sim t_{n-1} \tag{11}$$

Small absolute T-statistics value indicates small difference between expected loss and actual loss, implying an accurate model.

5 Empirical Results

This study assumed that assets are stock portfolio with β equal to 1. The profitability is assumed to that of Korea Composite Stock Price Index (KOSPI). The position of portfolio is assumed to be fixed in period of the study and δ-normal analysis is performed. Data are taken from daily closing KOSPI indices in the period of July 1, 1997 until October 22, 2004 where the period includes IMF bailout. The VaR analysis method is verified in the presence of large fluctuation of the financial market. The architecture of a neural network model for the proposed method is '10-10-1' and model training period is set to be 3 years (750 business days) and the *post priori* verification period is set to be 1 year (250 business days) according to the recommendation of Basle Committee that VaR is measured quarterly. Our experiment has been repeatedly performed with recently developed learning methods in [4], [6], [7] with the verification period divided into 7 quarters.

Table 1. Failure number of each models

	1st qt	2nd qt	3rd qt	4th qt	5th qt	6th qt	7th qt
Confidence level				EWMA			
95%	4	18	8	15	5	11	2
99%	1	6	3	4	1	2	0
Confidence level				GARCH			
95%	4	6	8	9	4	7	1
99%	1	2	3	2	1	0	0
Confidence level				Neural Networks			
95%	4	2	7	6	5	4	3
99%	1	1	1	1	0	0	0

Table 2. T-Statistics (difference between expected loss and actual loss) of each model

Method	EWMA	GARCH	Neural Networks
T-statistic	-1.03	-0.723	0.072

Table 1 summarizes the number of failures in each model. Overall number of failures is the smallest in the neural network model, and that of GARCH is smaller than that of EWMA. Concerning the evaluation of expected loss and actual loss (T-statistics), the neural network model leads to the smallest absolute T-statistics value among three methods, which indicates that the neural network model is superior to the other methods in terms of actual loss.

6 Conclusion

In this paper, we have proposed a novel volatility estimating method using neural network to calculate the CVaR. To demonstrate potential and predictive power of the proposed method, we've used the daily closing KOSPI indices in the period of July 1, 1997 until October 22, 2004. Experimental results indicate that EWMA is the best among the EWMA, the GARCH, and the neural network methods in terms of the number of failures. However, the results shows that the proposed method using neural networks is the best among three in terms of the number of failures as well as in terms of the T-statistics of the expected loss and the actual loss.

Acknowledgement

This work was supported by the Korea Research Foundation under grant number KRF-2004-041-D00785.

References

1. Coleman, T.F., Alexander, S., Li, Y.: Minimizing CVaR and VaR for a Portfolio of Derivatives. The 3rd International Workshop on Optimization and Control with Applications, Chongqing Chengdu Jiuzhaigou, China (2004)
2. Patton, A.J.: Modelling Time-Varying Exchange Rate Dependence Using the Conditional Copula. Working Paper, University of California, San Diego
3. Artzner, P., Delbaen, F., Eber, J.-M., Heath, D.: Coherent Measures of Risk. Mathematical Finance, **9** (1999) 203-228
4. Lee, D.-W., Choi, H.-J., Lee J.: A Regularized Line Search Tunneling for Efficient Neural Network Learning. Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg New York **3173** (2004) 239-243
5. Holton, G.A.: Value-at-Risk: Theoy and Practice. Academic Press, San Diego (2003)
6. Choi, H.J., Lee, H.S., Han, G.S., Lee, J.: Efficient Option Pricing via a Globally Regularized Neural Network. Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg New York **3174** (2004) 988-993
7. Lee, J.: Attractor-Based Trust-Region Algorithm for Efficient Training of Multilayer Perceptrons. Electronics Letters, **39** (2003) 71-72
8. Pearson, N.D.: Risk Budgeting: Portfolio Problem Solving with Value-at-Risk. Wiley Finance, New York (2002)
9. Locarek-junge, H., Prinzler, R.: Estimating Value-at-Risk Using Neural Networks. Informations Systeme in der Finanzwirtschaft (1998)
10. Rockafellar, R.T., Uryasev, S.: Conditional Value at Risk for General Loss Distributions. Journal of Banking and Finance, **26** (2002) 1443-1471

Application of Evidence Theory and Neural Network in Warning System of Financial Risk

Qingyu Xiong[1,2], Yinlin Huang[1], Shan Liang[1],
Weiren Shi[1], Songsong Tan[1], and Yinhua Lin[1]

[1] Automation Department of Chongqing University, Chongqing 400044, China
bearqingyu@yahoo.com, hyl-cqu@163.com,
lightsun@cqcnc.com, wrshi@cqu.edu.cn

[2] The Key Lab of High Voltage Engineering & Electrical New Technology of
Education Ministry of China, Chongqing 400044, China
xiong03@cqu.edu.cn

Abstract. In warning system of financial risk, there are different relations between financial risk and factors. So the traditional prediction method can't get the accurate result. Based on the theory of economical forecast and warning, this paper gives a new method that uses neural network to estimate deterministic factors and experts to forecast uncertain factors, to get the ideal result by evidential combination of the predictions come from the certain and uncertain factors.

1 Introduction

The financial risk is that the different factors in financial field result in financial fund and regulation destroyed because of their comprehensive influence. It's important for the economical stability.

The financial forecast & warning is to analyze different potential risks systematically and identify which risk should be dealt with. For this purpose, there are lots of methods used in this field, such as grey forecasting method [1], regressive analytic approach [2], time series analytic approach [3], artificial intelligent method [4], etc.

The main problem in financial forecast & warning is that, there are lots of factors influencing finance, each of which has a various relationship with finance. But people usually choose these factors quantified easily. This will lead to unsatisfied predicting result. On the other hand, there exists some uncertain factors (such as policy, politics, economy, climate conditions) which are very difficult to be quantified [5].

According to the above, based on the background of local financial risk, this paper proposes a new prediction method with evidence theory and neural network.

2 Evidence Theory

The evidence theory, also known as the Dempster-Shafer theory, can sum up and estimate the probability of the existential state of each part in system by kinds of information, then make correct decision and prediction.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 910–915, 2005.

In case of a problem, all possible results of this problem are expressed by set Θ. Each proposition is corresponding to one subset in Θ, which depends on our knowledge and level of understanding, also called frame of discernment.

One evidence can offer support to one or many propositions (corresponding to one or many subsets of Θ). It can be expressed by BPA(Basic Probability Assignment).

Definition. Θ is the frame of discernment, m is a function from 2^{Θ} to [0,1] verifying

$$\begin{cases} \mathrm{m}(\phi) = 0 \\ \sum\limits_{A \subseteq \Theta} \mathrm{m}(A) = 1 \end{cases} \tag{1}$$

m is called a basic probability assignment. Then we can define a belief or credibility function that associates with m.

The Dempster Combinative Principle can reflect unite and effect of various evidences. Firstly, to choose two basic probability assignments m_1 and m_2 without correlation, let A_i and B_j being focal elements of m_1 and m_2, where *i* =1, 2, ..., *k* and *j*=1, 2, ..., *l*, $A_i, B_j \subseteq \Theta$, then m_1 and m_2 are combined to obtain the integrate m committed to $A \subset \Theta$ according to the following combination or orthogonal sum formula [6].

$$m(A) = \begin{cases} \dfrac{\sum\limits_{A_i \cap B_j = A} m_1(A_i) \cdot m_2(B_j)}{1 - \sum\limits_{A_i \cap B_j = \Phi} m_1(A_i) \cdot m_2(B_j)} & A \neq \Phi \\ 0 & A = \Phi \end{cases} \tag{2}$$

If there are more evidences, one can use the above principle to produce new BPA.

3 Applications

We use BP network as the basic method of prediction in determinacy factors [7]. The weights are set by BP algorithm. There exists several stages as folows:

1) Initializing, to initialize each layer's weight W and deviation B with little random number and guarantee not to get great weight.
2) Expressing variable, to calculate the output of vector in each layer and network error *E*.
3) Revising weight, to calculate the errors from different layers and get new weights.
4) Calculating error square, if it is bigger than *SSE*, to continue training.

Here, we combine the data of a local finance from 2002 to 2003 and adopt method mentioned above to predict the influence of deterministic and uncertain factors.

3.1 The Influence of Deterministic Factors

Based on the target of local financial warning and certain factors especially in economic fluctuation orbit, prediction in economy, economy developing trend etc. So, the following index system used includes 5 classes and 10 indexes [8].

1) Output. whole industrial output in district (C_1), assets increasing rate of regional economy (C_2).
2) Circulating. retail sales in social consumption(C_3), main industrial sales in district(C_4),
3) Finance. Fiscal revenues of district (C_5), balance of saving in bank (C_6), balance of loan in bank (C_7).
4) Investment. stable produce endowment in district (C_8), root found fulfill (C_9).
5) Living. resident's consumption index (C_{10}). Each index's Risk Coefficient is set by the analytic approach of the level. (See.Tab.1).

The prediction of every economic index adopts BP neural network method. The step is that we move the historical data of each economic index in time series to train neural networks, give input, and then get the prediction. The import layer of BP neural work has 4 neurons that are corresponding with 4 months data in series. And the export layer has a neuron that is corresponding with the later month data. There are 4 neurons in hidden layers. To carry out this process and use C program language, we get the predictions of indexes.(See. Tab.1).

The division of warning line and warning area adopts the α_{Ci} is the average of absolute value that is the samples to average value. Calculating each index Ci and α_{Ci} ., Ci–α_{Ci} is as the boundary between lower and normal areas. Ci+α_{Ci} is as the boundary between high and normal areas.(See.Tab.2).

The financial state is reflected by the value of MI calculated. The step is:

1) Based on index predications and warning area divided, the index fetches different H_i (H_i=1,2,3).(See.Tab.3).
2) By calculating the Hi of whole indexes at present to gain MI. MI=$\sum W_i*H_i/3$, W_i is the risk coefficient of index i.(See.Tab.1).
3) According to experiences, to fetch 66%, 33% as 2 critical points of MI. Then:

IF MI$\geq$66% THEN Finance is in good operation conditions;
IF 33%$\leq$MI<66% THEN Finance is in normal state;
IF MI<33% THEN Finance is in worse state.

MI=(0.2281×3+0.2280×3+0.1521×3+0.0760×3+0.0525×2+0.0329×1+0.0329×1+0.0789×2+0.0394×2+0.9981×1)/3=0.8614

From the result of certain factors, we can know the local finance is in good state.

Table 1. The Warning Prediction By Ann & Risk Coefficient

	Predition	Area	Score	Risk Coefficient
C_1	0.3060	Higher	3	0.2281
C_2	0.1490	Higher	3	0.2280
C_3	0.3890	Higher	3	0.1521
C_4	0.6641	Higher	3	0.0760
C_5	0.3793	Nomal	2	0.0525
C_6	0.7734	Higher	1	0.0329
C_7	0.8423	Higher	1	0.0329
C_8	0.4552	Nomal	2	0.0789
C_9	0.4514	Nomal	2	0.0394
C_{10}	0.9981	Lower	1	0.0792

Table 2. The Index of Warning Area

(P)rediction	Lower	Normal	Higher
C_1	$P<0.0389$	$0.0389 \le P < 0.2275$	$0.2275 \le P$
C_2	$P < 0.0751$	$0.0751 \le P < 0.0919$	$0.1088 \le P$
C_3	$P < 0.1172$	$0.1172 \le P < 0.1718$	$0.2264 \le P$
C_4	$P < 0.1852$	$0.1852 \le P < 0.2809$	$0.4036 \le P$
C_5	$P < -0.0218$	$-0.0218 \le P < 0.3861$	$0.7941 \le P$
C_6	$P < 0.2029$	$0.2029 \le P < 0.3104$	$0.4179 \le P$
C_7	$P < 0.1762$	$0.1762 \le P < 0.3169$	$0.4576 \le P$
C_8	$P < 0.1123$	$0.1123 \le P < 0.6795$	$1.2467 \le P$
C_9	$P < -0.3084$	$-0.3084 \le P < 0.5771$	$1.4626 \le P$
C_{10}	$P < 0.9999$	$0.9999 \le P < 1.0037$	$1.0075 \le P$

Table 3. The Score of Warning Area

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
Higher	3	3	3	3	3	1	1	3	3	3
Normal	2	2	2	2	2	2	2	2	2	2
Lower	1	1	1	1	1	3	3	1	1	1

3.2 The Influence of Uncertain Factors

The uncertain factors influence local finance include International and domestic economic trend (f_1), ecological environment (f_2), state policies (f_3). In the article, it discusses the method that combines expert opinions and gives the values of prediction.

To estimate the influence is to set up quantitative model, comprehend *n* suggestions of experts and get the whole appraise finally. Because the uncertain factors are fuzzy, complex and expert has some fault. The forecast by expert is roughly, contains unreliable and incomplete information[9]. By evidence theory, it not only combines effectively, but also conceives Mass function with Ω={S_1, S_2,…, S_n }.

Because each expert has the differences in knowledge, understand and favor, the estimate of every expert might not reliable totally. So, to fetch different weights of n experts to index t_1 ,(λ_1^{t1} , λ_2^{t1} , …, λ_k^{t1}), λ_k^{t1} ≥0,k=1,2,…,n, $\sum_{k=1}^{n}\lambda_k^{t1}=1$,and assume the experts reliable state is the highest when the weight is the highest too. Other expert reliable state is correlation with not only own experience and favor, but also the difference of the expert who reliable state is highest. Here, to define the reliable state of the *j* expert is that.

$$\alpha_j^{t1}=\omega_j^{t1}[1-(\lambda_{\max}^{t1}-\lambda_j^{t1}] \text{ , j= 1,2,…,n} \qquad (3)$$

$\lambda_{\max}^{t1}=\max_{1\le i\le n}\lambda_i^{t1}$, ω_j^{t1} is the coefficient of reflecting expert's favor & experience. Generally , to fetch 0.9≤ ω_j^{t1} ≤1, 0≤ α_j^{t1} ≤1,j=1,2,…,n. If the λ_j^{t1} is nearness $\lambda_{\max}^{t1}$, the α_j^{t1} will be much bigger.

For there are some unreliable in expert estimate, the j expert's appraising P_j^{t1} should be calculated again. $P_j^{t1^*}=\alpha_j^{t1}P_j^{t1}$, $\sum_{k=1}^{3}\alpha_j^{t1}\mu_j^{t1}(A_k)\le 1$, 1- $\sum_{k=1}^{3}\alpha_j^{t1}\mu_j^{t1}(A_k)\le 1$ belongs in unknown information of expert estimate. Expressed by Mass function, we can get the degree of the j expert estimate to index t_1.

In the article, according to the actual conditions, to provide the estimates of four experts about International and domestic economic trend (f_1), ecological environment (f_2), state policies (f_3), and α=(0.9,0.85,0.80,0.86). Based on Ω={Good(S_1), Normal(S_2), Bad(S_3)}, to conceive Mass function and utilize D-S Principle[10], the results can be gained.

$m^{f1}=(0.038,0.012,0.012)$, $m^{f2}=(0.036,0.061,0.020)$, $m^{f3}=(0.036,0.019,0.027)$.

In order to combine exactly, we draw new mass function by changing to vector quantity.

$m^{f1'}=(0.612,0.194,0.194)$, $m^{f2'}=(0.307,0.521,0.172)$ $m^{f3'}=(0.439,0.232,0.329)$

Table 4. Mass Function of Ann Predication

	MI≥66%	33%≤MI<66%	MI<33%
S_1	0.5	0.25	0.2
S_2	0.3	0.5	0.3
S_3	0.2	0.25	0.5

3.3 Evidence Combining

To define value of Mass function when the prediction of BP neural network is in different area.(See.Tab.4). It can know m_{BP} ={0.5, 0.3, 0.2} here. According to the $m^{f1'}$, $m^{f2'}$, $m^{f3'}$ given by expert estimates of uncertain factors, the final m = (0.907,0.154,0.046) is gained by what D-S principle combined $m_{BP}, m^{f1'}, m^{f2'}, m^{f3'}$ (see *Eq.2*). From it, we can find out that the local finance is in very good state.

4 Conclusions

This paper considers the practical problem in local financial warning and introduces uncertain factor's influence. To discuss how to combine different expert estimates of uncertain factor, and use BP neural network to do with complex relations between deterministic factors and finance. Based on evidence theory and ANN, the new prediction method can get better result except in the predicted values that totally conflict. It gives a new thought in complex system prediction.

Acknowledgements

The work described in this paper was supported by Project 60375024 of NSFC, Project 2002AA2430031 of 863, and the Applying Basic Research Grants Committee of Science & Technology of Chongqing.

References

1. Luo, S.: Use of Grey Forecasting in The Tendency Analysis of Urban Sustainable Development and Counter Measures. China Population, Resources and Environment, **12** 2002(1) 114-119
2. Wu, W., Li, Y.: On the Profit Prediction of The Civil Project Item by Regression Analysis, Journal of Hefei University of Technology, (1999) 53-57
3. Ramaan, G., Liu, T.: Nonlinear Modeling and Prediction with Feed forward and Recurrent Networks. Physica D, **108** (1997) 119-134
4. Li, Y., Xi, Q.: On Developing Forecasting Support System in China. Journal of Northwestern Polytechnical University, (2001) 78-81
5. Cheng, S., Guo, B.: Fuzzy Predication. Guizhou: Guizhou Science and Technology Press (1994)
6. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press, 1996
7. Wang, Y., Tu, J.: The neuron network controls. Beijing: Publishing house of the mechanical industry (1998)
8. Guo Yuqing: The Construction of The System's The Warning Index in Local Financial Risk. Western Financial Accounting (2003)
9. Boutheina Ben Yaghlane: Belief function independence, International Journal of Approximate Reasoning, **4** (2003) 156-162.
10. Herrera, F., Herrera, E., Viedma, Lverdegay, J.: A Model of Consensus in Group Decision Making under Linguistic Assessments. Fuzzy Sets & Syst, **78** (1996) 73-87.

Novel Questionnaire-Responded Transaction Approach with SVM for Credit Card Fraud Detection

Rongchang Chen[1], Tungshou Chen[2], Yuer Chien[2], and Yuru Yang[2]

[1] Department of Logistics Engineering and Management
rcchens@ntit.edu.tw
[2] Department of Information Management
National Taichung Institute of Technology
No. 129, Sec. 3, Sanmin Rd., Taichung, Taiwan 404, China
tschen@ntit.edu.tw

Abstract. One of the most potential methods to prevent credit card fraud is the questionnaire-responded transaction (QRT) approach. Unlike traditional approaches founded on past real transaction data, the QRT approach proposes to develop a personalized model to avoid credit card frauds from the initial use of new cards. Though this approach is promising, there are still some issues needed investigating. One of the most important issues concerning the QRT approach is how to predict accurately with only few data. The purpose of this paper is to investigate the prediction accuracy of this approach by using support vector machines (SVMs). Over-sampling, majority voting, and hierarchical SVMs are employed to investigate their influences on the prediction accuracy. Our results show that the QRT approach is effective in obtaining high prediction accuracy. They also show that combined strategies, such as weighting and voting, majority voting, and hierarchical SVMs can increase detection rate considerably.

1 Introduction

Since the emergence of credit cards, preventing credit card fraud has long been one of the most important issues. In coping with this fraud problem, typically, massive amount of actual transaction data of other users are used to develop models to predict a new case [1],[2],[3],[4]. Rather than detecting credit card fraud by past transaction data, Chen et al. [5],[6] proposed a novel approach to solve the fraud problem. They proposed to develop a personalized model to detect fraud. One can first gather personal transaction data of users by an online, self-completion questionnaire system. The gathered questionnaire-responded transaction (QRT) data from the online system are considered as the transaction records and are utilized to build up a personalized model, which is sequentially in use to predict whether a new, actual transaction is a fraud or not. Since the illegal user's consumer behavior is usually dissimilar to the cardholder, the fraud can be avoided from initial use of a credit card, even without any transaction data.

The QRT approach is promising. However, there are still some problems needed investigating. One of the most important issues regarding the QRT approach is how to

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 916–921, 2005.

predict accurately with only few data, say 100 to 200, since the users are usually not willing to answer too many questions. In this paper, we will examine the influences of data number and data distribution on the prediction accuracy. Several typical ways for improving the prediction accuracy are also employed to study their influences.

Some useful tools have been successfully applied to fraud detection, such as, artificial neural network, learning machines, and so forth. Recently, support vector machines (SVMs) [7] are up-and-coming as a powerful machine learning technique to classify and do regression. The SVM has already been successfully used for a wide variety of problems, like bio-informatics, natural language learning, text mining, pattern recognition, and so forth. The SVM has some desirable properties that make it a very powerful technique to use [8]: computational complexity is enhanced by the use of the kernel trick, over-fitting is avoided by classifying with a maximum margin hyper-plane, and only a small subset of the training set is needed to separate the different classes of the problem. Consequently, we will use the SVM to deal with the credit card fraud.

The remainder of this paper is organized as follows. Section 2 gives a brief overview of SVMs. Section 3 describes the approach for detecting fraud. The experimental results are showed in Section 4. Finally, conclusions are presented.

2 Support Vector Machines (SVMs)

SVMs were developed by V.N. Vapnik [7] and have been demonstrated with sound theoretical justifications to give a good generalization performance compared to other algorithms [8]. They can generate regression or classification function from a set of training data. To do classification, SVMs operate by finding a hyper-plane in the space of possible inputs. This hyper-plane will divide the positive examples from the negative examples. The split will be chosen in such a way that it has the largest distance from the hyper-plane to the nearest of the positive and negative examples [8]. Instinctively, this makes the classification correct for testing data that is close to, but not identical to the training data. The problem is to find a decision surface that "best" separates the data points into two classes based on the Structural Risk Minimization Principle. The decision function is of the form

$$y = sign(\sum_{i=1}^{N} y_i \alpha_i K(x, x_i) + b)$$

where x is the vector of a test example, $y \in \{-1, 1\}$ is a class label, x_i is the vector for the i^{th} training example, N is number of training examples, $K(x, x_i)$ is a kernel function, $\alpha_i = \{\alpha_1, \ldots, \alpha_N\}$ and b are the parameters of the model.

3 Approach

Traditional approaches use actual transaction data to build up models for predicting new cases. They give good performance of fraud detection in some situations. Nevertheless, there are no or few real transaction data for a new card user. One common solution is to predict a new transaction based on the data of other users. This approach may cause lower prediction accuracy because the consumer behavior of an individual is normally different from that of a consumer group.

To improve the problem mentioned above, we will employ the QRT approach [5], which proposes to set up a personalized model for a new user from his/her initial use of a credit card, to prevent fraud. The procedure is shown in Fig. 1.

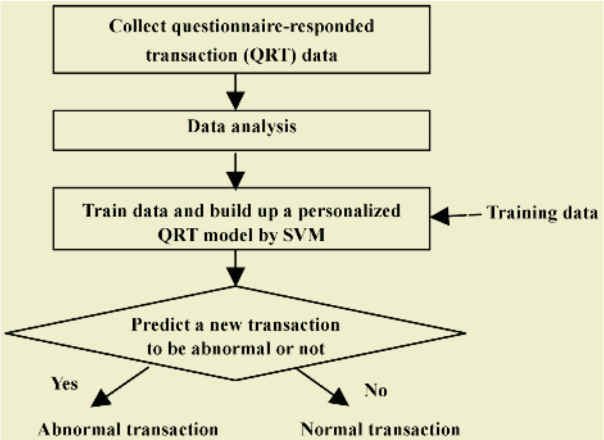

Collect questionnaire-responded transaction (QRT) data
Data analysis
Train data and build up a personalized QRT model by SVM
Training data
Predict a new transaction to be abnormal or not
Yes
No
Abnormal transaction
Normal transaction

Fig. 1. The QRT approach for predicting fraud [5]

To examine the influence of data number on the prediction accuracy, more than 800 data were gathered from a male student.

An online system is set up to establish an efficient and cost-saving questionnaire. The design platform of online questionnaire system is Windows 2000, IIS 5.0. The program we used is ASP. The database is SQL 2000. The questions are formed based on the individual's consuming preference derived from a survey on consumer behavior [9].

Consumer behavior varies significantly with each person. Hence, it is acceptable to classify their behavior according to some attributes. The gathered personal QRT data are mainly composed of several parts: gender, age, transaction time, transaction amount, and transaction items. The transaction items were divided into 6 subclasses.

The prediction result depends on the data amount and the ratio of positive to negative samples. A general rule for prediction is that the accuracy increases with an increase in data number when the quantity is small. The real transaction data are usually skewed [2],[4],[10], i.e., genuine data are much more than fraud data. In general, there are two basic methods to deal with the skewed problems [10]: over-sampling, which is to replicate the data in the minority class, and under-sampling, which is to throw away some data in the majority class. We will study the influence of over-sampling.

4 Results and Discussion

In this paper, we used mySVM [11] to train all QRT data. The base classifier we used is the radial kernel.

For the convenience of discussion, let us designate the number of data as N, the ratio of negative samples to total samples as Rn, true negative as TN, true positive as TP, false positive as FP, and average prediction accuracy as AVG.

Figure 2 shows the influence of Rn on the prediction accuracy. Rn varies from 0.3 to 0.7. N is set to be 100, 140, and 180, covering the question number that users are generally willing to answer. Each subclass is set to have the same amount of data. As Rn increases, TN rate increases. This trend is favorable for the QRT approach since we can acquire high TN rate if we increase Rn. Actually, this can easily be done via the gathering of abnormal QRT data by the online system. Alternatively, TP rate decreases with an increase in Rn. This result is reasonable since a higher Rn has fewer normal data, leading to a worse prediction model. The average prediction accuracy is about the same, regardless of Rn.

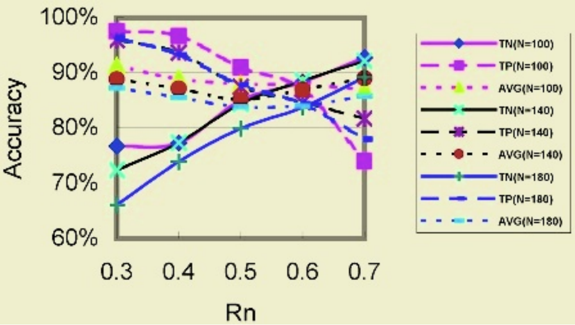


Fig. 2. The influence of negative samples ratio, Rn

For credit card detection, the TN rate (a high TN rate or a low FP rate) is the most concerned. When Rn is low, it is generally difficult to obtain a high TN rate. To get better TN rate, we can increase Rn. Over-sampling [10] is one of the most popular methods. The over-sampling is to replicate the data in the minority class. The replicating process is repeated several times, each time the same data are put into the training set. Instead of replicating the data, the QRT approach can improve the TN rate by adding the data. We can first collect the abnormal QRT data from the online system, then put them into the training set, and train them by using the SVMs. To examine the effectiveness of this method, we add 18 negative data each time to the original training set, which has 180 data with Rn = 0.1, and retrain. The testing is repeated 5 times to obtain an average value. As the times of adding increases, the TN rate increases. This result illustrates that the QRT approach is effective for better fraud detection rate. The improvement by adding the data is generally better, as we can see in Fig. 3, which illustrates the results of different data number ratios (5:4:3:2:2:2 and random ratio) of 6 subclasses.

The SVMs ensembles method [10] is also employed to investigate the influences of different strategies on detection rate. The architecture is depicted in Fig. 4. First, divide the positive examples into K partitions. The next step is to train SVMs separately on every subset of the training set, and finally combine all component SVMs by various strategies. The results are summarized in Table 1. The data set has 180 data with Rn = 0.1. It was divided into 9 partitions and trained. We can see that majority voting,

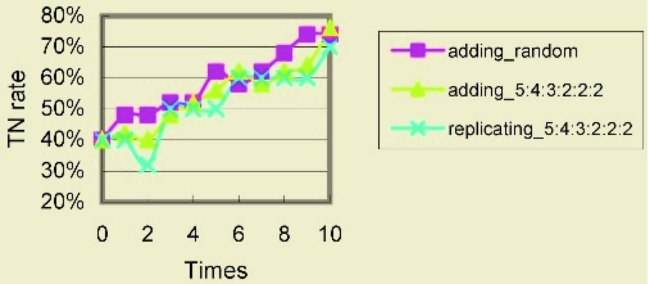
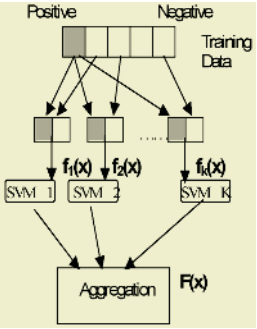


Fig. 3. The influences of adding (QRT) and replicating abnormal training data

more-than-3-negatives rule, 2 kinds of weighting and voting, and hierarchical SVMs [10] all can improve the detection rate and save fraud cost considerably.

The value of final decision function *F(x)* is also shown in Table 1. A higher absolute value of *F(x)* indicates a higher probability of being negative for a negative result. Table 1 also shows the cost saved by different methods of combination. Note that the reconfirmation cost is neglected. However, if it is significant when compared to fraud cost, it should be taken into consideration. A useful tool to consider the cost is the lift chart. One can use lift chart to find subsets of test instances with the greatest possible proportion of negative samples, higher than in the test set as a whole. The lift chart for the case of weighting by TN rate and voting is illustrated in Fig. 5. At about 40% of the sample size, the lift factor reaches a maximum value of about 1.4, as we can see from Fig. 5.



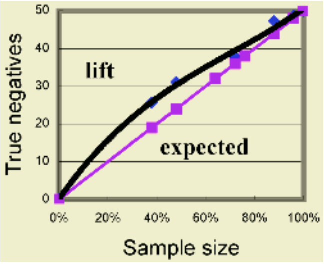


Fig. 4. Architecture of the SVM ensembles [10]

Fig. 5. Lift chart for the case of weighting by TN rate & Voting

5 Conclusions

We have employed the questionnaire-responded transaction (QRT) approach to deal with the credit card fraud problem. The SVM has been used to train and to build up a classifier for detecting abnormal data. The results prove that a higher rate of the negative samples to total samples influences the true negative rate positively. This result is favorable for the QRT approach since one can increase the ratio of negative samples

Table 1. Summary of TN rate and cost using different methods

Strategy or rule	TN rate	Average F(X)	Fraud Cost (%)	Cost Saved (%)
Base case	0.48	-	100	-
Majority Voting (more than 4 negatives)	0.74	-3.32	28.9	71.1
More than 3 negatives	0.82	-2.35	21.5	78.5
Weighting by 10-fold cross validation & Voting	0.74	-1.93	28.9	71.1
Hierarchical SVMs & Truncating	0.76	-2.95	26	74
Weighting by leave-one-out & Voting	0.74	-3.07	28.9	71.1

by collecting the abnormal QRT data. Results from this study also show that the QRT approach is effective and the detection rate can be improved significantly by combined strategies, like majority voting, weighting and voting, and hierarchical SVMs.

References

1. Maes, F.S., Tuyls, K., Vanschoenwinkel, B., Manderick, B.: Credit Card Fraud Detection Using Bayesian and Neural Networks. Proc. NEURO Fuzzy, Havana, Cuba (2002)
2. Chan, P.K., Fan, W., Prodromidis, A.L., Stolfo, S.J.: Distributed Data Mining in Credit Card Fraud Detection. IEEE Intel. Sys., Nov.-Dec (1999) 67-74
3. Brause, R., Langsdorf, T., Hepp, M.: Neural Data Mining for Credit Card Fraud Detection. IEEE Int. Conf. Tools with Artif. Intel, (1999)
4. Chan, P.K., Stolfo, S.J.: Toward Scalable Learning with Nonuniform Class and Cost Distributions: A Case Study in Credit Card Fraud Detection. Proc. 4th Int. Conf. Knowl. Disco. and Da. Min., Menlo Park, Calif (1997) 164-168
5. Chen, R.C., Chiu, M.L., Huang, Y.L., Chen, L.T.: Detecting Credit Card Fraud by Using Questionnaire-Responded Transaction Model Based on Support Vector Machines. Lect. Notes Comp. Sci., **3177** (2004) 800-806
6. Chen, R.C., Lin, C.J., Lai, L.J., and Chien, Y.E.: Employing Support Vector Machines to Detect Credit Card Fraud for New Card Users. To Appear In Asian J. Infor. Tech.
7. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer (1995)
8. http://www.kernel-machines.org/
9. http://www.104pool.com/
10. Yan, R., Liu, Y., Jin, R., and Hauptmann, A.: On Predicting Rare Classes with Ensembles in Scene Classification. IEEE Int. Conf. Acoustic, Speech and Signal Processing, Apr, (2003)
11. Rüping, S.: MySVM-Manual, Computer Science Department. AI Unit University of Dortmund, (2000)

Learning of Neural Networks for Fraud Detection Based on a Partial Area Under Curve

Lae-Jeong Park

Department of Electrical Engineering, Kangnung National University
Kangnung Gangwon-do 210-702, Korea
ljpark@kangnung.ac.kr

Abstract. This paper addresses an effective approach of training a neural network (NN) classifier for real-world credit card fraud detection. In the proposed approach, an evolutionary search algorithm is used to directly improve the performance of a NN classifier in a local operating range in terms of the detection rate of fraudulent usages by optimizing a partial area under a domain-specific curve. The experimental results on 'real' credit card transactions data demonstrate that the proposed approach produces classifiers of a higher detection rate in a desired range of false detection rates.

1 Introduction

As the number of credit card transactions grows, banks and credit card companies have been suffering from loss of fraudulent transactions by stolen or clone credit cards. They have adopted commercial or proprietary early warning systems to reduce the loss from the fraudulent transactions. In the machine learning and data mining communities, there has been some research on classification models and classifier learning algorithms for effective fraud detection[1]. Unlike conventional classification problems, learning classifiers for credit card fraud detection is plagued by some peculiar properties:

- Fraudulent transactions are overwhelmed by legitimate ones in number. The class ratio of fraudulent to legitimate transactions is severely imbalanced, ranging from 10^{-3} to 10^{-4} [2][3][4].
- Fraudulent transactions resembles legitimate ones so that it is impossible to detect fraudulent transactions without false detection of legitimate ones. The class distribution is severely overlapping.
- The class distribution often fluctuates or changes with time since usage patterns of card holders vary with time and new kinds of frauds occur.

Classification accuracy that is a common performance measure is not suitable to evaluation of classifiers for fraud detection due to the skewed and overlapping class distribution. For example, the classification accuracy of all-pass classifier is 99% in a class ratio of 1:99, but is meaningless in fraud detection[2]. Receiver

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 922–927, 2005.

operating characteristic (ROC) curve[1] has been widely used to evaluate classifier performance in the skewed and overlapping data sets since its introduction by [5] in the machine learning and data mining communities. The ROC curve makes it possible to visualize a tradeoff between true positive rate (TPR) and false positive rate (FPR), which should be made inevitably in fraud detection due to the severely overlapping class distribution. Previous studies in credit card fraud detection[3][4][6][7] have focused on learning methods or classification models to deal with the problems caused by the skewed class distribution. There have been, however, few previous works that have evaluated and compared classifier's discrimination performances explicitly on the ROC space, except [8] although it is crucial to assess classifier performance by taking into account a tradeoff between TPR and FPR.

In practice, a classifier for credit card fraud detection must be operated on a certain operating range on the ROC space, say, TPR of $15\% - 20\%$ and FPR of $5\% - 10\%$. The operating range on the ROC space is determined by taking into account several operational constraints such as the maximum number of fraudulent transactions that can be investigated daily or weekly in addition to a compromise between TPR and FPR. Hence, in credit card fraud detection, interests do not lie in the entire range of FPR, but only in a certain range of FPR. Therefore, it is more significant to evaluate and optimize classifier's performance, for example, TPR within a specific range of FPR associated with the chosen operating range. It should be also noted that classifiers are evaluated in an operating range, not a single operating point because the operating point changes with time, from a month to another, depending on the changing class distribution.

Neural network (NN)-based classifiers have been commonly adopted in credit card fraud detection because NNs are superior to other linear classifiers thanks to nonlinear discrimination capability. Unfortunately, the backpropagation(BP) algorithm that minimizes the mean square error (MSE) at the outputs of NNs is not suited for optimizing a classifier's performance in a local operating range. It is because the MSE minimization at the outputs does not correspond exactly to maximization of the performance in the local operating range. Recently, evaluation and learning algorithms based on an area under ROC curve (AUC)[5][9][10] have attracted much attentions because AUC enables us to measure and compare classifier performance informatively in the highly skewed data sets, compared with the error rate or MSE. It has been revealed by [11] that learning algorithms designed to minimize the error rate may not lead to the best AUC possible values. To my knowledge, there have been few researches on methods designed to train a classifier for fraud detection by optimizing a partial AUC value or local discrimination performance.

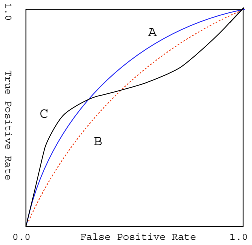
In this paper, an effective learning method is proposed that is designed to produce a NNs-based fraud detection classifier that shows high performance

[1] True positive rate is plotted on the Y axis and false positive rate is plotted on the X axis. The ROC curve of a classifier is obtained by adjusting a threshold on the classifier's continuous output.

locally in a desired operating range by utilizing a partial area under a domain-specific ROC-like curve, which is difficult to achieve with general MSE-based NN learning methods.

2 Learning Based on Partial Area Under Curve

The ROC curve of a NN classifier is obtained by varying a threshold, θ, on a single continuous output of the classifier, so-called a fraud score ranging from $[-1, 1]$. Transactions whose fraud scores are greater than θ are classified as frauds. For a given θ, a point of (FPR,TPR) on the ROC space is given by equations in Fig. 1. Fig. 1 shows an example of ROC curves of three classifiers. Classifier B is completely outperformed by classifier A. On the other hand, classifier C is better than classifier A at a lower FPR range although it is worse at a higher FPR range that is far from a desired operating range in credit card fraud detection. This is why classifiers for fraud detection should be trained and evaluated in a local operating range.



$$FPR = \frac{\text{incorrectly detected legitimate}}{\text{total legitimate}}$$

$$TPR = \frac{\text{correctly detected fraud}}{\text{total fraud}}$$

Fig. 1. Three ROC curves of three classifiers and definitions of FPR and TPR

Practically, it is desirable that the operating range of a classifier is represented by a rejection rate (true positive plus false positive rates) since the rate is a major monitoring variable that should be adjusted carefully due to the constraint imposed on the limitation of fraud investigation. Thus, a domain-specific curve is introduced in which the rejection rate (X axis) and the classifier's performance on the rejection rate (Y axis) are visualized more explicitly than in the ROC curve. For the domain-specific curve, the objective is to maximize an average of the correctly detected frauds in a chosen range of the rejection rates by maximizing an area under the curve (PAUC). The training of a NN classifier based on the PAUC is depicted in Fig. 2 and the PAUC is given by

$$PAUC(r_0, r_1) = \frac{1}{Z} \int_{r_0}^{r_1} N_F(r) \cdot w(r) dr, \tag{1}$$

where $N_F(r)$ is a function that represents the number of detected frauds with respect to the rejection rate $r(\%)$, $[r_0, r_1]$ is an operating range, $w(r)$ is a weighting window for averaging performances in $[r_0, r_1]$, and Z is a normalization factor.

Note that in addition to better performance in the local operating range, training with the PAUC criterion makes it possible to produce a NN classifier that is somewhat robust to the fluctuating class distribution because it is evaluated in a range of rejection rates. Since it is difficult to compute the partial derivatives of Eq. (1) with respect to the weights, a NN is trained by Evolutionary Programming (EP) rather than gradient-descent algorithms so that the weights and its threshold θ are chosen in terms of maximization of Eq. (1).

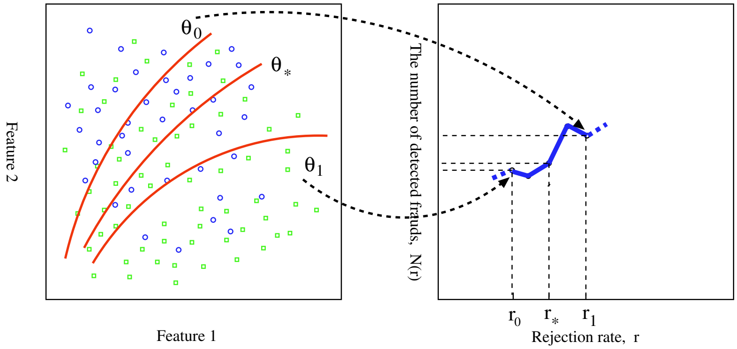


Fig. 2. The partial area under $N_F(r)$ curve in $[r_0, r_1]$, where r is a rejection rate and is dependent on thresholds θ on the output of a NN classifier

3 Experimental Results and Discussions

Two data sets of credit card transactions labeled as legitimate or fraudulent were provided by a credit card company in Korea. One set consisting of about $51,000$ transactions of $7,800$ customers collected selectively during one year is used as a training data set. Another data set of about 7 millions transactions that were collected during three consecutive months is used to evaluate the NN classifier. The ratios of fraudulent to legitimate transactions of the two data sets are 1:4 and 1:1700, respectively. The six features are extracted from each transaction based on the amount of money, the frequency of card usage and etc., and are used for the inputs of the NN classifier.

Lack of a sufficient history of customer's past transactions makes us choose a classification approach based on common features identifying fraudulent transactions, rather than a user profiling-based approach that relies on patterns of card usage or buying history. A multi-layered perceptron (MLP) with one hidden layer and sigmoidal functions is used as a NN classifier. The EP is used to train a NN classifier in order to maximize Eq. (1) and minimize the MSE for fair comparisons. The EP uses Gaussian mutation and its parameter values for all experiments are as follows. The number of generations and the population size are chosen as 500 and 10, respectively, for a reasonable convergence speed. The tournament size is 8 and σ in the Gaussian mutation is 1.0.

Two sets of experiments were performed to evaluate and compare the PAUC-based NN classifier with the MSE-based one. For each criterion, ten NNs were trained with 10 and 20 hidden nodes. The operating range was chosen as $[0.1, 0.4]$ to reflect reality. Fig. 3(a) shows a closer X-Y plot of the performances of ten classifiers trained with the PAUC criterion and ten ones with the MSE criterion on the (MSE,PAUC) space. The classifiers with almost equal MSE values have different PAUC values, and vice versa. The results imply that as pointed in [11],the MSE minimization does not tend to maximize the PAUC. A partial explanation could be that, the increase of $N_F(r)$ in $[r_0, r_1]$ by maximization of Eq.(1) forms the decision boundary of a NN classifier such that the data in the feature space corresponding to the rejection rates of $[r_1, 100]$ has larger MSE values. The operating ranges of high rejection rates greater than r_1 are of no consequence in credit card fraud detection.

Fig. 3(b) shows the average performances of the top five classifiers trained with the PAUC maximization and the top five ones with the MSE minimization on the test set. Even though training of a NN classifier with the PAUC criterion generates classifiers with large MSE values, it achieves the goal of increasing the partial area under $N_F(r)$ curve for rejection rates of $[0.1, 0.4]$, compared with the MSE-based classifiers. The improvement is consistent with the number of hidden nodes. To compensate the increase in a range of $[0.1, 0.4]$, the averaged $N_F(r)$ curve of the PAUC-based classifiers increases more slowly at a medium range of rejection rates than that of MSE-based classifiers (not shown in Fig. 3(b)).

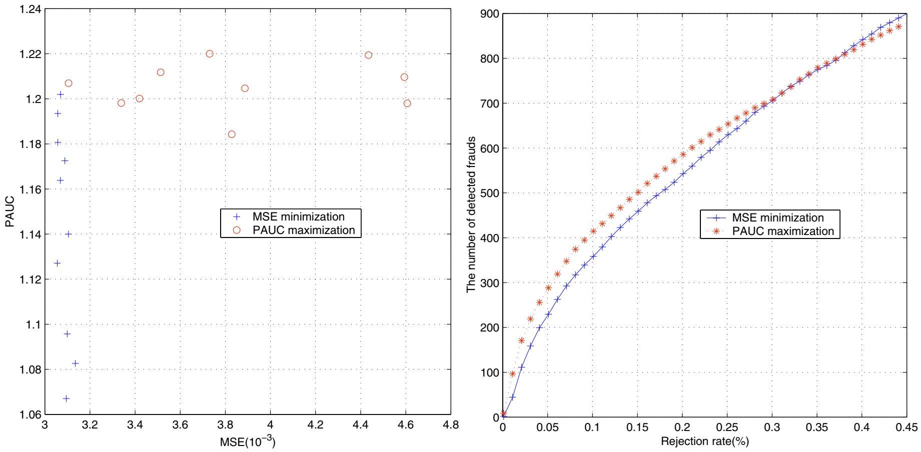


Fig. 3. (a) X-Y plot of twenty NN classifiers' performance on the (MSE,PAUC) space, a half trained with the PAUC, the others with the MSE. (b) The two average performances of the top five NN classifiers trained in terms of the PAUC and MSE, respectively. All NNs have 20 hidden nodes

4 Conclusions

An effective learning approach has been suggested that is designed to directly optimize a classifier's performance in a desired local operating range for a real-world credit card fraud detection. The experiments with real credit card transactions data have demonstrated that the neural network classifier could detect more frauds in a specific range of rejection rates by maximizing a partial area under a domain-specific curve, compared with classifiers trained by the MSE minimization.

Acknowledgements

This research was supported by the Ministry of Information and Communication), Korea, under the ITRC support program supervised by the IITA.

References

1. Bolton, R. J., & Hand, D. J.: Statistical Fraud Detection: A Review. Statistical Science, **17** (2002) 235–255
2. Provost, F.: Learning with Imbalanced Data Sets 101. AAAI'2000 Workshop on Imbalanced Data Sets (2000)
3. Dorronsoro, J. R., Ginel, F., Sanchez, C., & Cruz, S.: Neural Fraud Detection in Credit Card Operations. IEEE Trans. on Neural Networks, **8** (1997) 827–834
4. Park, L.-J., Kim, S.A., Cho, H.J., Kim, T.S., & Wang, B.H.: A Credit Card Fraud Detection System Based on Hybrid of Neural Networks and DecisionTree. Proc. of the Korean-U.S. Sic. & Tech. Symposium (1998)
5. Provost, F., & Fawcett, T.: Analysis and Visualization of Classifier Performance Comparison under Imprecise Class and Cost Distributions. Proc. Int'l Conf. Knowledge Discovery and Data Mining (1997) 43–48
6. Maes, S., Tuyls, K., Vanschoenwinkel, B., & Manderick, B.: Credit Card Fraud Detection Using Bayesian and Neural Networks. Proc. of Int'l NAISO Congress on Neuro Fuzzy Technologies (2002)
7. Chan, P. K., Fan, W., Prodromidis, A. L., & Stolfo, S. J.: Distributed Data Mining in Credit Card Fraud Detection. IEEE Intelligent Systems, **14** (1999) 67–74
8. Park, L.-J.: Evolutionary Learning of Neural Networks Classifiers for Credit Card Fraud Detection. Journal of Fuzzy Logics & Intelligent Systems, **11** (2001) 400–405 (in Korean)
9. Bradley, A. P.: The Use of the Area under the ROC Curve in the Evaluation of Machine Learning Algorithms. Pattern Recognition, **30** (1997) 1145-1159
10. Yan, L., Dodier, R., Mozer, M. C., & Wolniewicz, R.:, Optimizing Classifier Performance via the Wilcoxon-Mann-Whitney Statistics. Proc. Int'l Conf. Machine Learning (2003) 848–855
11. Cortes, C., & Mohri, M.: AUC Optimization vs. Error Rate Minimization. Advances in Neural Information Proceesing Systems, **15** (2003)
12. Bäck, T., Fogel, D. B., & Michalewicz, Z.: Handbook of Evolutionary Computation. IOP Publishing and Oxford University Press (1997)

Customer Churning Prediction Using Support Vector Machines in Online Auto Insurance Service

Yeon Hur[1] and Sehun Lim[2]

[1] Department of Business, Chung-Ang University
40-1, Naeri, Deaduck-Myun, Ansung City, Kyunggi-Do 456-050, South Korea
yeonhur@cau.ac.kr
[2] Department of Information System, Chung-Ang University
40-1, Naeri, Deaduck-Myun, Ansung City, Kyunggi-Do 456-050, South Korea
slimit@hanmail.net

Abstract. Support vector machines (SVMs) are promising methods for the prediction of online auto insurance customer churning because SVMs use a risk minimization principal that consists of the empirical error and the regularized term predicting the switching probability of an insured to other auto insurance company. In addition, this study examines the feasibility of applying SVM in online insurance customer churning by comparing it with other methods such as artificial neural network (ANN) and logit model. This study proves that SVM provides a promising alternative to predict customer churning in auto-insurance service.

1 Introduction

In general, ANN (Artificial Neural Network) is known to be the state-of-the-art forecasting tool in business management. For example, ANN is widely used to forecast a probability of bankruptcy or insolvent of banking business. It is also used to predict churning of customers and customer service management in the credit card industry.

With an expansion of the Internet users, recent auto insurance market in Korea experiences the following two difficulties: (1) the off-line insurers lose their market share to the on-line insurers because the on-line insurers suggest lower auto insurance premiums and provide various insurance information; (2) the on-line premium comparison(quote) service provided by on-line aggregators, independent agents or brokers, makes auto insurance customers to switch their insurers much easily without any complication. These on-line auto insurance marketers both on-line insurers and on-line aggregators allow the customers to reduce switching cost and information searching cost to almost nil. Therefore, the on-line market becomes more important to the auto insurers and requires them to have more proactive strategic responses on it to survive in the market and further to maximize their corporate values.

Data mining is one of the major tools to companies, i.e., insurers and credit card firms, where customer information is the most important source and base to setting strategies regarding customer relationship management. The data mining analyses

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3497, pp. 928–933, 2005.

related to customer churning include logit, multivariate discriminant analysis (MDA), case-based reasoning (CBR), artificial neural network (ANN), genetic algorithm (GA), and inductive method.

It is scarce to find, however, previous studies on the subject of customer churning in on-line auto insurance market, which reflects an embryonic stage of on-line insurance market in Korea. On the other hand, we have a bunch of papers that study customer churning in the credit card industry. They have employed various data mining tools such as ANN, CBR, CART, C5.0 in their analyses [1], [2]. These studies argue that the ANN provides better prediction than the logit model and the MDA in customer churning estimation. But the prediction performance using ANN analysis depends on various factors such as the number of hidden layers and nodes as well as learning rate and the number of momentum, etc. Also the method shows somewhat lower prediction level when the data has noises and complications.

It is known that the Support Vector Machine (SVM) based on the statistical theory on machine learning algorithm, developed by Vapnik et al. [13], outperforms the ANN algorithm in prediction performance [14]. It produces better results than any other methods introduced so far in terms of capability of pattern recognition, performance prediction, and solving classification problems. SVM is based on structure risk minimization principle whereas the traditional ANN algorithm is based on empirical risk minimization principle. That is, SVM is characterized to minimize the upper bound of a generalization error whereas the ANN model focuses to minimize false classification error.

SVM, enabling global minimum, can fix an overfitting problem inherent in the ANN model by overcome a limitation of the local minimum. Using obvious theory, it also eases the interpretation of the results and provides faster classification learning with a smaller sample of data.

Several studies using SVM have been done in the major area of business management such as in stock market index forecasting, prediction of default, evaluation of bond, and so forth. This paper applies the SVM in the prediction of customer churning in case of on-line auto insurance. Also this study compares the result of SVM analysis with those of the ANN model and the logit model. Our results can provide auto insurance firms practical connotations.

This study is structured as follows. Section 2 introduces basic concepts of SVM and previous research applications in business management. In section 3, this article selects prediction variables to measure customer churning in on-line auto insurance contracts. Then explanations on the research and experiment structure are followed. Section 4 analyzes the empirical results. Finally, this article concludes and mentions limitations of the study.

2 Research Background

2.1 Support Vector Machines and Their Application in Business

SVM, developed by Vapnik [12], [13], is a statistical learning theory based on the ANN model. SVM enables mathematical classification of data by matching nonlinear problems in an input space to linear problems in a high-dimensional feature space [6]. As mentioned, SVM can overcome the inherent problems of ANN technique. SVM was used in the area of pattern recognition and classification problem in its early stage. For example, Joachimes [5] proves that SVM outperforms various mechanical learning models in the document classification process. In the medical research, Guyon et al. [4] shows that SVM generates the best result out of various machine-learning algorithms in cancer cells classification test. Recently SVM is frequently employed for the purpose of classification and prediction in business management arena too (i.e. stock index prediction [7], and bond rating [14] that deals with time series data). This study applies SVM to predict the probability of customer churning in online auto insurance service.

3 Methodology

3.1 Research Variable

This study employs total number of 13,200 sample data from an on-line aggregator (independent agent). This company provides consumers auto insurance premium quote service as well as related information regarding auto insurance as well as other general insurance products. If a customer provides basic underwriting information and select insurance coverage, then he (she) can compare his (her) auto insurance premium quoted from all insurance companies at the same time. Then he (she) can select an insurer and purchase the coverage chosen through the Internet. Out of 25 variables which we believe delivers certain meaning to our analysis, this study employs a predefined experiment that can minimize the number of variables to be considered, thereby we can have a parsimonious model and easily interpret the results.

This study considers several indication variables. First, the output or dependent variable is defined as a binary variable. If a customer switches its insurance company next year, then the dependent variable has a value of "1", otherwise "0" (no switch). Independent variables are composed of two types of variables: indication variables and ratio (numeric) variables. To select meaningful variables, this study did Chi-Square test for indication variable data and t-test for numeric variable data. Based on the two tests, we select 15 meaningful variables to be considered in our model. The summary statistics are presented in Table 1.

This study uses total number of 13,200 sample data that are divided into two groups based on company switching results. One group of data is consisted of 6,600 observations that switched their insurance company when they renewed their insurance policy next year. The other group is also consisted of the same number of observations who renewed their contracts with the same insurers. Also the sample data are grouped into training data set and holdout data set to test predictability of the model. The composition of the training data and holdout data are 80:20 or 10,560 and 2,640 observations respectively.

Table 1. Summary statisics

Feature name	Mean	Std. Deviation	Minimum	Maximum
Age of Driver	39.14621	9.11668	19	95
Zip Code	3.637727	1.891552	1	7
Car Type	2.112273	1.404095	0	5
Price of Car	607.2974	550.3214	1	9600
Credit/Debit	66.69394	24.31643	40	200
Surcharge	2.347197	6.212417	0	50
Driver Endorsement	2.920076	1.080609	1	5
Age Endorsement	1.981894	0.13334	1	2
Number of Airbag	0.434394	0.656068	0	2
Property Liability	3258.258	2628.364	0	100000
Medic. Expense	5396.856	5945.052	0	20000
Deductible(auto)	4.092197	8.421139	-1	50
Last Premium(t)	145764.1	245559.1	0	2005720
Old Insurer's Quote(t+1)	442528.5	245609.1	0	2504970
Type of Coverage	1.216439	0.481858	1	3

3.2 Research Method

The research methods compared include SVM using Gaussian radial basis function, ANN, logit. Kim [7] argues that the prediction performance of SVM depends on the values of upper bound C and kernel parameter δ^2. He continues that improper selection of the value of the upper bound C and kernel parameter δ^2 can cause the overfitting and underfitting problems. The appropriate values of the upper bound C and kernel parameter δ^2 are believed to be between 1 and 100 [7, 10]. We also set the C value as 1, 25, 50, 75, 100 and kernel parameter δ^2 as 1, 25, 50, 75, 100. Other options in our analysis use default value of the SVM^{light}. This research varies the number of nodes in the hidden layer such as 3, 5, 10, 15, 20 for training. Options of ANN follow the default values of SPSS Clementine 8.1, such as quick algorithm, Alpha 0.9, Initial Eta 0.3, Eta decay 30, High Eta 0.1, Low Eta 0.01. For logit analysis we use SPSS 11.0 software.

4 Results

Table 2 compares the forecasting performances of logit, ANN, and SVM. As shown, SVM outperforms ANN and logit by 9.86 % and 9.11% for holdout data. The results indicate the feasibility of SVM business forecasting and are compatible with the conclusion of Kim [7].

To test the statistical difference between the accuracy of the models, we employ the McNemar test which is a non-parametric test using Chi-Square distribution. As in Table 3, SVM outperforms ANN and logit at 1% statistical significance level.

Table 2. Average prediction accuracy of logit, ANN, SVM

Model	Logit	ANN	SVM
Training Data	65.30	60.29	71.23
Holdout Data	60.00	59.35	69.11

Table 3. McNemar Values (p value)

Model	ANN	SVM
Logit	0.06 (0.8063)	13.277 (0.0003*)
ANN	-	17.338 (0.0001*)

5 Conclusions

This paper tests the performance of SVM in the prediction of customer churning ratio in on-line auto insurance contracts. To show the superiority of SVM in the prediction level, the results of logit model and ANN method are compared with that of SVM. The results tell that there is a significant statistical difference between SVM and other models. But there is no statistical difference between the logit model and ANN method. This result indicates that SVM can overcome the limitation of the overfitting and partial local minimum problems of ANN model. Also the results tell that SVM is one of excellent alternative to other methods in the prediction of customer churning in on-line auto insurance contracts. As Kim [7] notes, SVM forecasting performance depends on the values of upper bound C and kernel parameter δ. So the controlling method of those two variables remains as a interesting further research topic. Also, we need to study further on Feature Selection, Feature Weighting, Instance Selection using the combination of various machine-learning algorithm.

Acknowledgements

This research was supported by the Chung-Ang University Research Fund in 2005. Correspondence concerning this article should be addressed to slimit@hanmail.net.

References

1. Chu, S.J et al.: Knowledge-Based Methodologies for The Credit Rating: Application and Comparison. Korea Journal of Intelligent Information System, **5** (1999) 49-63
2. Lee, G, C, Jung N, H, Shin K, S.: Customer Churning Analysis by Using Data Mining in Credit Card Market. Korea Journal of Intelligent Information System, **8** (2002) 15-35
3. Tay, F.E.H., Cao, L.: Application of Support Vector Machines in Financial Time Series Forecasting. Omega, **29** (2001) 309-317
4. Guyon, Weston, J., Barnhill, S., Vapnik., V.: Gene Selection for Cancer Classification using Support Vector Machines. Machine Learning, **46** (2002) 389-422

5. Hearst, M, A., Dumaus, S.T, Osman, E., Platt, j., Scholkopf, D.: Support Vector Machines. IEEE Intelligent System, **13** (1998) 18-28
6. Joachims, T.: Text Categorization with Support Vector Machines. Proceedings of 10th the European Conference on Machine Learning (ECML), (1998) 415-422
7. Kim, K.J.: Financial Time Series Forecasting using Support Vector Machines. Neurocomputing, **55** (2003) 307-319
8. Cheung, K.W., Kwok, J.T., Law, M.H., Tsui, K.C.: Mining Customer Product Ratings for Personalized Marketing. Decision Support Systems, **35** (2003) 231-243
9. Osuna, E., Freund, R., Girosi, F.: Training Support Vector Machines and Application to Face Detection. Proceedings of Computer Vision and Pattern Recognition, **1** (1997) 130-136
10. Park Jungmin, Kim Kyung-jae, Han Ingoo.: Bankrupt Foresting using Support Vector Machine. KMIS conference, **1** (2004) 751-758
11. Stephen G. Nash, Areila Sofe.: Linear and Nonlinear Programming. McGraw-Hill, Companies, Inc. (1997)
12. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer-Verlag, New York (1995)
13. Vapnik, V.N.: Statistical Learning Theory. Wiley, New York (1998)
14. Zan Huang, Hsinchun Chen, Chiajune Hsu, Wunhwa Chen, Soushan Wu.: Credit Rating Analysis with Support Vector Machines and Neural Networks: A Market Comparative Study. Decision Support System, **37** (2004) 543-558

Author Index

Image Feature Extraction Based on an Extended Non-negative Sparse Coding Neural Network Model*

Li Shang, Deshuang Huang, Chunhou Zheng, and Zhanli Sun

Abstract. This paper proposes an extended non-negative sparse coding (NNSC) neural network model for natural image feature extraction. The advantage for our model is to be able to ensure to converge to the basis vectors, which can respond well to the edge of the original images. Using the criteria of objective fidelity and the negative entropy, the validity of image feature extraction is testified. Furthermore, compared with independent component analysis (ICA) technique, the experimental results show that the quality of reconstructed images obtained by our method outperforms the ICA method.

* In the original version of this paper the name of the first author was not correct. It should read “Li Shang”. The original online version for this chapter can be found at `http://dx.doi.org/10.1007/11427445_130`