Derong Liu Huaguang Zhang
Marios Polycarpou Cesare Alippi
Haibo He (Eds.)

Advances in Neural Networks – ISNN 2011

LNCS 6676

8th International Symposium on Neural Networks, ISNN 2011
Guilin, China, May/June 2011
Proceedings, Part II

2 Part II

Springer

Lecture Notes in Computer Science 6676

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

Derong Liu Huaguang Zhang
Marios Polycarpou Cesare Alippi
Haibo He (Eds.)

Advances in Neural Networks – ISNN 2011

8th International Symposium
on Neural Networks, ISNN 2011
Guilin, China, May 29 – June 1, 2011
Proceedings, Part II

Springer

Volume Editors

Derong Liu
Chinese Academy of Sciences, Institute of Automation
Key Laboratory of Complex Systems and Intelligence Science
Beijing 100190, China
E-mail: derong.liu@ia.ac.cn

Huaguang Zhang
Northeastern University, College of Information Science and Engineering
Shenyang, Liaoing 110004, China
E-mail: zhanghuaguang@ise.neu.edu.cn

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

Cesare Alippi
Politecnico di Milano, Dip. di Elettronica e Informazione
Piazza L. da Vinci 32, 20133 Milano, Italy
E-mail: alippi@elet.polimi.it

Haibo He
University of Rhode Island
Dept. of Electrical, Computer and Biomedical Engineering
Kingston, RI 02881, USA
E-mail: he@ele.uri.edu

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-21089-1 e-ISBN 978-3-642-21090-7
DOI 10.1007/978-3-642-21090-7
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011926887

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

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

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

Printed on acid-free paper

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

Preface

ISNN 2011 – the 8th International Symposium on Neural Networks – was held in Guilin, China, as a sequel of ISNN 2004 (Dalian), ISNN 2005 (Chongqing), ISNN 2006 (Chengdu), ISNN 2007 (Nanjing), ISNN 2008 (Beijing), ISNN 2009 (Wuhan), and ISNN 2010 (Shanghai). ISNN has now become a well-established conference series on neural networks in the region and around the world, with growing popularity and increasing quality. Guilin is regarded as the most picturesque city in China. All participants of ISNN 2011 had a technically rewarding experience as well as memorable experiences in this great city.

ISNN 2011 aimed to provide a high-level international forum for scientists, engineers, and educators to present the state of the art of neural network research and applications in diverse fields. The symposium featured plenary lectures given by worldwide renowned scholars, regular sessions with broad coverage, and some special sessions focusing on popular topics.

The symposium received a total of 651 submissions from 1,181 authors in 30 countries and regions across all six continents. Based on rigorous reviews by the Program Committee members and reviewers, 215 high-quality papers were selected for publication in the symposium proceedings. We would like to express our sincere gratitude to all reviewers of ISNN 2011 for the time and effort they generously gave to the symposium. We are very grateful to the National Natural Science Foundation of China, the Institute of Automation of the Chinese Academy of Sciences, the Chinese University of Hong Kong, and the University of Illinois at Chicago for their financial support. We would also like to thank the publisher, Springer, for cooperation in publishing the proceedings in the prestigious series of *Lecture Notes in Computer Science*.

Guilin, May 2011

Derong Liu
Huaguang Zhang
Marios Polycarpou
Cesare Alippi
Haibo He

ISNN 2011 Organization

General Chairs

Marios M. Polycarpou	University of Cyprus, Cyprus
Paul J. Werbos	National Science Foundation, USA

Advisory Committee Chairs

Ruwei Dai	Chinese Academy of Sciences, China
Bo Zhang	Tsinghua University, China

Advisory Committee Members

Hojjat Adeli	Ohio State University, USA
Shun-ichi Amari	RIKEN Brain Science Institute, Japan
Dimitri P. Bertsekas	Massachusetts Institute of Technology, USA
Amit Bhaya	Federal University of Rio de Janeiro, Brazil
Tianyou Chai	Northeastern University, China
Guanrong Chen	City University of Hong Kong, Hong Kong
Andrzej Cichocki	RIKEN Brain Science Institute, Japan
Jay Farrell	University of California, Riverside, USA
Russell Eberhart	Indiana University-Purdue University, USA
David B. Fogel	Natural Selection, Inc., USA
Walter J. Freeman	University of California-Berkeley, USA
Kunihiko Fukushima	Kansai University, Japan
Marco Gilli	Politecnico di Torino, Italy
Aike Guo	Chinese Academy of Sciences, China
Zhenya He	Southeast University, China
Tom Heskes	Radboud University Nijmegen, The Netherlands
Janusz Kacprzyk	Polish Academy of Sciences, Poland
Nikola Kasabov	Auckland University of Technology, New Zealand
Okyay Kaynak	Bogazici University, Turkey
Anthony Kuh	University of Hawaii, USA
Deyi Li	National Natural Science Foundation of China, China
Yanda Li	Tsinghua University, China
Chin-Teng Lin	National Chiao Tung University, Taiwan
Robert J. Marks II	Baylor University, USA
Erkki Oja	Helsinki University of Technology, Finland
Nikhil R. Pal	Indian Statistical Institute, India
Jose C. Príncipe	University of Florida, USA

Leszek Rutkowski	Czestochowa University of Technology, Poland
Jennie Si	Arizona State University, USA
Youxian Sun	Zhejiang University, China
DeLiang Wang	Ohio State University, USA
Fei-Yue Wang	Chinese Academy of Sciences, China
Shoujue Wang	Chinese Academy of Sciences, China
Zidong Wang	Brunel University, UK
Cheng Wu	Tsinghua University, Beijing, China
Donald Wunsch II	Missouri University of Science and Technology, USA
Lei Xu	The Chinese University of Hong Kong, Hong Kong
Shuzi Yang	Huazhong University of Science and Technology, China
Xin Yao	University of Birmingham, UK
Gary G. Yen	Oklahoma State University, USA
Nanning Zheng	Xi'An Jiaotong University, China
Jacek M. Zurada	University of Louisville, USA

Steering Committee Chair

Jun Wang	Chinese University of Hong Kong, Hong Kong

Steering Committee Members

Jinde Cao	Southeast University, China
Shumin Fei	Southeast University, China
Min Han	Dalian University of Technology, China
Xiaofeng Liao	Chongqing University, China
Bao-Liang Lu	Shanghai Jiao Tong University, China
Yi Shen	Huazhong University of Science and Technology, China
Fuchun Sun	Tsinghua University, China
Hongwei Wang	Huazhong University of Science and Technology, China
Zongben Xu	Xi'An Jiaotong University, China
Zhang Yi	Sichuan University, China
Wen Yu	National Polytechnic Institute, Mexico

Organizing Committee Chairs

Derong Liu	Chinese Academy of Sciences, China
Huaguang Zhang	Northeastern University, China

Program Chairs

Cesare Alippi	Politecnico di Milano, Italy
Bhaskhar DasGupta	University of Illinois at Chicago, USA
Sanqing Hu	Hangzhou Dianzi University, China

Plenary Sessions Chairs

Frank L. Lewis	University of Texas at Arlington, USA
Changyin Sun	Southeast University, China

Special Sessions Chairs

Amir Hussain	University of Stirling, UK
Jinhu Lu	Chinese Academy of Sciences, China
Stefano Squartini	Università Politecnica delle Marche, Italy
Liang Zhao	University of Sao Paulo, Brazil

Finance Chairs

Hairong Dong	Beijing Jiaotong University, China
Cong Wang	South China University of Technology, China
Zhigang Zeng	Huazhong University of Science and Technology, China
Dongbin Zhao	Chinese Academy of Sciences, China

Publicity Chairs

Zeng-Guang Hou	Chinese Academy of Sciences, China
Manuel Roveri	Politecnico di Milano, Italy
Songyun Xie	Northwestern Polytechnical University, China
Nian Zhang	University of the District of Columbia, USA

European Liaisons

Danilo P. Mandic	Imperial College London, UK
Alessandro Sperduti	University of Padova, Italy

Publications Chairs

Haibo He	Stevens Institute of Technology, USA
Wenlian Lu	Fudan University, China
Yunong Zhang	Sun Yat-sen University, China

Registration Chairs

Xiaolin Hu	Tsinghua University, China
Zhigang Liu	Southwest Jiaotong University, China
Qinglai Wei	Chinese Academy of Sciences, China

Local Arrangements Chairs

Xuanju Dang	Guilin University of Electronic Technology, China
Xiaofeng Lin	Guangxi University, China
Yong Xu	Guilin University of Electronic Technology, China

Electronic Review Chair

Tao Xiang	Chongqing University, China

Symposium Secretariat

Ding Wang	Chinese Academy of Sciences, China

ISNN 2011 International Program Committee

Jose Aguilar	Universidad de los Andes, Venezuela
Haydar Akca	United Arab Emirates University, UAE
Angelo Alessandri	University of Genoa, Italy
Luís Alexandre	Universidade da Beira Interior, Portugal
Bruno Apolloni	University of Milan, Italy
Marco Antonio Moreno Armendáriz	Instituto Politecnico Nacional, Mexico
K. Vijayan Asari	Old Dominion University, USA
Amir Atiya	Cairo University, Egypt
Monica Bianchini	Università degli Studi di Siena, Italy
Salim Bouzerdoum	University of Wollongong, Australia
Ivo Bukovsky	Czech Technical University, Czech Republic
Xindi Cai	APC St. Louis, USA
Jianting Cao	Saitama Institute of Technology, Japan
M. Emre Celebi	Louisiana State University, USA
Jonathan Hoyin Chan	King Mongkut's University of Technology, Thailand
Ke Chen	University of Manchester, UK
Songcan Chen	Nanjing University of Aeronautics and Astronautics, China
YangQuan Chen	Utah State University, USA
Yen-Wei Chen	Ritsumeikan University, Japan
Zengqiang Chen	Nankai University, China

Jianlin Cheng	University of Missouri Columbia, USA
Li Cheng	NICTA Australian National University, Australia
Long Cheng	Chinese Academy of Sciences, China
Xiaochun Cheng	University of Reading, UK
Sung-Bae Cho	Yonsei University, Korea
Pau-Choo Chung	National Cheng Kung University, Taiwan
Jose Alfredo Ferreira Costa	Federal University, UFRN, Brazil
Sergio Cruces-Alvarez	University of Seville, Spain
Lili Cui	Northeastern University, China
Chuangyin Dang	City University of Hong Kong, Hong Kong
Xuanju Dang	Guilin University of Electronic Technology, China
Mingcong Deng	Okayama University, Japan
Ming Dong	Wayne State University, USA
Gerard Dreyfus	ESPCI-ParisTech, France
Haibin Duan	Beihang University, China
Wlodzislaw Duch	Nicolaus Copernicus University, Poland
El-Sayed El-Alfy	King Fahd University of Petroleum and Minerals, Saudi Arabia
Pablo Estevez	Universidad de Chile, Chile
Jay Farrell	University of California Riverside, USA
Wai-Keung Fung	University of Manitoba, Canada
John Gan	University of Essex, UK
Junbin Gao	Charles Sturt University, Australia
Xiao-Zhi Gao	Helsinki University of Technology, Finland
Anya Getman	University of Nevada Reno, USA
Xinping Guan	Shanghai Jiao Tong University, China
Chengan Guo	Dalian University of Technology, China
Lejiang Guo	Huazhong University of Science and Technology, China
Ping Guo	Beijing Normal University, China
Qing-Long Han	Central Queensland University, Australia
Haibo He	Stevens Institute of Technology, USA
Zhaoshui He	RIKEN Brain Science Institute, Japan
Tom Heskes	Radboud University Nijmegen, The Netherlands
Zeng-Guang Hou	Chinese Academy of Sciences, China
Zhongsheng Hou	Beijing Jiaotong University, China
Chun-Fei Hsu	Chung Hua University, Taiwan
Huosheng Hu	University of Essex, UK
Jinglu Hu	Waseda University, Japan
Guang-Bin Huang	Nanyang Technological University, Singapore
Ting Huang	University of Illinois at Chicago, USA
Tingwen Huang	Texas A&M University at Qatar
Marc Van Hulle	Katholieke Universiteit Leuven, Belgium
Amir Hussain	University of Stirling, UK
Giacomo Indiveri	ETH Zurich, Switzerland

Danchi Jiang	University of Tasmania, Australia
Haijun Jiang	Xinjiang University, China
Ning Jin	University of Illinois at Chicago, USA
Yaochu Jin	Honda Research Institute Europe, Germany
Joarder Kamruzzaman	Monash University, Australia
Qi Kang	Tongji University, China
Nikola Kasabov	Auckland University, New Zealand
Yunquan Ke	Shaoxing University, China
Rhee Man Kil	Korea Advanced Institute of Science and Technology, Korea
Kwang-Baek Kim	Silla University, Korea
Sungshin Kim	Pusan National University, Korea
Arto Klami	Helsinki University of Technology, Finland
Leo Li-Wei Ko	National Chiao Tung University, Taiwan
Mario Koeppen	Kyuhsu Institute of Technology, Japan
Stefanos Kollias	National Technical University of Athens, Greece
Sibel Senan Kucur	Istanbul University, Turkey
H.K. Kwan	University of Windsor, Canada
James Kwok	Hong Kong University of Science and Technology, Hong Kong
Edmund M.K. Lai	Massey University, New Zealand
Chuandong Li	Chongqing University, China
Kang Li	Queen's University Belfast, UK
Li Li	Tsinghua University, China
Michael Li	Central Queensland University, Australia
Shaoyuan Li	Shanghai Jiao Tong University, China
Shutao Li	Hunan University, China
Xiaoou Li	CINVESTAV-IPN, Mexico
Yangmin Li	University of Macao, Macao
Yuanqing Li	South China University of Technology, China
Hualou Liang	University of Texas at Houston, USA
Jinling Liang	Southeast University, China
Yanchun Liang	Jilin University, China
Lizhi Liao	Hong Kong Baptist University
Alan Wee-Chung Liew	Griffith University, Australia
Aristidis Likas	University of Ioannina, Greece
Chih-Jen Lin	National Taiwan University, Taiwan
Ju Liu	Shandong University, China
Meiqin Liu	Zhejiang University, China
Yan Liu	Motorola Labs, Motorola, Inc., USA
Zhenwei Liu	Northeastern University, China
Bao-Liang Lu	Shanghai Jiao Tong University, China
Hongtao Lu	Shanghai Jiao Tong University, China
Jinhu Lu	Chinese Academy of Sciences, China
Wenlian Lu	Fudan University, China

Yanhong Luo	Northeastern University, China
Jinwen Ma	Peking University, China
Malik Magdon-Ismail	Rensselaer Polytechnic Institute, USA
Danilo Mandic	Imperial College London, UK
Francesco Marcelloni	University of Pisa, Italy
Francesco Masulli	Università di Genova, Italy
Matteo Matteucci	Politecnico di Milano, Italy
Patricia Melin	Tijuana Institute of Technology, Mexico
Dan Meng	Southwest University of Finance and Economics, China
Yan Meng	Stevens Institute of Technology, USA
Valeri Mladenov	Technical University of Sofia, Bulgaria
Roman Neruda	Academy of Sciences of the Czech Republic, Czech Republic
Ikuko Nishikawa	Ritsumei University, Japan
Erkki Oja	Aalto University, Finland
Seiichi Ozawa	Kobe University, Japan
Guenther Palm	Universität Ulm, Germany
Christos Panayiotou	University of Cyprus, Cyprus
Shaoning Pang	Auckland University of Technology, New Zealand
Thomas Parisini	University of Trieste, Italy
Constantinos Pattichis	University of Cyprus, Cyprus
Jaakko Peltonen	Helsinki University of Technology, Finland
Vincenzo Piuri	University of Milan, Italy
Junfei Qiao	Beijing University of Technology, China
Manuel Roveri	Politecnico di Milano, Italy
George Rovithakis	Aristole University of Thessaloniki, Greece
Leszek Rutkowski	Technical University of Czestochowa, Poland
Tomasz Rutkowski	RIKEN Brain Science Institute, Japan
Sattar B. Sadkhan	University of Babylon, Iraq
Toshimichi Saito	Hosei University, Japan
Karl Sammut	Flinders University, Australia
Edgar Sanchez	CINVESTAV, Mexico
Marcello Sanguineti	University of Genoa, Italy
Gerald Schaefer	Aston University, UK
Furao Shen	Nanjing University, China
Daming Shi	Nanyang Technological University, Singapore
Hideaki Shimazaki	RIKEN Brain Science Institute, Japan
Qiankun Song	Chongqing Jiaotong University, China
Ruizhuo Song	Northeastern University, China
Alessandro Sperduti	University of Padua, Italy
Stefano Squartini	Università Politecnica delle Marche, Italy
Dipti Srinivasan	National University of Singapore, Singapore
John Sum	National Chung Hsing University, Taiwan
Changyin Sun	Southeast University, China

Johan Suykens	Katholieke Universiteit Leuven, Belgium
Roberto Tagliaferri	University of Salerno, Italy
Norikazu Takahashi	Kyushu University, Japan
Ah-Hwee Tan	Nanyang Technological University, Singapore
Ying Tan	Peking University, China
Toshihisa Tanaka	Tokyo University of Agriculture and Technology, Japan
Hao Tang	Hefei University of Technology, China
Qing Tao	Chinese Academy of Sciences, China
Ruck Thawonmas	Ritsumeikan University, Japan
Sergios Theodoridis	University of Athens, Greece
Peter Tino	Birmingham University, UK
Christos Tjortjis	University of Manchester, UK
Ivor Tsang	Nanyang Technological University, Singapore
Masao Utiyama	National Institute of Information and Communications Technology, Japan
Marley Vellasco	PUC-Rio, Brazil
Alessandro E.P. Villa	Université de Lausanne, Switzerland
Draguna Vrabie	University of Texas at Arlington, USA
Bing Wang	University of Hull, UK
Dan Wang	Dalian Maritime University, China
Dianhui Wang	La Trobe University, Australia
Ding Wang	Chinese Academy of Sciences, China
Lei Wang	Australian National University, Australia
Lei Wang	Tongji University, China
Wenjia Wang	University of East Anglia, UK
Wenwu Wang	University of Surrey, USA
Yingchun Wang	Northeastern University, China
Yiwen Wang	Hong Kong University of Science and Technology, Hong Kong
Zhanshan Wang	Northeastern University, China
Zhuo Wang	University of Illinois at Chicago, USA
Zidong Wang	Brunel University, UK
Qinglai Wei	Chinese Academy of Sciences, China
Yimin Wen	Hunan Institute of Technology, China
Wei Wu	Dalian University of Technology, China
Cheng Xiang	National University of Singapore, Singapore
Degui Xiao	Hunan University, China
Songyun Xie	Northwestern Polytechnical University, China
Rui Xu	Missouri University of Science and Technology, USA
Xin Xu	National University of Defense Technology, China
Yong Xu	Guilin University of Electronic Technology, China
Jianqiang Yi	Chinese Academy of Sciences, China
Zhang Yi	Sichuan University, China

Dingli Yu	Liverpool John Moores University, UK
Xiao-Hua Yu	California Polytechnic State University, USA
Xiaoqin Zeng	Hohai University, China
Zhigang Zeng	Huazhong University of Science and Technology, China
Changshui Zhang	Tsinghua University, China
Huaguang Zhang	Northeastern University, China
Jianghai Zhang	Hangzhou Dianzi University, China
Jie Zhang	University of New Castle, UK
Kai Zhang	Lawrence Berkeley National Lab, USA
Lei Zhang	Sichuan University, China
Nian Zhang	University of the District of Columbia, USA
Xiaodong Zhang	Wright State University, USA
Xin Zhang	Northeastern University, China
Yunong Zhang	Sun Yat-sen University, China
Dongbin Zhao	Chinese Academy of Sciences, China
Hai Zhao	Shanghai Jiao Tong University, China
Liang Zhao	University of São Paulo, Brazil
Mingjun Zhong	University of Glasgow, UK
Weihang Zhu	Lamar University, USA
Rodolfo Zunino	University of Genoa, Italy

Table of Contents – Part II

Supervised Learning and Unsupervised Learning

Kernel Methods and Support Vector Machines

Mixture Models and Clustering

Visual Perception and Pattern Recognition

Motion, Tracking and Object Recognition

Natural Scene Analysis and Speech Recognition

Neuromorphic Hardware, Fuzzy Neural Networks and Robotics

Multi-agent Systems and Adaptive Dynamic Programming

Maximum Variance Sparse Mapping

Bo Li[1,2], Jin Liu[1], and Wenyong Dong[3]

[1] State Key Lab. of Software Engineering, Wuhan University, 430072, Wuhan, China
[2] College of Computer Science and Technology,
Wuhan University of Science and Technology, Wuhan, China
[3] Computer School, Wuhan University, 430072, Wuhan, China

Abstract. In this paper, a multiple sub-manifold learning method oriented classification is presented via sparse representation, which is named maximum variance sparse mapping. Based on the assumption that data with the same label locate on a sub-manifold and different class data reside in the corresponding sub-manifolds, the proposed algorithm can construct an objective function which aims to project the original data into a subspace with maximum sub-manifold distance and minimum manifold locality. Moreover, instead of setting the weights between any two points directly or obtaining those by a square optimal problem, the optimal weights in this new algorithm can be approached using L1 minimization. The proposed algorithm is efficient, which can be validated by experiments on some benchmark databases.

Keywords: MVSM, sub-manifold, Sparse representation.

1 Introduction

Recently, feature extraction methods based on manifold learning have been attracting much attention. Among these original manifold learning methods and their extensions, one representative is Laplacian Eigenmaps (LE) [1], which is based on graph mapping method. LE constructs the nearest graph with K Nearest Neighbors (KNN) criterion on the training data points and sets the weights between any two points either belonging to K Nearest Neighbors or not respectively. Then an objective function can be formed which connections to the graph Laplacian, the Laplacian Beltrami operator on the manifold and the heat equation. LE seeks the optimal feature subspace by solving the objective function where locality can be preserved. However, LE is a nonlinear dimensionality reduction approach with less generalization ability. That is to say, the image of a test in low dimensional space can not be easily acquired with the projection results of the training set, which is also entitled out-of-sample problem [2]. Linearization, kernelization and tensorization are some often used techniques to overcome the problem [3]. For example, Locality Preserving Projection (LPP) [4, 5] is a linear approximation of LE. It is the linearization to LE that the LPP algorithm shows its merits on favorable clustering and low computational cost [5]. In most methods, the linear transformation matrix is also under the orthogonal constraint to reduce the redundancy of the data. However, this constraint does not defined in LPP. In order to solve the

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 1–8, 2011.

problem, an Orthogonal LPP (OLPP) algorithm is presented by Cai, which shows more discriminating than LPP [6]. Recently, Yang also proposed an Unsupervised Discriminant Projection (UDP) algorithm [7] to consider both manifold locality and non-locality. Later, UDP can be viewed as a simplified version of LPP under constraint that the local density is uniform [8]. After characterizing the local scatter and the non-local scatter, UDP aims to look for a linear projection to maximize non-local scatter and to minimize the local scatter simultaneously. Therefore UDP is more intuitive and more powerful than most of up-to-date methods. However, it must be noted that UDP is a linear approximation to the manifold learning approaches without any class information involved. Nevertheless, the class information has been considered to have much to do with discriminating features for classification. So combining to UDP, Li proposed an Orthogonal Discriminant Projection (ODP) by taking the data labels into account [9]. All the methods mentioned above are linear versions of the original LE with all kinds of constraints, so there are some points in common for them. First, Unlike Locally Linear Embedding (LLE) [10,11], which obtains the reconstruction weights by solving a square optimal problem, either LE or its extensions including LPP, OLPP, UDP and ODP set the weights between points simply 0, 1 or the value of a function, thus the weights can not be always optimal. Second, the weights are not robust to noise and outlier. Third, most of the mentioned methods pay more attention to the locality preserving and lose sight of the class information.

In this paper, a new feature extraction method, named Maximum Variance Sparse Mapping (MVSM), is proposed to overcome the problems mentioned above. Making full consideration of class information, a multiple sub-manifold model for classification is constructed, then the weights between K nearest neighbors can be gained by a sparse representation with L1 normalization, which will be robust to noise and outlier. The aim of MVSM is to locate the original data on a subspace where the distance between sub-manifolds is maximized and the sparse locality is minimized.

2 The Proposed Algorithm

In proposed method, a multiple sub-manifold classification model is proposed, which make use of labels to construct sub-manifolds. In other words, data with the same label should locate on a sub-manifold and different class data should reside in the corresponding sub-manifolds. Moreover, the sub-manifolds distance is defined based on the class information. At the same time, a spare optimized objective function is adopted to compute the optimal sparse weights, which is robust to noisy and outlier [12]. And then the locality of sub-manifolds can be determined with the optimal sparse weights. In the following, definitions of the sub-manifolds distance, the sparse optimal weights and the locality of sub-manifolds are described, respectively.

2.1 The Sub-manifolds Distance

In most algorithms, Euclidean distance is often used to measure similarity. If the Euclidean distance between two points is very large, it can be said that these two points

will be of high dissimilarity. Otherwise, it will probably be similar to each other. Therefore, Euclidean distance can also be taken use of scaling the distance of sub-manifolds. However, it must be noted that the dissimilarities also exist between different manifolds. How to distinguish one manifold from the others will heavily depend on the manifold class labels. Based on the multiple sub-manifolds model, the data distributed on a manifold are belonging to the same class. So a label matrix H can be designed to mark the class relation between points. The label matrix H is stated as follows:

$$H_{ij} = \begin{cases} 0 & \text{if } X_i \text{ and } X_j \text{have the same class label} \\ 1 & otherwise \end{cases} \tag{1}$$

Based on the label matrix H , we define the sub-manifolds distance to be the sum of the squared distance between sub-manifolds as follows, which can also be found in Constrained Maximum Variance Mapping (CMVM) [13].

$$\begin{aligned} J_D &= \sum_{i,j}^{n} H_{ij}(Y_i - Y_j)(Y_i - Y_j)^T \\ &= 2\sum_i Y_i Q_{ii} Y_i^T - 2\sum_{ij} Y_i H_{ij} Y_j^T = 2Y(Q-H)Y^T \end{aligned} \tag{2}$$

where $Q_{ii} = \sum_j H_{ij}$

2.2 The Weights through Sparse Representation

In the past few years, sparse representations of signals have received a great deal of attentions, which is initially proposed as a substitute to traditional signal processing methods such as Fourier and wavelet. The problem solved by sparse representation is to search for the most compact representation of a signal in terms of linear combination of atoms in an over-complete dictionary. Compared to methods based on orthonornal transforms or direct time domain processing, sparse representation usually offers better performance with its capacity for efficient signal modeling. In the proposed algorithm, the sparse representation is introduced to compute the weights between points in sub-manifolds instead of directly setting them a simply value or result of a heat kernel function. In the proposed MVSM, a point can be linear representation by data with the same labels contained in k nearest neighbors, that is to say, the k nearest neighbors must be selected from data belonging to the same class, thus an objective function can be constructed as follows:

$$\begin{aligned} S_i &= \min \|S_i\|_1 \\ s.t.\ X_i &= XS_i \end{aligned} \tag{3}$$

Or

$$\begin{aligned} &S_i = \min \left\| S_i \right\|_1 \\ &s.t.\ \left\| X_i - XS_i \right\| < \varepsilon \end{aligned} \tag{4}$$

Where S_i denotes the weighted vector by representing X_i with its K nearest neighbors belonging to the same class.

The l_1 minimization problem can be solved by LASSO [14] or LARS [15]. Thus repeat l_1 minimization problem to all the points, the sparse weights matrix can be expressed to $S = [S_1, S_2, ..., S_n]$.

2.3 The Locality of Sub-manifolds

Based on the above discuss, on the one hand, the sparse weight matrix S can reflect intrinsic geometric information because it can be obtained by K nearest neighbors; on the other hand it also offer much discriminant information due to the K nearest neighbors all with the same class label. So using the sparse weight matrix S, the locality of sub-manifolds can be deduced via the following sparse representation.

$$\begin{aligned} J_L &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} S_{ij}\left(Y_i - Y_j\right)^2 = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} S_{ij}\left(X_i - X_j\right)\left(X_i - X_j\right)^T \\ &= \sum_{ij} X_i S_{ij} X_i^T - \sum_{ij} X_i S_{ij} X_j^T = \sum_{i} X_i D_{ii} X_i^T - \sum_{ij} X_i S_{ij} X_j^T \\ &= X(D-S)X^T = XLX^T \end{aligned} \tag{5}$$

Where diagonal matrix $D_{ii} = \sum_j S_{ij}$ and $L = D - S$

2.4 Justification

In the above Subsections, the sub-manifolds distance and the locality of sub-manifolds via sparse representation have been offered. However, MVSM is a manifold learning based method which inevitable encounters out-of-sample problem and small sample size problem. So if there is a linear transformation which can project the original data into a subspace with the maximum sub-manifold distance, it not only naturally overcomes out-of-sample problem but also shows its superiority to classification; secondly, due to introducing a linear transformation, minimizing the locality of sub-manifolds means to map the original data into a subspace where manifold geometry can be well preserved with less computational cost. So if the sub-manifold distance can be maximized and the locality of sub-manifolds can be minimized by a linear transformation simultaneously, we will find an optimal linear subspace with higher recognition rate and lower computational cost.

Provided that the linear features Y can be obtained by a linear transformation, i.e., $Y_i = A^T X_i$. Then J_L and J_D can be rewritten into the following forms:

$$J_L = A^T XLX^T A \tag{6}$$

$$J_D = A^T X(Q-H)X^T A \tag{7}$$

According to the motivation mentioned above, the corresponding objective function can be represented as follows:

$$J(A) = \max_A \frac{J_D}{J_L} = \max_A \frac{A^T X(Q-H)X^T A}{A^T XLX^T A} \tag{8}$$

This optimization problem can be figured out by enforcing the following Lagrange multiplier:

$$L(A) = \frac{A^T X(Q-H)X^T A}{A^T XLX^T A} \tag{9}$$

Then, the optimal transformation matrix A can be obtained from the following expression:

$$\frac{\partial L(A,\lambda)}{\partial A} = 0 \tag{10}$$

At last we have:

$$X(Q-H)X^T A_i = \lambda_i A^T XLX^T A_i \tag{11}$$

So it can be found that A is composed of the eigenvectors associated with the d top eigenvalues by solving the above generalized eigen-equation. However, it must be noted that if matrix, i.e. XLX^T, is singular, Eqn. (16) has no solution. The case always exists in real-word applications when sample numbers is less than the dimensions of the original data. Thus in order to avoid the problem, a preprocessing method can be performed when encountering the case mentioned above. Generally speaking, the dimensions of the original data can be reduced to some extent to ensure the matrix XLX^T to be positively definite. In this study, PCA will be adopted because of its simplicity. After the proposed algorithm is applied to the preprocessed data, the transformation matrix including the preprocessing can be expressed as follows:

$$A = A_{PCA} A_{MVSM} \tag{12}$$

3 Experiments

In this Section, ORL face data, YALE face data and Palmprint [17]are applied to evaluate the performance of the proposed MVSM algorithm, which is compared with those of LPP, CMVM and Sparsity Preserving Projection (SPP) [16].

3.1 Experiment on ORL Face Database

The ORL database contains 400 images of 40 individuals, including variation in facial expression, pose and other details like glasses/non-glasses. Fig.1 illustrates a sample subject of the ORL database along with all ten views, which are cropped to 32×32 size. For each person, the first 5 images are selected for training and the rest are used for testing. For MVSM, LPP, CMVM and SPP, the dimension of preprocessed data obtained is $n-c$ dimensions and 100 percent image energy is kept when performing PCA, where c denotes the class number. Then k nearest neighbor criterion is adopted to construct the adjacency graph and k is set to 4. At last, the nearest neighbor classifier is also taken to classify the test data.

From Table 1, it can be found that MVSM obtains the lowest error rate compared with LPP,CMVM and SPP.

Fig. 1. The cropped sample face images from ORL database

Table 1. Performance comparison by using MVSM, LPP, CMVM and SPP on ORL face

Methods	LPP	CMVM	SPP	MVSM
Recognition rate	95.6%	97.8%	96.7%	98.9%
Dimensions	24	26	42	30

3.2 Experiment on Tumor Gene Expressive Data

In this subsection, we used another two tumor gene expressive data datasets for experimentation. One is the Leukemia dataset [17]. We randomly selected 10 cases of ALL_B, 4 cases of ALL_T and 6 cases of AML as the training set, and use the rest samples as test data. Another dataset is the central nervous system tumors dataset [18], which is composed of four types of central nervous system embryonal tumors. We randomly selected 5 medulloblasomas, 5 malignant gliomas, 5 rhabdoids and 3 normals as training set, and use the rest samples as test data, at last KNN is adopted to classify the features extracted by LPP, CMVM, SPP and MVSM. These experimental results are displayed in Table 2. It can be found that the proposed method gains the best results.

Table 2. The multi-class gene expressive data classification results by different methods

Methods	Leukemia dataset		Central Nervous System Tumors	
	Accuracy(%)	Dimensions	Accuracy(%)	Dimensions
LPP	95.21	6	92.33	6
CMVM	98.33	2	94.38	3
SPP	97.66	3	93.97	4
MVSM	99.12	4	95.56	3

4 Conclusion

In this paper, a multiply sub-manifold learning method via sparse representation, namely MVSM, is proposed for classification. The proposed algorithm uses the sparse local information to construct the locality of sub-manifolds as well as the class information of the data to model the sub-manifold learning. So the proposed algorithm becomes more suitable for the tasks of classification. This result is validated either from the theoretical analysis or from experiments on real-world data set.

Acknowledgment. This work was supported by the grants of the National Natural Science Foundation of China (61070013, 60703018, 61070012&90924026), 973 Program (2007CB310800), Twelfth Five Years Plan Key National Project(GFZX0101050302), 863 Program (2008AA022503, 2008AA01Z208, 2009AA01Z405), the Science and Technology Commission of Wuhan Municipality "Chenguang Jihua" (201050231058), the 111 Project (B07037), Postdoctoral Science Foundation of China (20100470613), Natural Science Foundation of Hubei Province (2010CDB03302), Shanghai Key Laboratory of Intelligent Information Processing, China (IIPL-2010-004), the Open Fund Project of State Key Lab. of Software Engineering (Wuhan University, China, SKLSE08-11) and Science Foundation of Wuhan University of Science and Technology(2010XG7&2010XZ015).

References

1. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput. 15(6), 1373–1396 (2003)
2. Bengio, Y., Paiement, J.-F., Vincent, P.: Out-of-sample extensions for LLE, Isomap, MDS, eigenmaps, and spectral clustering, Technical Report 1238, Universit' e deMontreal (2003)
3. Yan, S., Xu, D., Zhang, B., Zhang, H.-J.: Graph Embedding: A General Framework for Dimensionality Reduction. IEEE Trans. Pattern Analysis and Machine Intelligence 29(1), 40–51 (2007)
4. He, X., Yang, S., Hu, Y., Niyogi, P., Zhang, H.J.: Face Recognition Using Laplacianfaces. IEEE Trans. Pattern Analysis and Machine Intelligence 27(3), 328–340 (2005)
5. He, X., Niyogi, P.: Locality preserving projections. In: Neural Information Processing Systems, NIPS 2003, Vancouver, Canada, vol. 16 (2003)
6. Cai, D., He, X., Han, J., Zhang, H.: Orthogonal Laplacianfaces for Face Recognition. IEEE Trans. on Image Processing 15(11), 3609–3614 (2006)

7. Yang, J., Zhang, D., Yang, J.Y., Niu, B.: Globally Maximizing, Locally Minimizing: Unsupervised Discriminant Projection with Application to Face and Palm Biometrics. IEEE Trans. Pattern Analysis and Machine Intelligence 29(4), 650–664 (2007)
8. Deng, W., Hu, J., Guo, J., Zhang, H., Zhang, C.: Comments on Globally Maximizing, Locally Minimizing: Unsupervised Discriminant Projection with Application to Face and Palm Biometrics. IEEE Trans. Pattern Analysis and Machine Intelligence (accepted)
9. Li, B., Wang, C., Huang, D.-S.: Supervised feature extraction based on orthogonal discriminant projection. Neurocomputing 73(1-3), 191–196 (2009)
10. Saul, L.K., Roweis, S.T.: Think globally, fit locally: unsupervised learning of low dimensional manifolds. J. Mach. Learning Res. 4, 119–155 (2003)
11. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326 (2000)
12. Wright, J., Yang, A., Sastry, S., Ma, Y.: Robust face recognition via sparse representation. IEEE Trans. Pattern Anal. Mach. Intell. 31(2), 210–227 (2009)
13. Li, B., Huang, D.-S., Wang, C., Liu, K.-H.: Feature extraction using constrained maximum variance mapping. Pattern Recognition 41(11), 3287–3294 (2008)
14. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society 58(1), 267–288 (1996)
15. Drori, I., Donoho, D.: Solution of L1 minimization problems by LARS/Homotopy methods. In: ICASSP, vol. 3, pp. 636–639 (2006)
16. Qiao, L., Chen, S., Tan, X.: Sparsity preserving projections with applications to face recognition. Pattern Recognition 43(1), 331–341 (2010)
17. Brunet, J.P., Tamayo, P., Golun, T.R., Mesirov, J.P.: Metagenes and Molecular Pattern Discovery Using Matrix Factorization. Proc. Natl. Acad. Sci. 101, 4164–4416 (2004)
18. Pomeroy, S.L., Tamayo, P., et al.: Prediction of Central Nervous System Embryonal Tumour Outcome Based on Gene Expression. Nature 415, 436–442 (2002)

Contourlet-Based Texture Classification with Product Bernoulli Distributions

Yongsheng Dong and Jinwen Ma*

Department of Information Science, School of Mathematical Sciences and LMAM
Peking University, Beijing, 100871, China

Abstract. In this paper, we propose a novel texture classification method based on product Bernoulli distributions (PBD) and contourlet transform. In particular, product Bernoulli distributions (PBD) are employed for modeling the coefficients in each contourlet subband of a texture image. By investigating these bit-plane probabilities (BPs), we use the weighted L_1-norm to discriminate the bit-plane probabilities of the corresponding subbands of two texture images and establish a new distance between the two images. Moreover, the K-nearest neighbor classifier is utilized to perform supervised texture classification. It is demonstrated by the experiments that our proposed method outperforms some current state-of-the-art approaches.

Keywords: Texture classification, Contourlet transform, Product Bernoulli distributions(PBD), Bit-plane probability (BP).

1 Introduction

Texture classification plays an important role in computer vision with a wide variety of applications. Examples include classification of regions in satellite images, automated inspection, medical image analysis and document image processing. During the last three decades, numerous methods have been proposed for image texture classification and retrieval [1]-[13]. Among these approaches, wavelet-based methods may be the most popular due to the multiresolution and orientation representation of wavelets which is consistent with the human visual system [10].

However, two-dimensional wavelets are only good at catching point discontinuities but do not capture the geometrical smoothness of the contours. As a newly developed two-dimensional extension of the wavelet transform using multiscale and directional filter banks, the contourlet transform can effectively capture the intrinsic geometrical structure that is key in visual information, because the contourlet expansion can achieve the optimal approximation rate for piecewise smooth functions with C^2 contours in some sense [14]. Recently, the contourlet transform has been successfully used in content-based texture retrieval [15], palmprint classification and handwritten numeral recognition [16].

* Corresponding author, jwma@math.pku.edu.cn

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 9–18, 2011.

Of course, the contourlet transform also provides us an efficient tool to extract the features from a texture image for texture classification.

Modeling wavelet detail subband coefficients via the Product Bernoulli Distributions (PBD) [11]-[12] has received a lot of interest. The PBD model makes use of a binary bit representation for wavelet subband histograms and the so-called Bit-plane Probability (BP) signature is constructed based on the model parameters. Essentially, the main merits of BP approach are its efficiency for signature extraction and similarity measurement based on Euclidean metric, and the statistical justification of the model parameters for use in image processing applications [10]-[11]. However, it has two main disadvantages. First, the wavelet transform used in [11] cannot capture directional information and then the wavelet coefficients don't represent a texture image well. So the recognition performance is not satisfying. Second, the minimum distance classifier used in [11] doesn't work well because the BP signature is obtained by concatenating all the bit-plane probabilities of all high-pass subbands, and the distance between a new image and a texture class is obtained by the $weighted - L_1$ distance of the BP signature of the test sample and the mean of the BP signatures of all training samples in each class.

Motivated by the advantages and disadvantages of PBD, we propose a new method for texture classification using contourlet transform and PBD together. More specifically, this paper makes the following contributions. First, we use product Bernoulli distributions to model the contourlet coefficients instead of wavelet coefficients. Second, we present a new distance of two images, which is measured by summing up all the $weighted - L_1$ metrics of the bit-plane probabilities of the corresponding subbands. Finally, we apply the PBD model in the contourlet domain to supervised texture classification through the K-nearest neighbor classifier, and experimental results on large texture datasets reveal that our proposed method with the use of the new distance performs better than the method based on the PBD in the wavelet domain [11], and outperforms the current state-of-the-art method based on M-band ridgelet transform [17].

The rest of the paper is organized as follows. Section 2 introduces the contourlet transform. In Section 3, we present a new texture classification method based on the product Bernoulli distributions in the contourlet domain. Experimental results on three large datasets are conducted in Section 4 to demonstrate the effectiveness of our proposed texture classification method. Finally, we conclude briefly in Section 5.

2 Contourlet Transform

The contourlet transform was recently developed by Do and Vetterli [14] in order to get rid of the limitations of wavelets. Actually, they utilized a double filter bank structure in which at first the Laplacian pyramid (LP) [18] is used to capture the point discontinuities, and then a directional filter bank (DFB) [19] is used to link point discontinuities into a linear structure. So, the overall result of such a transform is based on an image expansion with basis elements like contour segments, and thus it is referred to as the contourlet transform.

Due to its cascade structure accomplished by combining the Laplacian pyramid (LP) with a directional filter bank (DFB) at each scale, multiscale and directional decomposition stages in the contourlet transform are independent of each other. Therefore, one can decompose each scale into any arbitrary power of two's number of directions, and different scales can be decomposed into different numbers of directions. Therefore, it can represent smooth edges in the manner of being close to the optimal efficiency. Fig. 1 shows an example of the contourlet transform on the "Barbara" image. For the visual clarity, only two-scale decompositions are shown. The image is decomposed into two pyramidal levels, which are then decomposed into four and eight directional subbands, respectively.

More recent developments and applications on the contourlet transform can be found in [15], [16] and [20].

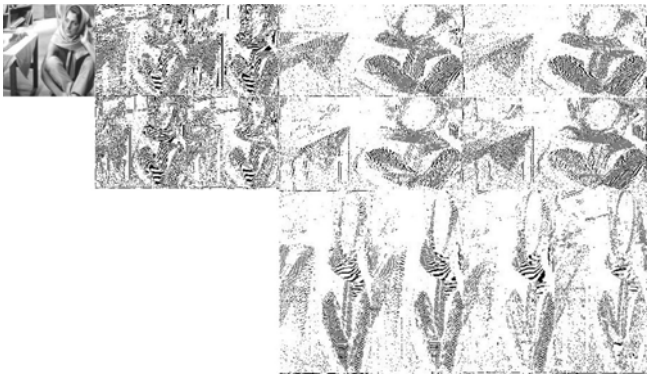


Fig. 1. The result of the Contourlet transform of the "Barbara" image

3 Proposed Texture Classification Method

3.1 Product Bernoulli Distributions

For L-scale contourlet decompositions of a given texture image, we find the average amplitude of the coefficients increases almost exponentially with the scale i ($i = 1, 2 \cdots, L$). Hence, to model the contourlet coefficients at different scales uniformly, we regularize them by multiplying the factor $1/2^i$ to those in the high-pass directional subbands at the i-th scale, and multiplying the factor $1/2^{2L}$ to those in the low-pass subband. For simplicity, the contourlet coefficients in the following will represent the regularized coefficients without explanation.

Considering one particular contourlet subband, we quantize each coefficient into n bits using deadzone quantization that step size equals 1.0 [11]. It follows that each of the quantized contourlet coefficients can be expanded into n binary bit-planes. Thus, we express it as a random variable:

$$Y = \sum_{i=0}^{n-1} 2^i Y_i, \tag{1}$$

where Y_i is a random variable representing the i-th binary bit of the coefficient. That is, each bit-plane is composed of either 0 or 1, and the joint probability distribution of the quantized coefficients is $P(Y = y) = P(Y_0 = y_0, Y_1 = y_1, \cdots, Y_{n-1} = y_{n-1})$ where $y_i \in \{0, 1\}$ is the i-th binary bit of y. If we assume Y_i's are statistically independent variables and denote the model parameter by $p_i = P(Y_i = 1)$, the joint distribution can be written as a product of Bernoulli distributions (PBD) [11]:

$$f^{PBD}(Y = y) = \prod_{i=0}^{n-1} p_i^{y_i}(1 - p_i)^{1-y_i}, \tag{2}$$

which can be characterized by the bit-plane probabilities: $P = (p_0, p_1, \cdots, p_{n-1})$.

In this way, for a given particular contourlet subband with a set of absolute quantized coefficients $\boldsymbol{y} = (y^1, y^2, \cdots, y^k)$ where $y^j \in Z^+$ is the j-th component of $\boldsymbol{y}$ and Z^+ denotes the set of all the nonnegative integer numbers, the likelihood function of $\boldsymbol{y}$ can be defined as

$$\tilde{L}(\boldsymbol{y}; P) = \log \prod_{j=1}^{k} f^{PBD}(Y = y^j; P) = \log \prod_{j=1}^{k} \prod_{i=0}^{n-1} p_i^{y_i^j}(1 - p_i)^{1-y_i^j}, \tag{3}$$

where y_i^j is the i-th binary bit of the j-th component of $\boldsymbol{y}$ and

$$P = (p_0, p_1, \cdots, p_{n-1}). \tag{4}$$

Thus, the ML estimator [11] of P can be obtained by

$$\frac{\partial \tilde{L}(\boldsymbol{y}; P)}{\partial p_i} = 0, \tag{5}$$

namely, $\hat{p}_i = \frac{1}{k}\sum_{j=1}^{k} y_i^j$, where $i = 0, 1, \cdots, n-1$. That is, the ML estimator of the model parameter is equivalent to the probabilities of one-bit occurrence for each of the bit-planes. Therefore, we can compute the bit-plane probabilities (BP) for each contourlet subband using the above ML estimators. As we all know, a sufficient statistic for a model parameter is a statistic that captures all possible information in the data about the model parameter. In the same manner as in [11], the sufficiency of the parameter estimators can also be proved by the Fisher-Neyman factorization theorem [11].

3.2 Discrepancy Measurement and *K*-Nearest Neighbor Classifier

Once the bit-plane probabilities (BP) of all subbands are obtained for every texture, we can compare the corresponding BPs of two subbands using a metric. In [11], it has been investigated and demonstrated that the $Relative - L_1$ (RL1) distance is suitable for comparing BPs. Hence, we still use RL1 as the metric of two BPs P^1 and P^2, which is given by

$$RL_1(P^1, P^2) = \sum_{i=0}^{n-1} \frac{|p_i^1 - p_i^2|}{1 + p_i^1 + p_i^2} \tag{6}$$

where $P^1 = (p_0^1, p_1^1, \cdots, p_{n-1}^1)$ and $P^2 = (p_0^2, p_1^2, \cdots, p_{n-1}^2)$. Note that the RL1 distance is a weighted L_1 one.

For two given images I_1 and I_2, we can obtain M contourlet subbands $(B_1^{I_1}, B_2^{I_1}, \cdots, B_M^{I_1})$ and $(B_1^{I_2}, B_2^{I_2}, \cdots, B_M^{I_2})$, respectively, after having implemented an L-level contourlet transform on them, and then define the distance between the two images by

$$DL(I_1, I_2) = \sum_{j=1}^{M} d_j, \tag{7}$$

where $d_j = RL_1(P_j^1, P_j^2)$ is the $Relative - L_1$ distance between the two BPs P_j^1 and P_j^2 corresponding to the subbands $B_j^{I_1}$ and $B_j^{I_2}$, respectively for $j = 1, 2, \cdots, M$.

Given a single test sample I^* and a training set, we will utilize the K-nearest-neighbor classifier to perform texture classification. In particular, we compare I^* with each training sample, and then assign it to the class to which the majority of these k nearest neighbors belong. This classifier performs better than the minimum distance classifier used in [11], which will be demonstrated in the following section.

4 Experimental Results

In this section, various experiments are carried out to demonstrate our proposed method for texture classification. In our experiments, we select the pyramid and directional filters by the "9-7" filters in the contourlet transform, which are the biorthogonal wavelet filters. In addition, we impose that each image is decomposed into one low-pass subband and four high-pass subbands at four pyramidal levels. The four high-pass subbands are then decomposed into four, four, eight, and eight directional subbands, respectively. It follows that the total number of directional subbands, M, is 25. For texture images, $n = 8$ bits are sufficient for subband coefficients. For the sake of clarity, we refer to our proposed method based on the bit-plane probability model in the contourlet domain and K-NN classifier as BPC+KNN.

4.1 Performance Evaluation

We first evaluate our method for texture classification on a typical set of 64 grey 640×640 images (shown in Fig. 2 and denoted by Set-1) from the Brodatz database [21], which was also used in [23].

In the experiments on Set-1, each image is divided into 25 128×128 nonoverlapping patches, and thus there are totally 1600 samples available. We select N_{tr} training samples from each of 64 classes and let the other samples for test with $N_{tr} = 2, 3, \cdots, 24$. The partitions are furthermore obtained randomly and the average classification accuracy rate (ACAR) is computed over the experimental results on 10 random splits of the training and test sets at each value of N_{tr}.

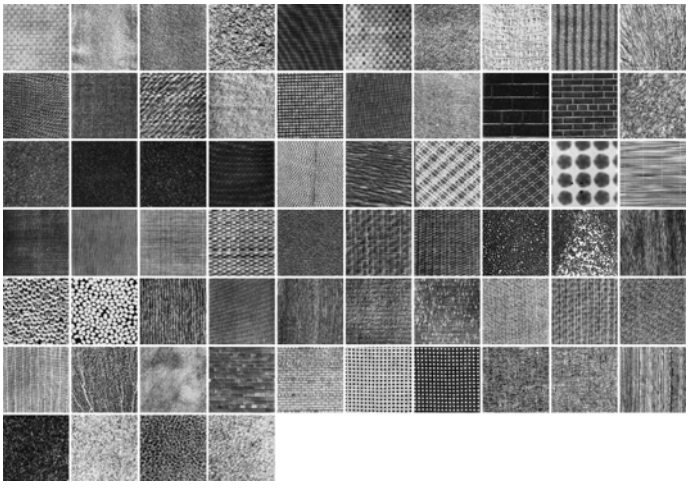


Fig. 2. The set of 64 texture images used in [23]

For Set-1, we compare our proposed BPC+KNN with two other methods. The first is the method based on the bit-plane probability model in the wavelet domain and minimum distance classifier (called BPW+MD) [11]. By this approach, the distance between a test sample and a given class is defined by the RL_1 distance between the input BP signature of the test sample and the mean of the BP signatures of all training samples in the class. Once the distances between the test sample and each class are obtained, the label of the class that has the minimum distance from the test sample is assigned to the test sample. The second method is BPC+MD, which is the same as BPC+KNN but the minimum distance (MD) classifier. For the MD classifier, the distance between a test sample and a given class is defined as the mean of the ones, defined by DL, between the test sample and all training samples in the texture class.

Fig. 3 plots the average classification accuracy rates (with error bars) of BPC+KNN, BPC+MD, and BPW+MD with respect to the number of training samples N_{tr}. As can be seen, the ACAR of BPC+KNN increases monotonically with the number of training samples. However, the ACAR of BPW+MD does not have the same regularity as that of BPC+KNN. We can also see that BPC+MD performs better than BPW+MD by about 2.0%-4.0% for each value of N_{tr}, which implies PBDs in the contourlet domain outperforms those in the wavelet domain. BPC+KNN ($K = 1$) slightly outperforms BPC+KNN (K=3) and performs better than BPC+MD by 1.9%-4.5% for each value of N_{tr}, which implies that the KNN classifier outperforms the MD classifier for the PBD model. Note that the errors are also shown in Fig. 3 where each error bar is a distance of one standard deviation above and below the average classification accuracy rate. All the values of standard deviation of BPC+KNN with $K = 1$ and $K = 3$ at each value of N_{tr} are about 0.50%, which are slightly less than the average value of standard deviation of BP Method, 1.54%. In other words, the variation of

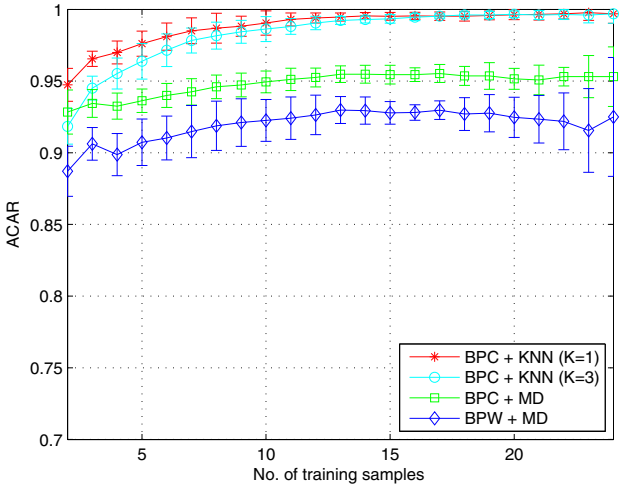


Fig. 3. The sketches of the average classification accuracy rates of our BPC+KNN and the BPW+MD with respect to the number of training samples

the classification accuracy rates of BPC+KNN for different number of training samples is small, which affirms the robustness of our proposed BPC+KNN.

We then apply BPC+KNN and BPW+MD to the Vistex dataset [22] of 40 640×640 texture images (shown in Fig. 4 and denoted by Set-2), which has also been used in [7]. Each texture image is divided into 16 128×128 non-overlapping patches. We randomly select 3 training samples from each of 40 classes, and let the other samples for test (that is, the test and training samples are separated). The ACAR is computed over the experimental results on 10 random splits of the training and test sets, which is listed in Table 1. We see that BPC+KNN ($K = 1$) outperforms BPW+MD by 8.11%.

In order to provide additional justification of our proposed method, we compare BPC+KNN ($K = 1$) with BPW+MD and M-band ridgelet transform based

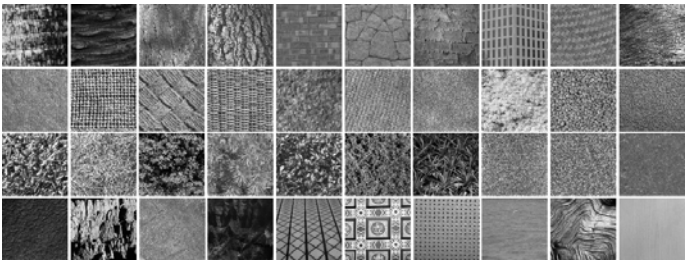


Fig. 4. The set of 40 texture images used in [7]

Table 1. The average classification accuracy rates (%) of the three methods on the two texture datasets

	BPC + KNN ($K = 1$)	BPW + MD [11]	MR Method [17]
Set-2	89.90 ± 2.01	81.79 ± 1.82	n.a.
Set-3	80.52 ± 0.89	66.28 ± 1.56	79.10

Table 2. The time for texture classification (TTC) and ACARs of the four methods on the 64 texture dataset (in seconds) in the 8-training samples case

	BPC + KNN ($K = 1$)	BPC + KNN ($K = 3$)	BPC + MD	BPW + MD [11]
TTC	206.17	206.16	207.06	692.84
ACAR	98.69%	98.17%	94.60%	91.88%

method (called MR Method) [17] on the Vistex dataset [22] of 129 512×512 texture images (denoted by Set-3), which has also been used in [17]. By the MR Method, the texture features are extracted by the M-band ridgelet transform, which is obtained by combining the M-band wavelet with the ridgelet. Then, the support vector machine classifier is used to perform supervised texture classification.

Each texture image is divided into 16 128×128 non-overlapping patches. We select 8 training samples and compute the ACAR over the experimental results on 10 random splits of the training and test sets. The average classification results of these methods are listed in Table 1. It can be seen from Table 1 that BPC+KNN ($K = 1$) outperforms MR Method and BPW+MD by 1.42% and 14.24%, respectively, on the large dataset.

4.2 Computational Cost

We further compare our proposed method with the other methods on computational cost. All the experiments conducted here have been implemented on a workstation with Intel(R) Core(TM) i5 CPU (3.2GHz) and 3G RAM in Matlab environment.

Tabel 2 reports the time for texture classification (TTC) and ACARs using the BPC + KNN ($K = 1$), BPC + KNN ($K = 3$), BPC + MD and BPW + MD approaches on the 64 texture dataset. The number of training samples used in the experiments is 8. For this dataset, the BPC + KNN ($K = 3$) method is the most efficient. In contrast, BPW + MD is the most time-consuming method among them. The TTC using BPC + KNN ($K = 1$) is 206.17 s, which is about 3 times faster than the BPW + MD. In addition, BPC + KNN ($K = 1$) is also slightly more efficient than BPC + MD. If we take into account the TTC and ACAR, the results clearly show that BPC + KNN ($K = 1$) outperforms the other methods.

5 Conclusions

We have investigated the distribution of the coefficients in each contourlet subband and tried to use the product Bernoulli distributions for modeling them. We then apply the PBD model with the use of KNN classifier to supervised texture classification. The various experiments have shown that our proposed method considerably improves the texture classification accuracy in comparison with the current state-of-the-art method based on product Bernoulli distributions in the wavelet domain as well as the method based on the M-band ridgelet transform.

Acknowledgments

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

References

1. Laine, A., Fan, J.: Texture classification by wavelet packet signatures. IEEE Transactions on Pattern Analysis and Machine Intelligence 15(11), 1186–1191 (1993)
2. Randen, T., Husoy, J.H.: Filtering for texture classification: a comparative study. IEEE Transactions on Pattern Analysis and Machine Intelligence 21(4), 291–310 (1999)
3. Unser, M.: Texture classification and segmentaion using wavelet frames. IEEE Transactions on Image Processing 4(11), 1549–1560 (1995)
4. Wouwer, G.V.D., Scheunders, P., Dyck, D.V.: Statistical texture characterization from discrete wavelet representations. IEEE Transactions on Image Processing 8(4), 592–598 (1999)
5. Kim, S.C., Kang, T.J.: Texture classification and segmentation using wavelet packet frame and Gaussian mixture model. Pattern Recognition 40(4), 1207–1221 (2007)
6. Selvan, S., Ramakrishnan, S.: SVD-based modeling for image texture classification using wavelet transformation. IEEE Transactions on Image Processing 16(11), 2688–2696 (2007)
7. Do, M.N., Vetterli, M.: Wavelet-based texture retrieval using generalized gaussian density and Kullback-Leibler distance. IEEE Transactions on Image Processing 11(2), 146–158 (2002)
8. Liu, X., Wang, D.L.: Texture classification using spectral histograms. IEEE Transactions on Image Processing 12(6), 661–670 (2003)
9. Choy, S.K., Tong, C.S.: Supervised texture classification using characteristic generalized gaussian density. Journal of Mathematical Imaging and Vision 29(1), 35–47 (2007)
10. Li, L., Tong, C.S., Choy, S.K.: Texture classification using refined histogram. IEEE Transactions on Image Processing 19(5), 1371–1378 (2010)
11. Choy, S.K., Tong, C.S.: Statistical properties of bit-plane probability model and its application in supervised texture classification. IEEE Transactions on Image Processing 17(8), 1399–1405 (2008)

12. Pi, M., Tong, C.S., Choy, S.K., Zhang, H.: A fast and effective model for wavelet subband histograms and its application in texture image retrieval. IEEE Transactions on Image Processing 15(10), 3078–3088 (2006)
13. Choy, S.K., Tong, C.S.: Statistical wavelet subband characterization based on generalized Gamma density and its application in texture retrieval. IEEE Transactions on Image Processing 19(2), 281–289 (2010)
14. Do, M.N., Vetterli, M.: The contourlet transform: An efficient directional multiresolution image representation. IEEE Transactions on Image Processing 14(12), 2091–2106 (2005)
15. Po, D.D.-Y., Do, M.N.: Directional multiscale modeling of images using the contourlet transform. IEEE Transactions on Image Processing 15(6), 1610–1620 (2006)
16. Che, G.Y., Kegl, B.: Invariant pattern recognition using contourlets and AdaBoost. Pattern Recognition 43(3), 579–583 (2010)
17. Qiao, Y.L., Song, C.Y., Zhao, C.H.: M-band ridgelet transform based texture classification. Pattern Recognition Letters 31(3), 244–249 (2010)
18. Burt, P.J., Adelson, E.H.: The Laplacian pyramid as a compact image code. IEEE Transactions on Communications COM 31(4), 532–540 (1983)
19. Bamberger, R.H., Smith, M.J.T.: A filter bank for the directional decomposition of images: theory and design. IEEE Transactions on Signal Processing 40(4), 882–893 (1992)
20. Eslami, R., Radha, H.: Translation-invariant contourlet transform and its application to image denoising. IEEE Transactions on Image Processing 15(11), 3362–3374 (2006)
21. Brodatz database, http://www.ux.uis.no/~tranden/brodatz.html
22. MIT Vision and Modeling Group. Vision Texture, http://ww.vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html
23. Lategahn, H., Gross, S., Stehle, T., Aach, T.: Texture classification by modeling joint distributions of local patterns with Gaussian mixtures. IEEE Transactions on Image Processing 19(6), 1548–1557 (2010)

Resampling Methods versus Cost Functions for Training an MLP in the Class Imbalance Context

R. Alejo[1], P. Toribio[1], J.M. Sotoca[2], R.M. Valdovinos[3], and E. Gasca[4]

[1] Tecnológico de Estudios Superiores de Jocotitlán
Carretera Toluca-Atlacomulco KM. 44.8, col. Ejido de San Juan y San Agustín, Jocotitlán
[2] Institute of New Imaging Technologies, Universitat Jaume I
Av. Sos Baynat s/n, 12071 Castelló de la Plana, Spain
[3] Centro Universitario UAEM Valle de Chalco, Universidad Autónoma del Estado de México
Hermenegildo Galena No.3, Col. Ma. Isabel, 56615 Valle de Chalco, Mexico
[4] Lab. Reconocimiento de Patrones, Instituto Tecnológico de Toluca
Av. Tecnológico s/n, 52140 Metepec, México

Abstract. The class imbalance problem has been studied from different approaches, some of the most popular are based on resizing the data set or internally basing the discrimination-based process. Both methods try to compensate the class imbalance distribution, however, it is necessary to consider the effect that each method produces in the training process of the Multilayer Perceptron (MLP). The experimental results shows the negative and positive effects that each of these approaches has on the MLP behavior.

Keywords: MLP, random sampling, cost function, class imbalance problem.

1 Introduction

Recently, the class imbalance problem has been recognized as a crucial problem in data mining and machine learning [1]. This inconvenience occurs in real-world domains, where the decision system is aimed to detect a rare but important case. For instance, in detection of oil spills, in satellite radar images, fraudulent telephone calls or fraudulent credit cards [2].

Some of the most popular strategies for handling this problem are resampling techniques (over-sampling and under-sampling) [3,2]. Over-sampling replicates samples in the minority classes and the under-sampling eliminates samples in the majority classes [4]. These two basic methods for resizing the training data set (TDS) produce a class distribution more balanced. However, both strategies have shown important drawbacks: Under-sampling may eliminate potentially useful data, while over-sampling increases the TDS size and hence the training time [5]. In the last years, research has been focused on improving these basic methods (for example see [6,7]).

In the Multilayer perceptron (MLP) which applies these methods (over or under-sampling) has demonstrated notable improvements in the classifier performance [4,1] and it has been verified that random under-sampling techniques provide better results than those obtained by means of random over-sampling [8]. However, in [1] is affirmed

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 19–26, 2011.

that the under-sampling methods produce negative effects when the TDS is seriously imbalanced.

Other popular strategy to deal with the class imbalance problem consists in internally biasing the discrimination-based process and compensate the class imbalance [9,10]. With the MLP, this approach consists mainly in including a cost function in the training phase or in the test phase. Empirical studies have shown that the usage of the cost functions improves the classifier performance. Nonetheless, the effects of the cost function in the training process is similar than over-sampling techniques, i.e., it causes changes in probability distribution data [5].

This paper studies empirically the effects of the random sampling and the cost function in the training phase of the MLP. In this way, we could identify methods that are effective in the moderate and severe imbalance problems.

2 The MLP and the Class Imbalance Problem

The MLP neural network usually comprises one input layer, one or more hidden layers, and one output layer. Input nodes correspond to features, the hidden layers are used for computing, and output layers are related with the number of classes. A neuron is the elemental unit of each layer. It computes the weighted sum of its inputs, adds a bias term and drives the result through a generally nonlinear (commonly a sigmoid) activation function to produce a single output.

The most popular training algorithm for MLP is the back-propagation strategy, which uses a training set for the learning process. Given a feedforward network, the weights are initialized with small random numbers. Each training instance is sent through the network and the output from each unit is computed. The output target is compared with the estimated one by the network, computing the error which is fed-back through the network.

To adjust the weights, the back-propagation algorithm uses a descendant gradient to minimize the squared error. Each unit in the network starts from the output unit and it is moved to the hidden units. The error value is used to adjust the weights of its connections as well as to reduce the error. This process is repeated for a fixed number of times, or until the error is minimum or it cannot be reduced.

Empirical studies of the back-propagation algorithm [11] show that class imbalance problem generates unequal contributions to the mean square error (MSE) in the training phase. Clearly the major contribution to the MSE is produced by the majority class.

We can consider a TDS with two classes ($m = 2$) such that $N = \sum_i^m n_i$ and n_i is the number of samples from class i. Suppose that the MSE by class can be expressed as $E_i(U) = \frac{1}{N} \sum_{n=1}^{n_i} \sum_{p=1}^{L} (d_p^n - y_p^n)^2$, where d_p^n is the desired output and y_p^n is the actual output of the network for sample n. Then the overall MSE can be expressed as $E(U) = \sum_{i=1}^{m} E_i = E_1(U) + E_2(U)$.

If $n_1 << n_2$ then $E_1(U) << E_2(U)$ and $\|\nabla E_1(U)\| << \|\nabla E_2(U)\|$, consequently $\nabla E(U) \approx \nabla E_2(U)$. So, $-\nabla E(U)$ it is not always the best direction to minimize the MSE in both classes.

Regarding that the imbalance problem affects negatively in the back-propagation algorithm due to the disproportionate contributions in the MSE, it is possible to consider

two options: A) Resizing the training data set in order to cause that the class distribution of the TDS become more balanced (for instance, replicating samples in the minority classes or eliminating samples in the majority classes [4]). B) Including a cost function in the back-propagation algorithm for avoiding that the minority classes are ignored in the learning process, and it could accelerate the convergence of the neural network.

Consider a cost function (γ) that balance the TDS class imbalance as follows: $E(U) = \sum_{i=1}^{m} \gamma(i)\ E_i = \gamma(1)E_1(U) + \gamma(2)E_2(U) = \frac{1}{N}\sum_{i=1}^{m}\gamma(i)\sum_{n=1}^{n_i}\sum_{p=1}^{L}(d_p^n - y_p^n)^2$, where $\gamma(1)\|\nabla E_1(U)\| \approx \gamma(2)\|\nabla E_2(U)\|$ avoiding that the minority class be ignored in the learning process. In this work, the cost function is defined as $\gamma(i) = \|\nabla E_{max}(U)\|/\|\nabla E_i(U)\|$, where $\|\nabla E_{max}(U)\|$ corresponds to the largest majority class.

3 Methodology

The experiments were carried out on eleven severely imbalanced datasets. These datasets were obtained from the transformation of Cayo into two-class problems (reducing a m-class problem to a set of m two-class sub-problems). The main characteristics of these subsets have been summarized in the Table 1. To increase statistical significance of the results the k-fold cross validation technique (with k=10) has been applied. About 90% out of the total number of samples available has been used for the TDS and the rest for a test set.

The MLP used in this study was trained by the back-propagation algorithm in batch mode. The learning rate (η) was set 0.1. One hidden layer was used with four neurons. The stop criterion was established at 25000 iterations or an MSE below 0.001. The training MLP has been repeated ten times. The results here included correspond to the average of those achieved in the ten repetitions and of ten partitions.

A general criterion to measure the classifier performance is the overall accuracy (Acc). $Acc = 1 - n_e/n$ where n_e is the number of misclassified examples and n is the

Table 1. Main characteristics of the eleven subsets obtained from Cayo; notice that the distribution of data is presented in different forms to simplify their interpretation

Data	Samples	Features	Distribution	Ratio by class	Ratio
C01	6019	4	838/5181	0.14/0.86	0.16
C02	6019	4	293/5726	0.05/0.95	0.05
C03	6019	4	624/5395	0.10/0.90	0.12
C04	6019	4	322/5697	0.05/0.95	0.06
C05	6019	4	133/5886	0.02/0.98	0.02
C06	6019	4	369/5650	0.06/0.94	0.07
C07	6019	4	324/5695	0.05/0.95	0.06
C08	6019	4	722/5297	0.12/0.88	0.14
C09	6019	4	789/5230	0.13/0.87	0.15
C10	6019	4	833/5186	0.14/0.86	0.16
C11	6019	4	722/5247	0.13/0.87	0.15

total number of testing examples. Nevertheless, in the class imbalance problems this is not the most suitable measure [6]. The geometric mean (*g-mean*) is one of the most widely accepted criterion, and is defined as $g = \sqrt{a^+ \cdot a^-}$, where $a^+ = 1 - n_e^{cls^+}/n^{cls^+}$ is the accuracy on the minority class (cls^+) and $a^- = 1 - n_e^{cls^-}/n^{cls^-}$ is the accuracy on the majority class (cls^-). In this work, *g-mean*, Acc, a^- and a^+ were applied to measure classifier performance.

4 Experimental Results

The random over-sampling strategies in a severe-imbalance context cannot be considered suitable alternatives given the considerable increase in the computing cost, they would generate in the neural network a slow training process. Therefore, this paper is focused on analyze the random under-sampling and cost function strategies, (see section 2) and its convenience to be used on a context of a severe class imbalance problem.

Japkowicz [4,1] observe that the random under-sampling method can improve considerably the classifier performance by compensating the class imbalance and by reducing the computational cost associated to the model. But, which is its performance when the TDS has severe multi-class imbalance problem?.

On the other hand, empirical studies have shown that using a cost function can improve the classifier performance [9]. Functionally, using a cost function is equivalent to apply random over-sampling, but it does not increase (significantly) the computational cost in the training process. However, as in the case of the random under-sampling technique is necessary to ask, what are the effects of the cost function when the TDS is severely imbalanced?.

In Table 2 the obtained results with the cost function and random under-sampling are shown. It is possible to observe that the performance of under-sampling is better than the obtained with the original dataset, but worse than the produced by the cost function. Moreover, the under-sampling performance is comparable to the cost function (except in C04, C05 and C06).

For more detail, in the Fig. 1 the accuracy by class is shown (a^+ and a^-). The Fill boxes symbolize the cost function and the not fill ones the random under-sampling strategy. Observe that in the corresponding image, the cls^+ (Fig. 1a), in most of the datasets, a^+, is almost the same. In the sets related to C05 and C06, random under-sampling presents a a^+ inferior than the obtained with the cost function. Nonetheless, these results are higher than those shown by the MLP with the original dataset (a^+=47.55% for C05 and a^+=0.0% for C06). In other words, the random under-sampling increases the a^+ in relation to the standard back-propagation algorithm, but it does not shows better results than those from the cost function.

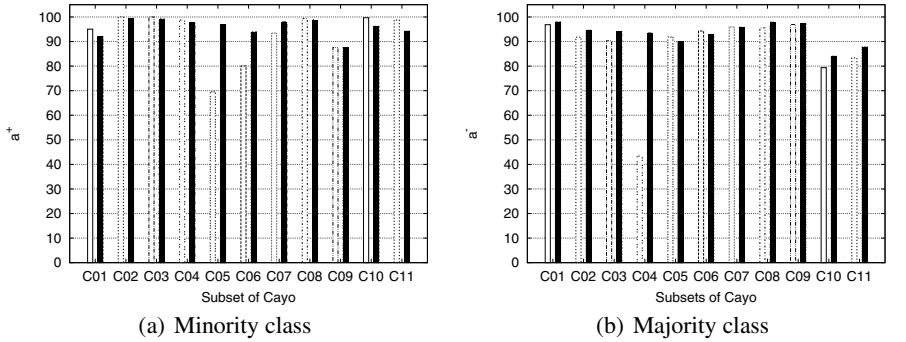
See Fig.1b (majority class) that in most of the subsets a^- both with cost function and random under-sampling is similar. Also, a tendency towards presenting better results is observed with cost function. In the particular case of the C04, the a^- obtained with under-sampling (a^-=43.87%) is very low regarding with original dataset (a^-=100%) and with the cost function (a^-=93.46%).

In summary, the results reported in the Table 2 suggest that on the severe class imbalance, random under-sampling can be a good choice. However, in some results were

Table 2. MLP: Average accuracy rate

Acc	Original	Cost function	Under-sampling
C01	97.81(0.03)	97.13(0.63)	96.61(0.74)
C02	97.42(0.36)	94.90(0.02)	92.18(0.15)
C03	96.40(0.04)	94.54(0.20)	91.30(0.42)
C04	94.65(0.00)	93.69(2.93)	46.36(0.92)
C05	97.81(0.00)	90.12(8.30)	91.34(6.17)
C06	96.79(0.04)	92.87(0.10)	93.41(0.36)
C07	94.58(0.00)	95.90(0.24)	95.80(3.43)
C08	98.54(1.01)	98.04(1.17)	96.03(2.75)
C09	98.12(0.06)	96.06(0.34)	95.71(0.71)
C10	92.33(0.72)	85.69(0.39)	82.17(0.06)
C11	89.32(0.97)	88.68(1.12)	85.31(0.46)
g-mean	Original	Cost function	Under-sampling
C01	93.60(0.03)	94.95(1.30)	95.95(1.47)
C02	68.25(6.34)	97.01(0.01)	95.85(0.08)
C03	94.32(0.18)	96.62(0.09)	95.07(0.25)
C04	0.00(0.00)	95.58(1.39)	65.50(0.72)
C05	0.00(0.00)	93.31(4.40)	79.95(4.80)
C06	68.99(0.47)	93.43(0.22)	86.91(0.93)
C07	0.00(0.00)	96.93(0.19)	94.71(2.37)
C08	96.67(3.66)	98.44(1.03)	97.49(1.66)
C09	93.37(0.03)	92.37(0.18)	92.17(0.39)
C10	75.14(3.99)	89.94(0.25)	88.99(0.05)
C11	53.34(13.22)	91.04(1.01)	90.76(0.21)

observed that the sampling methods produce a negative effect when the TDS is seriously imbalanced (confirm previous hypotheses [1]). So, the question here is, why?, or where reflected this effect?. To answer these questions, the outputs MLP of the C04 and C03 subsets are analyzed.

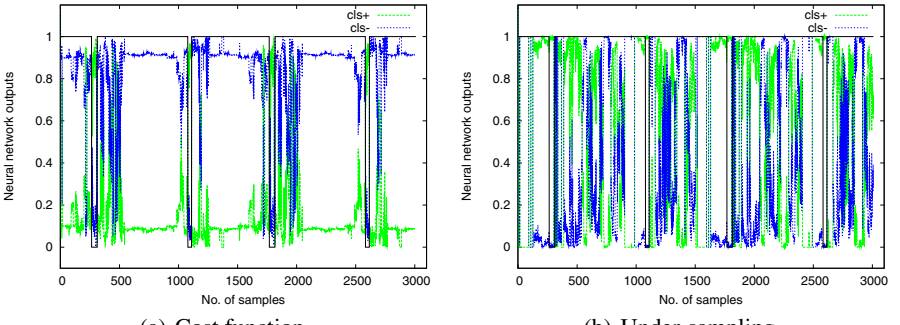


(a) Minority class (b) Majority class

Fig. 1. Accuracy of the minority class (a) and the majority class (b). Numbers in x axis represent the dataset used. The Fill boxes symbolize the cost function and the not fill boxes the random under-sampling strategy.

The Fig. 2 contains the graphic representation of MLP outputs for each sample utilized by evaluate the network performance with C04. Let us notice that x axis indicates the samples contained in the evaluation set, while y axis reflects the value of the network output (values between 0 and 1). The continuous line (black) establishes the separation between cls^+ and cls^- samples. Observe that in the largest intervals limited by this line, contain only samples of the most represented class and, in the smallest the elements of the cls^+. Green is associated to the neuron outputs corresponding to cls^+ and blue to the neuron associated to cls^-. For instance, in ideal conditions, i.e., when the network classify correctly, the behavior should be: when a sample is cls^+ the output value of the neuron associated to the cls^+ must be high (high values in green) and low for the neuron of cls^- (low values in blue) and viceversa when it is the case of cls^-.

Observe in Fig. 2a (cost function), the outputs for samples of the cls^- (blue) are high (values close to 1) when it corresponds of its class and low in other way. This behavior is constant in the Fig. 2a, which the cost function has a good performance in both cases, cls^+ and cls^-. An utterly different situation is shown in Fig. 2b (under-sampling) where the previous behavior is not observed. The Fig. 2b show a good performance on the cls^+, and very irregular on the cls^-.



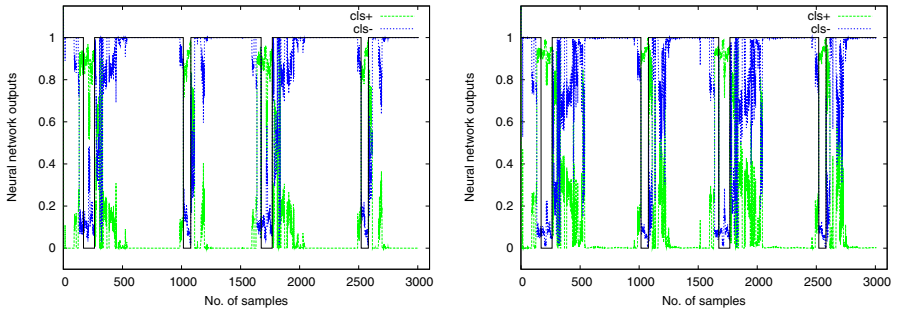
(a) Cost function (b) Under-sampling

Fig. 2. MLP outputs for the C04 subset, after to apply cost function and random under-sampling strategies. The line in black shows the separation between the outputs of both classes.

The results obtained with this subset suggest a weak learning on the cls^- (when random under-sampling is applied). Nevertheless, in the rest of the subsets it seemed as though this massive elimination of samples does not affect the cls^- (as it was observed in Fig. 1), but, what happen with this?.

The answer is in the sense that the class most represented (cls^-) is less learned when training samples are massively eliminated, i.e., the outputs present more irregular tendency than the cost function. To exemplify this, the subset C03 was utilized, i.e., this subset as the rest of the subsets does not present a significant difference related to random under-sampling and the cost function in their accuracy values, both for the cls^- and $cls+$ (Fig. 1).

In Fig. 3, the MLP outputs for the C03 subset are presented after to applying a cost function and random under-sampling technique. In Fig. 3a (cost function) a steadier



(a) Cost Function (b) Under-sampling

Fig. 3. MLP outputs for the C03 subset, after apply the cost function and random under-sampling strategies. The line in black shows the separation between the outputs of both classes.

tendency is observed in the MLP outputs, while in the Fig. 3b shows a more irregular behavior. However, the negative effects on the cls^- are not very severe than the case of C04 subset.

These results confirm that other problems, such as the overlap between classes or noise in the TDS, should be taken into account in classification tasks when the TDS is imbalanced [12, 13].

5 Conclusion

In this paper the suitable random under-sampling and cost functions for handling the severe class imbalance was empirically studied. The results suggest that when the imbalance is severe, the random under-sampling presents a tendency to have a weak learning on the cls^-. This situation becomes graver when there are very few elements in the cls^+, then a tendency toward over fitting (over fitting cls^+) appears. However, the main cause of this situation is the existence of overlap or noise in the TDS. Also, was observed that others subsets with same imbalance level their performance was high.

Finally, under a context of severe class imbalance the best alternative is to use a cost function for compensate the class imbalance. The application of cost functions improves two fundamental aspects: they prevent a weak learning on the cls^- (because avoids losing potentially useful data) and, they increase the cls^+ participation in the MLP learning.

Acknowledgment

This work has been partially supported by the Spanish Ministry of Science and Education under project CSD2007-00018, UAEMCA-114 and SBI112 from the Mexican SEP, the 2703/2008U from the UAEM project and SDMAIA-010 from the TESJO project.

References

1. Zhou, Z.-H., Liu, X.-Y.: Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Transactions on Knowledge and Data Engineering 18, 63–77 (2006)
2. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. 21(9), 1263–1284 (2009)
3. Visa, S.: Issues in mining imbalanced data sets - a review paper. In: Artificial Intelligence and Cognitive Science Conference, pp. 67–73 (2005)
4. Japkowicz, N., Stephen, S.: The class imbalance problem: a systematic study. Intelligent Data Analysis 6, 429–449 (2002)
5. Lawrence, S., Burns, I., Back, A., Tsoi, A.C., Giles, C.L.: Neural network classification and prior class probabilities. In: Orr, G., Müller, K.-R., Caruana, R. (eds.) NIPS-WS 1996. LNCS, vol. 1524, pp. 299–314. Springer, Heidelberg (1998)
6. Kubat, M., Matwin, S.: Detection of oil-spills in radar images of sea surface. Machine Learning (30), 195–215 (1998)
7. Chawla, N.V., Bowyer, K.W., Hall, L.O., Philip Kegelmeyer, W.: Smote: Synthetic minority over-sampling technique. J. Artif. Intell. Res. (JAIR) 16, 321–357 (2002)
8. Japkowicz, N., Myers, C., Gluck, M.: A novelty detection approach to classification. In: Proceedings of the Fourteenth Joint Conference on Artificial Intelligence, pp. 518–523 (1995)
9. Anand, R., Mehrotra, K., Mohan, C.K., Ranka, S.: Efficient classification for multiclass problems using modular neural networks. IEEE Transactions on Neural Networks 6(1), 117–124 (1995)
10. Bruzzone, L., Serpico, S.B.: Classification of imbalanced remote-sensing data by neural networks. Pattern Recognition Letters 18, 1323–1328 (1997)
11. Anand, R., Mehrotra, K.G., Mohan, C.K., Ranka, S.: An improved algorithm for neural network classification of imbalanced training sets. IEEE Transactions on Neural Networks 4, 962–969 (1993)
12. Visa, S., Ralescu, A.: Learning imbalanced and overlapping classes using fuzzy sets. In: Workshop on Learning from Imbalanced Datasets(ICML 2003), pp. 91–104 (2003)
13. Prati, R.C., Batista, G.E.A.P.A., Monard, M.C.: Class imbalances *versus* class overlapping: An analysis of a learning system behavior. In: Monroy, R., Arroyo-Figueroa, G., Sucar, L.E., Sossa, H. (eds.) MICAI 2004. LNCS (LNAI), vol. 2972, pp. 312–321. Springer, Heidelberg (2004)

Prediction of Urban Stormwater Runoff in Chesapeake Bay Using Neural Networks

Nian Zhang

University of the District of Columbia, Department of Electrical and Computer Engineering, 4200 Connecticut Avenue, NW, Washington D.C. 20008 USA
nzhang@udc.edu

Abstract. Runoff carries pollutants such as oil, heavy metals, bacteria, sediment, pesticides and fertilizers into streams or groundwater. The combined impacts of hydrologic changes and water pollution can be disastrous for streams and rivers in urban areas and the Chesapeake Bay. Therefore, evaluations of stormwater runoff are imperative to enhance the performance of an assessment operation and develop better water resources management and plan. In order to accomplish the goal, a recurrent neural network based predictive model trained by the Levenberg-Marquardt backpropagation training algorithm is developed to forecast the runoff discharge using the gage height and the previous runoff discharge. The experimental results showed that Levenberg-Marquardt backpropagation training algorithm proved to be successful in training the recurrent neural network for the stormwater runoff prediction. Based on the comparison studies about the impact of discharge and gage height on the runoff forecast accuracy, it was found that when both the previous discharge and gage height were used, the network achieved lower mean squared error, and better time series response than the case when the gage height is the only input or target.

Keywords: Urban Runoff Prediction, Recurrent Neural Networks, Levenberg-Marquardt Backpropagation Training Algorithm, Chesapeake Bay.

1 Introduction

Stormwater from urban and suburban areas contributes a significant amount of pollutants to the Chesapeake Bay. Any gage height in an urban or suburban area that does not evaporate or soak into the ground, but instead pools and travels downhill, is considered stormwater. Stormwater is also referred to as urban stormwater, runoff and polluted runoff. Increased development across the Chesapeake Bay watershed has made stormwater runoff the fastest growing source of pollution to the Chesapeake Bay and its rivers [1][2]. The Chesapeake Bay is the largest estuary in the United States. It lies off the Atlantic Ocean, surrounded by Maryland and Virginia. The Chesapeake Bay's drainage basin covers 64,299 square miles in the District of Columbia and parts of six states: New York, Pennsylvania, Delaware, Maryland, Virginia, and West Virginia [3]. More than 150 rivers and streams drain into the Bay.

Many research studies have been performed to forecast the runoff. They benefit substantially from the progress of computational intelligence techniques [4]. The techniques

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 27–36, 2011.

include neural networks [5][6], fuzzy logic [7], evolutionary algorithm [8], support vector machine [9], particle swarm optimization [10], or the combination of them [11][12]. Comparatively, various runoff forecast models based on neural networks perform much better in accuracy than many conventional prediction models.

However, a fact could not be neglected that most of the existing computational intelligence based models have not yet satisfied researchers in forecast precision. In addition, none of the above computational intelligence methods have been used for the urban runoff prediction in the District of Columbia and the suburbs, although a few runoff quality analysis tools of urban catchments with probabilistic models were developed [13]. To fill this gap, it is very important to investigate state-of-the-art computational intelligence with the potential for higher rates for urban runoff forecast.

This paper is organized as follows. In Section 2, the data for the study area is introduced, and then the design methods including the neural network architecture and the learning algorithm are presented. In Section 3, experimental results are demonstrated. The comparison between the cases when the discharge and gage height are inputs, and when the gage height is used as the only input is conducted to investigate the impact of discharge and gage height on the runoff prediction accuracy. In Section 4, the conclusions are given.

2 Design Method and Algorithm

2.1 Data

The study area will focus on the Four Mile Run at Alexandria, VA. The Four Mile Run is 9.2 miles long, and is a direct tributary of the Potomac River, which ultimately carries the water flowing from Four Mile Run to the Chesapeake Bay. The stream passes from the Piedmont through the fall line to the Atlantic Coastal Plain, and eventually empties out into the Potomac River. Potomac River was determined to be one of the most polluted water bodies in the nation mainly due to the CSOs and stormwater discharges and wastewater treatment plant discharges. In addition, because of the highly urbanized nature of the Four Mile Run watershed, the neighborhoods and businesses adjacent to this portion of the run were subjected to repeated flooding, beginning in the 1940s. Therefore, the flood-control solutions are the major concern. Runoff prediction would provide a promising solution for flood-control.

The real-time USGS data for the Four Mile Run station include both the discharge data and gage height data, which is useful for investigating their impact to the long-run discharge forecast. The runoff data was retrieved for 120 days between August 28, 2010 and December 4, 2010. The runoff discharge (cubic feet per second) data is plotted in Fig. 1, and the gage height (feet) data is illustrated in Fig. 2. 70% of the data is used for training. They are presented to the network during training, and the network is adjusted according to its error. 15% of the data is used for validation. They are used to measure network generalization, and to halt training when generalization stops improving. The last 15% of the data is used for testing. They provide an independent measure of network performance during and after training.

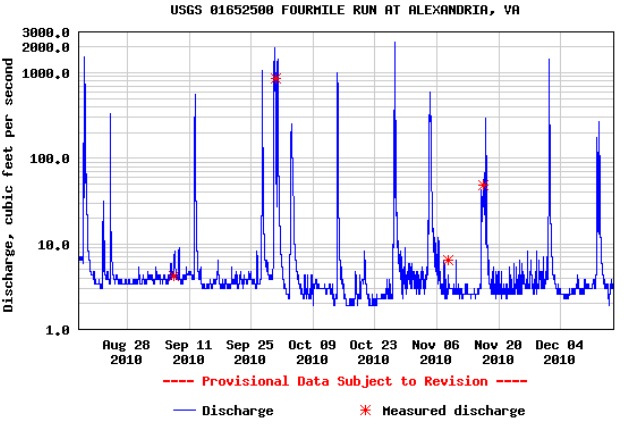


Fig. 1. The runoff discharge data (cubic feet per second) collected at the Four Mile Run site at Alexandria, VA during August 28, 2010 to December 4, 2010

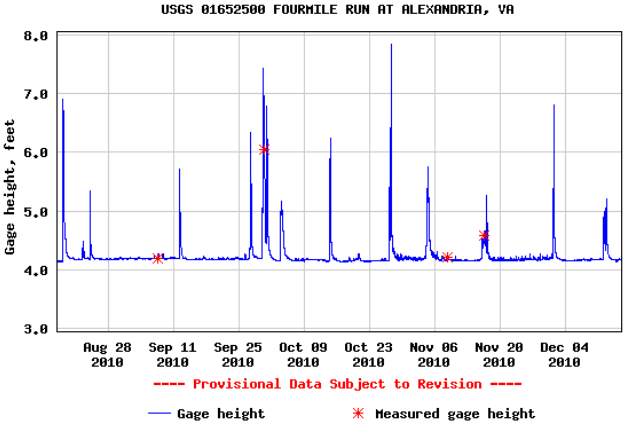


Fig. 2. Gage height (feet) collected at the Four Mile Run site at Alexandria, VA during August 28, 2010 to December 4, 2010

2.2 Neural Network Architecture

Two neural network based predictive models are to be developed to predict future values of runoff discharge, based on the previous runoff discharge and/or gage height. The first model can be represented mathematically by predicting future values of the discharges time series y(t) from past values of that time series and past values of the precipitation time series x(t), as shown in Fig. 3.

This form of prediction can be written as follows:

$$y(t) = f(y(t-1),\ldots,y(t-d),x(t-1),\ldots,x(t-d)) \quad (1)$$

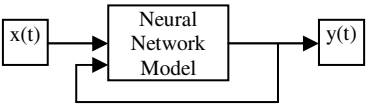


Fig. 3. The first neural networks based prediction model. The future values of the discharges y(t) can be predicted from past values of y(t) and past values of the gage height time series x(t).

In the second predictive model, the future values of the discharges time series y(t) could be predicted from past values of that time series, as shown in Fig. 4.

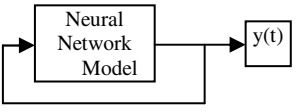


Fig. 4. The second neural networks based prediction model. The future values of the discharges y(t) can be predicted from past values of y(t).

The corresponding form of prediction can be written as follows:

$$y(t) = f(y(t-1), \ldots, y(t-d)) \tag{2}$$

The above neural network models are two-layer feedforward networks, with a sigmoid transfer function in the hidden layer and a linear transfer function in the output layer. W is the weight matrix, and b is the bias. This network also uses tapped delay lines to store previous values of the x(t) and y(t) sequences. There are 50 neurons in the hidden layer, and 1 neuron in the output layer. 11 delay lines are used. The output of the network, y(t), is fed back to the input of the network through delays, since y(t) is a function of y(t-1), y(t-2), ..., y(t-d). However, the network will be created and trained in this open loop form.

2.3 Network Learning Algorithm

While backpropagation with gradient descent technique is a steepest descent algorithm, the Levenberg-Marquardt algorithm is an approximation to Newton's method [14]. If a function $V(x)$ is to be minimized with respect to the parameter vector x, then Newton's method would be [15]:

$$\Delta x = -[\nabla^2 V(x)]^{-1} \nabla V(x) \tag{3}$$

where $\nabla^2 V(x)$ is the Hessian matrix and $\nabla V(x)$ is the gradient. If V(x) is expressed as:

$$V(x) = \sum_{i=1}^{N} e_i^2(x) \tag{4}$$

Then it can be shown that:

$$\nabla V(x) = J^T(x)e(x) \tag{5}$$

$$\nabla^2 V(x) = J^T(x)J(x) + S(x) \tag{6}$$

where J(x) is the Jacobian matrix and

$$S(x) = \sum_{i=1}^{N} e_i \nabla^2 e_i(x) \tag{7}$$

For the Gauss-Newton method it is assumed that $S(x) \approx 0$, and the equation (3) becomes:

$$\Delta x = [J^T(x)J(x)]^{-1} J^T(x)e(x) \tag{8}$$

The Levenberg-Marquardt modification to the Gauss-Newton method is:

$$\Delta x = [J^T(x)J(x) + \mu I]^{-1} J^T(x)e(x) \tag{9}$$

The parameter μ is multiplied by some factor (β) whenever a step would result in an increased $V(x)$. When a step reduces $V(x)$, μ is divided by β. When the scalar μ is very large the Levenberg-Marquardt algorithm approximates the steepest descent method. However, when μ is small, it is the same as the Gauss-Newton method. Since the Gauss-Newton method converges faster and more accurately towards an error minimum, the goal is to shift towards the Gauss-Newton method as quickly as possible. The value of μ is decreased after each step unless the change in error is positive; i.e. the error increases.

3 Experimental Results

3.1 Number of Hidden Neurons and Delays

We continuously increase both the number of neurons in the hidden layer and the number of delays in the tapped delay lines until the network performed well in terms of the mean square error (MSE) and the error autocorrelation function. After several trials, the best number of hidden neurons is determined to be 50, and the best number of delays in the tapped delay lines is 11.

3.2 Mean Squared Error

Because the true output is available during the training of the network, we can use the open-loop architecture, in which the true output is used instead of feeding back the estimated output. This has two advantages. The first is that the input to the feedforward network is more accurate. The second is that the resulting network has a purely feedforward architecture, and therefore a more efficient algorithm can be used for training. In this case, Levenberg-Marquardt backpropagation training algorithm was used.

The mean squared error is the averaged squared difference between outputs and targets. Training automatically stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples. The best validation performance is 0.00025259 at epoch 19 when the input is gag height and the target is discharge, as shown in Fig. 5. For discharge and gauge height, the MSE are 8.98160e-5, 2.52594e-4, and 3.71716e-4 for the training data, validation data, and testing data, respectively.

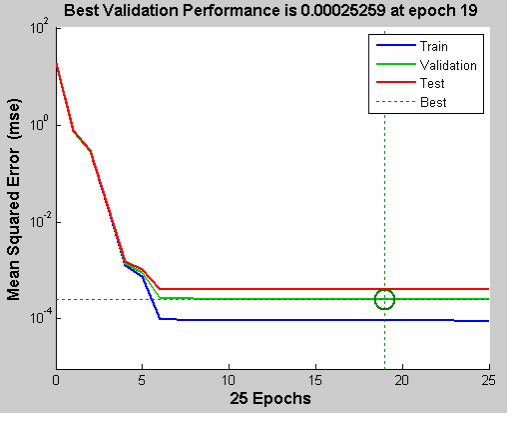


Fig. 5. The best validation performance is 0.00025259 at epoch 19. The input is gag height, and the target is discharge.

Comparatively, for gage height as the only input (i.e. used as target), the best validation performance is 0.00025259 at epoch 1000, as shown in Fig. 6. The MSE are 1.48464e-5, 2.07226e-2, and 2.08129e-3 for the training data, validation data, and testing data, respectively.

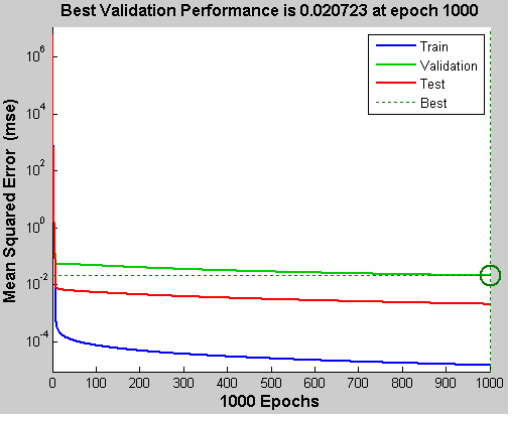


Fig. 6. The best validation performance is 0.020723 at epoch 1000. The only input (i.e. target) is gage height.

3.3 Error Autocorrelation Function

The error autocorrelation function is used to validate the network performance. The error autocorrelation function is demonstrated in Fig. 7. It describes how the prediction errors are related in time. For a perfect prediction model, there should only be one nonzero value of the autocorrelation function, and it should occur at zero lag, i.e. this is the mean square error. This would mean that the prediction errors were completely uncorrelated with each other (white noise). If there was significant correlation in the prediction errors, then it is possible to improve the prediction by increasing the number of delays in the tapped delay lines [16][17]. In Fig. 7, the correlations, except for the one at zero lag, fall approximately within the 95% confidence limits around zero, so the model seems to be adequate. In comparison, if gage height was used as the only input/target, the error autocorrelation function is plotted in Fig. 8.

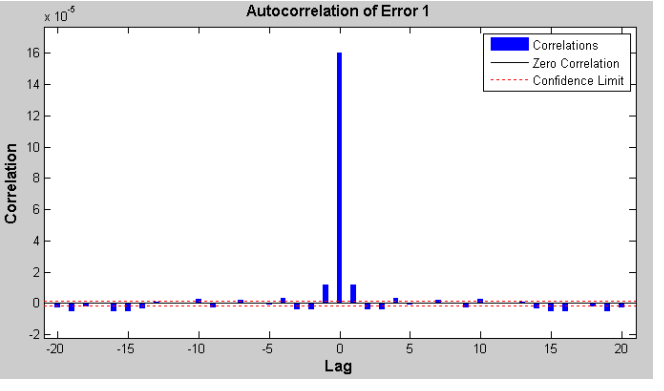


Fig. 7. Error autocorrelation function when the input is gage height, and the target is discharge. It describes how the prediction errors are related in time.

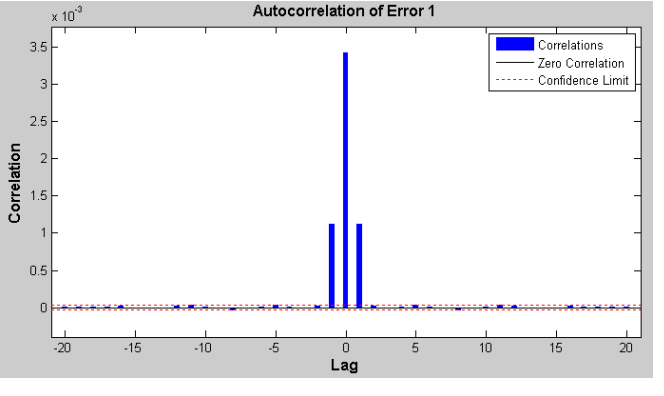


Fig. 8. Error autocorrelation function when the only input is gage height. It describes how the prediction errors are related in time.

3.4 Time Series Response

A comparative study was performed between the case when the discharge and gage height are inputs, and when the gage height is used as the only input. The time series response when both the discharge and gage height are inputs is demonstrated in Fig. 9. The time series response when the gage height is the only input/target is shown in Fig. 10. The top plot displays the outputs and targets versus time. For each selected

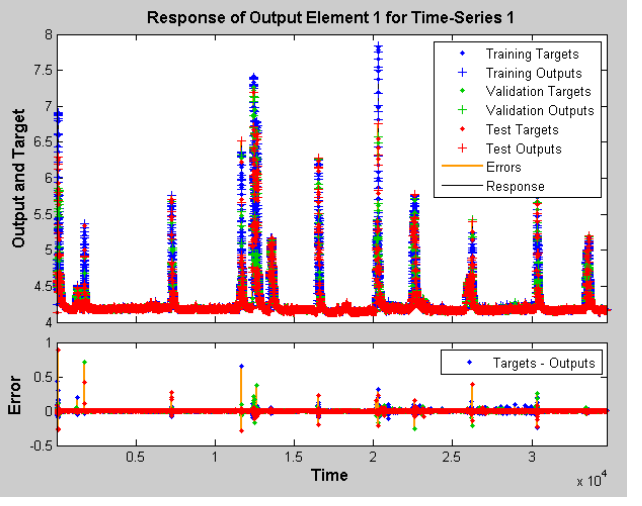


Fig. 9. The time series response when both the discharge and gage height are inputs

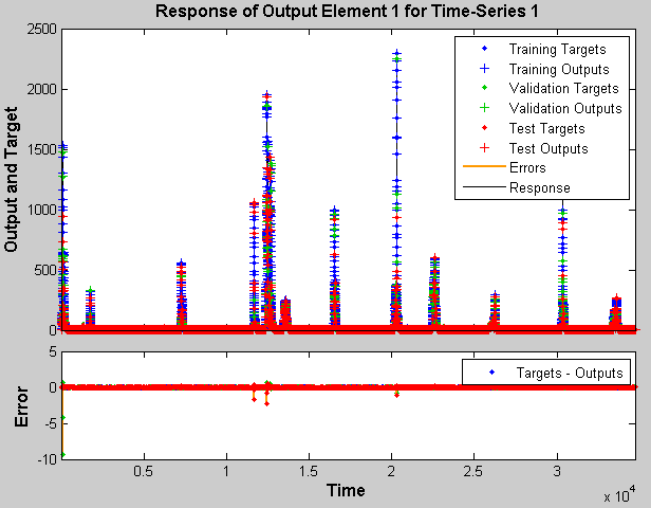


Fig. 10. The time series response when the gage height is the only input/target

time point for training, testing and validation, all the training targets, training outputs, validation targets, validation outputs, test targets, and test outputs are plotted. The bottom plot shows the error versus time. At those selected time point for training, testing and validation, the errors for training target, validation target, and test target are plotted. The solid line is used to measure the magnitude of errors.

4 Conclusions

This paper proposed a recurrent neural network based predictive model trained by the Levenberg-Marquardt backpropagation algorithm to forecast the stormwater runoff using the gage height and the previous stormwater runoff.

A two-layer feedforward network, with a sigmoid transfer function in the hidden layer and a linear transfer function in the output layer was developed to investigate the influence of the past runoff discharge and gage height on runoff discharge prediction accuracy. The experimental results show that Levenberg-Marquardt back-propagation training algorithm proved to be successful in training the recurrent neural network for the stormwater runoff prediction. Based on the comparison studies about the impact of discharge and gage height on the runoff forecast accuracy, it was found that when both the previous discharge and gage height were used, the network achieved lower mean squared error, and better time series response than the case when the gage height is the only input or target.

Acknowledgment

The author would like to express thanks to the National Science Foundation (Award #: OISE-1066140).

References

1. U.S. Environmental Protection Agency (EPA). Washington, DC. Protecting Water Quality from Urban Runoff. Document No. EPA 841-F-03-003 (2003)
2. United States. National Research Council. Washington, DC. Urban Stormwater Management in the United States, pp. 18–20 (2008)
3. Fact Sheet 102-98 - The Chesapeake Bay: Geologic Product of Rising Sea Level. U. S. Geological Survey (1998), http://pubs.usgs.gov/fs/fs102-98/
4. Haykins, S.: Neural Networks: A Comprehensive Foundation, 2nd edn. Prentice Hall, Englewood Cliffs (1998)
5. Solaimani, K.: Rainfall-runoff Prediction Based on Artificial Neural Network (A Case Study: Jarahi Watershed). American-Eurasian J. Agric. & Environ. Sci. 5(6), 856–865 (2009)
6. Guo, W., Wang, H., Xu, J., Zhang, Y.: RBF Neural Network Model Based on Improved PSO for Predicting River Runoff. In: 2010 International Conference on Intelligent Computation Technology and Automation (ICICTA), pp. 968–971 (2010)
7. Wang, W., Qiu, L.: Prediction of Annual Runoff Using Adaptive Network Based Fuzzy Inference System. In: 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), vol. 3, pp. 1324–1327 (2010)

8. Bai, P., Song, X., Wang, J., Shi, W., Wang, Q.: A Hillslope Infiltration and Runoff Prediction Model of Neural Networks Optimized by Genetic Algorithm. In: 2010 International Conference on Mechanic Automation and Control Engineering (MACE), pp. 1256–1259 (2010)
9. Liu, J., Wang, B.: Parameters Selection for SVR Based on the SCEM-UA Algorithm and its Application on Monthly Runoff Prediction. In: 2007 International Conference on Computational Intelligence and Security, pp. 48–51 (2007)
10. Guo, W., Wang, H., Xu, J., Zhang, Y.: RBF Neural Network Model Based on Improved PSO for Predicting River Runoff. In: 2010 International Conference on Intelligent Computation Technology and Automation (ICICTA), vol. 2, pp. 968–971 (2010)
11. Bo, H., Dong, X., Deng, X.: Application of GA-ANN Hybrid Algorithms in Runoff Prediction. In: 2010 International Conference on Electrical and Control Engineering (ICECE), pp. 5039–5042 (2010)
12. Liu, Y., Chen, Y., Hu, J., Huang, Q., Wang, Y.: Long-term Prediction for Autumn Flood Season in Danjiangkou Reservoir Basin Based on OSR-BP Neural Network. In: 2010 Sixth International Conference on Natural Computation (ICNC), vol. 4, pp. 1717–1720 (2010)
13. Behera, P., Li, J., Adams, B.: Runoff Quality Analysis of Urban Catchments with Analytical Probabilistic Models. Journal of Water Resources Planning and Management, ASCE (2006)
14. Marquardt, D.: An Algorithm for Least Squares Estimation of Non-Linear Parameters. Journal of the Society for Industrial and Applied Mathematics, 431–441 (1963)
15. Kisi, O.: Multi-Layer Perceptrons with Levenberg-Marquardt Training Algorithm for Suspended Sediment Concentration Prediction and Estimation. Hydrological Sciences - Journal – des Sciences Hydrologiques 49(6), 1025–1040 (2004)
16. Cai, X., Zhang, N., Venayagamoorthy, G.K., Wunsch II, D.C.: Time Series Prediction with Recurrent Neural Networks Trained by a Hybrid PSO-EA Algorithm. Neurocomputing 70(13-15), 2342–2353 (2007)
17. Cai, X., Zhang, N., Venayagamoorthy, G.K., Wunsch II, D.C.: Time Series Prediction with Recurrent Neural Networks Using Hybrid PSO-EA Algorithm. In: INNS-IEEE International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, July 25-28, vol. 2, pp. 1647–1652 (2004)

One-Shot Learning of Poisson Distributions in Serial Analysis of Gene Expression

Peter Tiňo

School of Computer Science
The University of Birmingham, Birmingham, B15 2TT, UK
P.Tino@cs.bham.ac.uk
http://www.cs.bham.ac.uk/~pxt

Abstract. Traditionally, studies in learning theory tend to concentrate on situations where potentially ever increasing number of training examples is available. However, there are situations where only extremely small samples can be used in order to perform an inference. In such situations it is of utmost importance to theoretically analyze what and under what circumstances can be learned. One such scenario is detection of differentially expressed genes. In our previous study (BMC Bioinformatics, 2009) we theoretically analyzed one of the most popular techniques for identifying genes with statistically different expression in SAGE libraries - the Audic-Claverie statistic (Genome Research, 1997). When comparing two libraries in the Audic-Claverie framework, it is assumed that under the null hypothesis their tag counts come from the same underlying (unknown) Poisson distribution. Since each SAGE library represents a single measurement, the inference has to be performed on the smallest sample possible - sample of size 1. In this contribution we compare the Audic-Claverie approach with a (regularized) maximum likelihood (ML) framework. We analytically approximate the expected K-L divergence from the true unknown Poisson distribution to the model and show that while the expected K-L divergence to the ML-estimated models seems to be always larger than that of the Audic-Claverie statistic, the most divergence appears for true Poisson distributions with small mean parameter. We also theoretically analyze the effect of regularization of ML estimates in the case of zero observed counts. Our results constitute a rigorous analysis of a situation of great practical importance where the benefits of Bayesian approach can be clearly demonstrated in a quantitative and principled manner.

Keywords: Audic-Claverie statistic, Bayesian averaging, Poisson distribution, Kullback-Leibler divergence, differential gene expression.

1 Introduction

Studies in (computational) learning theory mostly tend to concentrate on situations where potentially ever increasing number of training examples is available. While such results can lead to deep insights into the workings of learning algorithms, e.g. linking together characteristics of the data generating distributions,

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 37–46, 2011.

learning machines and sample sizes, there are situations where, by very nature of the problem, only extremely small samples are available. In such situations it is of utmost importance to theoretically analyze exactly what and under what circumstances can be learned.

An example of such a scenario is detection of differentially expressed genes. One way in which biologists learn about diverse gene functionalities is the analysis of expression levels of selected genes in different tissues, possibly obtained under different conditions or treatment regimes. Even subtle changes in gene expression levels can be indicators of biologically crucial processes [1]. Measurement of gene expression levels can be performed within several frameworks, the one we concentrate on in this paper is termed 'Serial Analysis of Gene Expression' (SAGE) [2]. The SAGE procedure results in a library of short sequence tags, each representing an expressed gene. It is assumed that every mRNA copy in the tissue has the same chance of ending up as a tag in the library. The crucial task is identification of genes that are differentially expressed under different conditions/treatments. This is done by comparing the number of specific tags found in the two SAGE libraries corresponding to different conditions or treatments.

Several statistical tests have been suggested for identifying differentially expressed genes via comparison of digital expression profiles, e.g. [1,3,4,5]. Audic and Claverie [3] studied the influence of random fluctuations and sampling size on the reliability of digital expression profile data in a systematic manner. Even though there have been further developments in comparison techniques for cDNA libraries (e.g. [1,6]), the Audic-Claverie method has been and still continues to be a popular approach used in current biological research (e.g. [7,8,9,10,11]).

When comparing two libraries in the Audic-Claverie framework, it is assumed that under the null hypothesis the tag count x (for a given gene) in one library comes from the same underlying (unknown) Poisson distribution $P(\cdot|\lambda)$ as the tag count y (for the same gene) in the other library. Crucially, since each SAGE library represents a single measurement, the inference has to be performed on the smallest sample possible - sample of size 1! One can, of course, be excused for being highly skeptical about the relevance of such inferences, yet the methodology has apparently been used in a number of successful studies. In an attempt to build theoretical foundations behind such inference schemes, we proved a rather surprising result [12]: The expected K-L divergence from the true unknown Poisson distribution to its *model learned from a single realization* in the Audic-Claverie framework never exceeds 1/2 bit.

In this contribution we extend our previous study [12] by comparing the average behavior of the maximum likelihood and the Audic-Claverie approaches. It turns out that there is a sense in which the advantage of the Bayesian approach taken in the Audic-Claverie framework can be quantified, as a function of the mean λ of the underlying (unknown) Poisson source $P(\cdot|\lambda)$.

The paper has the following organization: The Audic-Claverie approach is briefly introduced in section 2. Section 3 contains theoretical comparison of the

Audic-Claverie and maximum likelihood approaches. Concluding comments are presented in section 4.

2 Bayesian Averaging in the Audic-Claverie Statistic

Consider a random selection of N clones from a cDNA library. For a given message (tag), let x denote the number of times it is picked. When repeating the experiment, possibly under different conditions, by again selecting N clones at random and generating the sequence tags, the same message will be picked y times. Under the null hypothesis, the quantity of interest is the probability of observing y occurrences of a clone already observed x times. For a transcript representing a small fraction of the library and a large number N of clones, the probability of observing x tags of the same gene will be well-approximated by the Poisson distribution parametrized by $\lambda \geq 0$:

$$P(X = x|\lambda) = e^{-\lambda}\frac{\lambda^x}{x!}. \tag{1}$$

The unknown parameter λ signifies the number of transcripts of the given type (tag) per N clones in the cDNA library.

The probability of count y, given the observed count x from the same (unknown) Poisson distribution is:

$$\begin{aligned} P_{AC}(y|x) &= \int_0^\infty P(y|\lambda)\ p(\lambda|x)\ \mathrm{d}\lambda \\ &= \int_0^\infty P(y|\lambda)\ \frac{P(x|\lambda)\ p(\lambda)}{\int_0^\infty P(x|\lambda')\ p(\lambda')\ \mathrm{d}\lambda'}\ \mathrm{d}\lambda. \end{aligned}$$

Imposing flat (improper) prior $p(\lambda)$ over the Poisson parameter λ results in

$$\begin{aligned} P_{AC}(y|x) &= \frac{1}{y!}\frac{\int_0^\infty e^{-2\lambda}\ \lambda^{x+y}\ \mathrm{d}\lambda}{\int_0^\infty e^{-\lambda}\ \lambda^x\ \mathrm{d}\lambda} \\ &= \frac{1}{2^{x+y+1}}\binom{x+y}{x}. \end{aligned} \tag{2}$$

We refer to $P_{AC}(y|x)$ as *Audic-Claverie statistic* (A-C statistic) based on counts x and y. The A-C statistic can be used e.g. for principled inferences, construction of confidence intervals or statistical testing. Note that $P_{AC}(y|x)$ is symmetric, i.e. for $x, y \geq 0,\ P_{AC}(y|x) = P_{AC}(x|y)$, which is quite desirable since if the counts x, y are related to two libraries of the same size, they should be interchangeable when analyzing whether they come from the same underlying process or not. For further details we refer the interested reader to [3].

3 Expected Divergence from the True Underlying Poisson Distribution

Consider a 'true' underlying Poisson distribution $P(y|\lambda)$ (1) over possible counts $y \geq 0$ with unknown parameter λ. We first generate a count x and then use the

A-C statistic $P_{AC}(y|x)$ ([2]) to define a distribution over y, given the already observed count x. We ask: If we repeated the process above, how different, in terms of Kullback-Leibler (K-L) divergence, are on average the two distributions over y? For the A-C statistic to work, one would naturally like $P_{AC}(y|x)$ to be sufficiently representative of the true unknown distribution $P(y|\lambda)$.

In [12] we proved that, given an underlying Poisson distribution $P(x|\lambda)$, if we repeatedly generated a 'representative' count x from $P(x|\lambda)$, the average divergence of the corresponding A-C statistic $P_{AC}(y|x)$ from the truth $P(y|\lambda)$ would never exceed $1/2$ bit.

Theorem 1. *([12]) Consider an underlying Poisson distribution $P(\cdot|\lambda)$ parametrized by some $\lambda > 0$. Then*

$$E_{P(x|\lambda)}[D_{KL}[P(y|\lambda)\|P_{AC}(y|x)]] = \frac{1}{2}\log 2 \ + \ O\left(\frac{1}{\lambda}\right),$$

where $D_{KL}[P(y|\lambda)\|P_{AC}(y|x)]$ is the K-L divergence from $P(y|\lambda)$ to $P_{AC}(y|x)$,

$$D_{KL}[P(y|\lambda)\|P_{AC}(y|x)] = \sum_{y=0}^{\infty} P(y|\lambda) \ \log \frac{P(y|\lambda)}{P_{AC}(y|x)}.$$

The expected divergence (in bits) can be well-approximated (up to order $O(\lambda^{-3})$) by [12]:

$$\begin{aligned} E_{P(x|\lambda)}[D_{KL}[P(y|\lambda)\|P_{AC}(y|x)]] &\approx \frac{1}{2} - \frac{1}{12\lambda}\left(1-\frac{1}{2}\right) - \frac{1}{24\lambda^2}\left(1-\frac{1}{2^2}\right) \\ &= D_{AC}(\lambda). \end{aligned} \tag{3}$$

We now repeat the same analysis with the maximum likelihood estimate $P_{ML}(y|x)$ instead of the A-C statistic $P_{AC}(y|x)$. However, in this case one has to be careful, as Poisson distribution $P(y|\lambda)$ is only defined for positive λ. In the case of observing zero count $x = 0$, we cannot directly use the 'maximum likelihood estimate' $P(y|0)$. Note that no such problem occurs when the A-C statistic is used. $P_{AC}(y|0)$ is well defined with the support shared by all Poisson distributions (the set of positive integers). There are two options for dealing with observed zero count and maximum likelihood estimation of the underlying Poisson distribution:

1. Extend the definition of Poisson distribution to the case $\lambda = 0$ by postulating that the zero count is the only possible outcome:

$$P(0|0) = 1 \text{ and } P(y|0) = 0, \ y \geq 1.$$

Such an extension is in line with the notion that both the mean and variance of $P(y|\lambda)$ are equal to λ. In this case the divergence $D_{KL}[P(y|\lambda)\|P(y|0)]$ is not well defined, since the support of $P(y|\lambda)$, $\lambda > 0$ is not a subset of the support of $P(y|0)$.

2. When $x = 0$ is observed, allow for some form of model regularization, e.g. infer a Poisson model $P(y|\epsilon)$, for some small $\epsilon > 0$: If a count $x \geq 1$ is observed, follow the standard maximum likelihood procedure and infer $P_{ML}(y|x) = P(y|x)$ as the Poisson model (with mean $\lambda = x$). If a zero count is observed, $x = 0$, infer $P_{ML}(y|0) = P(y|\epsilon)$ for some fixed $\epsilon \in (0, 1]$. This is the route we follow in this study.

Let us evaluate the expected divergence between the true Poisson source and its (regularized) maximum likelihood estimate based on a single observation:

$$\Upsilon(\lambda, \epsilon) = E_{P(x|\lambda)}[D_{KL}[P(y|\lambda)\|P_{ML}(y|x)]] = \sum_{x=1}^{\infty} P(x|\lambda)\ D_{KL}[P(y|\lambda)\|P(y|x)] + P(0|\lambda)\ D_{KL}[P(y|\lambda)\|P(y|\epsilon)]. \quad (4)$$

We first note that for $\lambda, \lambda' > 0$, the K-L divergence from $P(y|\lambda)$ to $P(y|\lambda')$ can be evaluated as

$$D_{KL}[P(y|\lambda)\|P(y|\lambda')] = \lambda' - \lambda + \lambda \log \frac{\lambda}{\lambda'}. \quad (5)$$

Also, $P(0|\lambda) = e^{-\lambda}$ and $\sum_{x=1}^{\infty} P(x|\lambda) = 1 - e^{-\lambda}$. We calculate

$$\begin{aligned}\sum_{x=1}^{\infty} P(x|\lambda)\ D_{KL}[P(y|\lambda)\|P(y|x)] &= \sum_{x=1}^{\infty} P(x|\lambda)\ (x - \lambda + \lambda \log \lambda - \lambda \log x) \\ &= -P(x|\lambda) \cdot 0\ + \sum_{x=0}^{\infty} P(x|\lambda)\ x \\ &\quad +\lambda\ (\log \lambda - 1)\ (1 - e^{-\lambda}) \\ &\quad -\lambda \sum_{x=1}^{\infty} P(x|\lambda)\ \log x\end{aligned}$$

to obtain (see (4))

$$\begin{aligned}\Upsilon(\lambda, \epsilon) &= \lambda + \lambda\ (\log \lambda - 1)\ (1 - e^{-\lambda}) - \lambda \sum_{x=1}^{\infty} P(x|\lambda)\ \log x \\ &\quad +e^{-\lambda}\ (\epsilon - \lambda + \lambda \log \lambda - \lambda \log \epsilon) \\ &= \lambda\ \left(\log \lambda - \sum_{x=1}^{\infty} P(x|\lambda)\ \log x\right) + e^{-\lambda}\ (\epsilon - \lambda \log \epsilon). \end{aligned} \quad (6)$$

We further have,

$$\begin{aligned}\log \lambda &= \log \sum_{x=0}^{\infty} P(x|\lambda)\ x \\ &= \log \sum_{x=1}^{\infty} P(x|\lambda)\ x\end{aligned}$$

so that

$$\log\lambda - \sum_{x=1}^{\infty} P(x|\lambda)\ \log x = \log \sum_{x=1}^{\infty} P(x|\lambda)\ x - \sum_{x=1}^{\infty} P(x|\lambda)\ \log x.$$

By Jensen's inequality, we have

$$\begin{aligned}\log \frac{\sum_{x=1}^{\infty} P(x|\lambda)\ x}{\sum_{x=1}^{\infty} P(x|\lambda)} - \frac{\sum_{x=1}^{\infty} P(x|\lambda)\ \log x}{\sum_{x=1}^{\infty} P(x|\lambda)} &= \log \frac{\sum_{x=1}^{\infty} P(x|\lambda)\ x}{1-e^{-\lambda}} \\ &\quad - \frac{\sum_{x=1}^{\infty} P(x|\lambda)\ \log x}{1-e^{-\lambda}} \\ &\geq 0,\end{aligned}$$

we get

$$\log\lambda - \sum_{x=1}^{\infty} P(x|\lambda)\ \log x \geq e^{-\lambda}\ \log\lambda + (1-e^{-\lambda})\ \log(1-e^{-\lambda}). \tag{7}$$

By plugging (7) into (6) we obtain a lower bound on the expected divergence,

$$\begin{aligned}\Upsilon(\lambda,\epsilon) &\geq \lambda\ \left(e^{-\lambda}\ \log\lambda + (1-e^{-\lambda})\ \log(1-e^{-\lambda})\right) \\ &\quad + e^{-\lambda}\ (\epsilon - \lambda\log\epsilon) \\ &= B_{ML}(\lambda;\epsilon).\end{aligned} \tag{8}$$

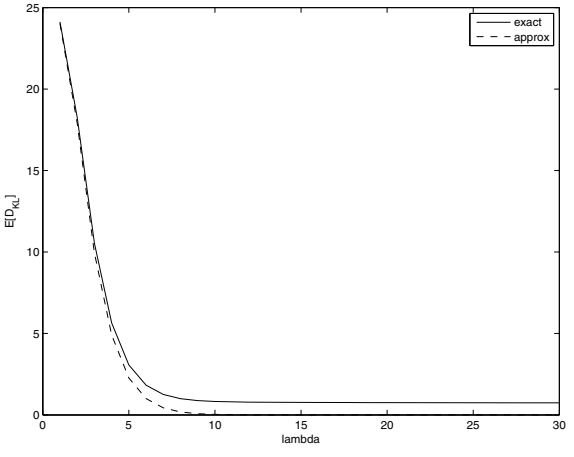


Fig. 1. Expected divergence $E_{P(x|\lambda)}[D_{KL}[P(y|\lambda)\|P_{ML}(y|x)]]$ (in bits) as a function of the mean λ of the underlying Poisson source $P(x|\lambda)$ ($\epsilon = 10^{-20}$). The numerically determined divergence (6) and its analytical approximation (lower bound) $B_{ML}(\lambda;\epsilon)$ are shown as solid and dashed lines, respectively.

As an illustration, we numerically determined the expected divergence[1] $\Upsilon(\lambda, \epsilon)$ (6) for a range of mean parameters λ of the underlying Poisson source $P(x|\lambda)$. The divergence $\Upsilon(\lambda, \epsilon)$ as a function of λ (ϵ is set to 10^{-20}) is shown in figure 1. We also show the analytical lower bound $B_{ML}(\lambda; \epsilon)$ (8) on the expected divergence. For small values of the mean parameter λ, the analytical bound $B_{ML}(\lambda; \epsilon)$ closely approximates $E_{P(x|\lambda)}[D_{KL}[P(y|\lambda)\|P_{ML}(y|x)]]$.

To appreciate the influence of Bayesian averaging in the A-C statistic as opposed to maximum likelihood (ML) estimation, we evaluated the difference between the expected divergences from $P(y|\lambda)$ to maximum likelihood estimates $P_{ML}(y|x)$ and to the A-C statistic $P_{AC}(y|x)$:

$$\begin{aligned} \Delta(\lambda; \epsilon) &= E_{P(x|\lambda)}[D_{KL}[P(y|\lambda)\|P_{ML}(y|x)]] - E_{P(x|\lambda)}[D_{KL}[P(y|\lambda)\|P_{AC}(y|x)]] \\ &= E_{P(x|\lambda)}[E_{P(y|\lambda)}\left[\log \frac{P_{AC}(y|x)}{P_{ML}(y|x)}\right]]. \end{aligned} \tag{9}$$

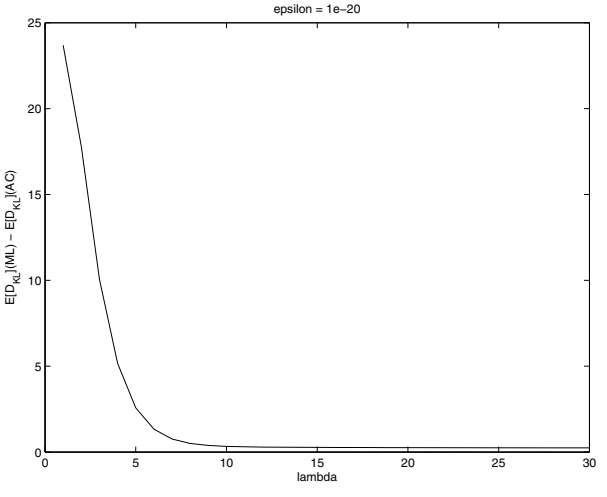


Fig. 2. A graph of $\Delta(\lambda; \epsilon)$ for $\epsilon = 10^{-20}$

A graph of $\Delta(\lambda; 10^{-20})$ is shown in figure 2. The expected K-L divergence to the ML-estimated models seems to be always larger than that of the Audic-Claverie statistic. Furthermore, as expected, the maximum likelihood approach suffers most for true Poisson distributions with small mean parameter due to the highly peaked character of Poisson distributions $P(y|a)$ for small $a > 0$.

In view of the large cost associated with inferring $P_{ML}(y|0) = P(y|\epsilon)$ for small ϵ, one can ask if smaller expected divergences $\Upsilon(\lambda, \epsilon)$ could be obtained, at least for smaller true expected rates λ, if we inferred $P_{ML}(y|0) = P(y|1)$, e.g. if we set $\epsilon = 1$ when $x = 0$ is observed. It turns out that using 'regularization' $\epsilon = 1$

[1] By computing the double expectation up to high values of x and y.

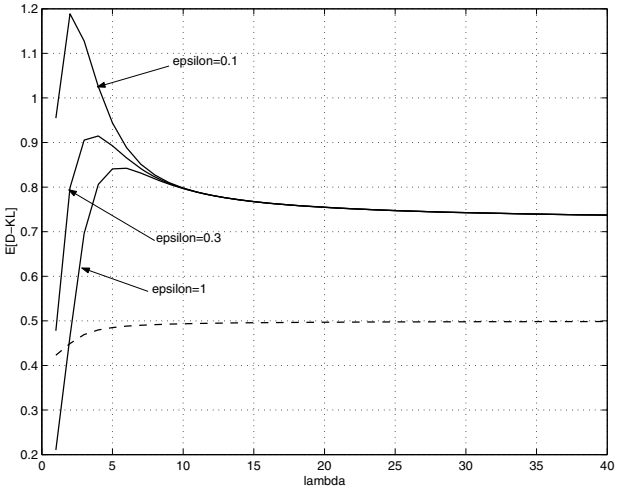


Fig. 3. Expected divergences $\Upsilon(\lambda, \epsilon)$ for a range of moderate regularization parameter values (solid lines). Also shown is the expected divergence of the original A-C statistic (dashed line).

is universally better in maximum likelihood estimation than any $0 < \epsilon < 1$, regardless of the underlying mean rate $\lambda > 0.7$.

Theorem 2. *For any mean rate $\lambda \geq \ln 2$ parametrizing the underlying Poisson distribution $P(\cdot|\lambda)$, the expected divergence $\Upsilon(\lambda, 1)$ using $\epsilon = 1$ is strictly smaller than the one using $0 < \epsilon < 1$, i.e.*

$$\Upsilon(\lambda, 1) < \Upsilon(\lambda, \epsilon) \quad \textit{for any } \epsilon \in (0, 1),\ \lambda \geq \ln 2.$$

Proof. By (6) ,

$$\begin{aligned} \Upsilon(\lambda, \epsilon) - \Upsilon(\lambda, 1) &= e^{-\lambda}\ (\epsilon - \lambda \log \epsilon) - e^{-\lambda} \\ &= e^{-\lambda}\ (\epsilon - 1 - \lambda \log \epsilon). \end{aligned} \tag{10}$$

We ask for which λ the above quantity becomes positive. This happens iff

$$\epsilon - 1 - \lambda \log \epsilon > 0$$

which is equivalent to (note that $\log \epsilon < 0$)

$$\lambda > \lambda(\epsilon) = \frac{1 - \epsilon}{-\log \epsilon} = \frac{(1 - \epsilon) \cdot \ln 2}{-\ln \epsilon}.$$

Now, for $\epsilon \in (0, 1)$, it holds $1 - \epsilon < -\ln \epsilon$ and we have

$$\lambda(\epsilon) < \ln 2 < 1.$$

In figure 3 we show the expected divergences $\Upsilon(\lambda, \epsilon)$ for a range of regularization parameter values. For larger λ, the divergence $\Upsilon(\lambda, \epsilon)$ is dominated by the term $\lambda[\log \lambda - \sum_{x=1}^{\infty} P(x|\lambda) \log x]$, while the small λ regimes are influenced by $e^{-\lambda}$ $(\epsilon - \lambda \log \epsilon)$. Also shown is $E_{P(x|\lambda)}[D_{KL}[P(y|\lambda) \| P_{AC}(y|x)]]$, the expected divergence of the original A-C statistic. Note that, apart from the small λ regime favored by the maximum likelihood regularization, the expected divergence of the original A-C statistic is smaller than that of the ML estimates.

4 Conclusion

There are situations where only extremely small samples can be used. In this study we concentrated on one such scenario - detection of differentially expressed genes. We extended our previous theoretical study [12] of one of the most popular techniques for identifying genes with statistically different expression in cDNA expression arrays - the Audic-Claverie (A-C) statistic [3].

In the Audic-Claverie framework the true unknown Poisson distribution must be learned based on a single observation. As a result of Bayesian averaging employed in the A-C statistic, the expected K-L divergence from the true unknown Poisson distribution to the model never exceeds 1/2 bit [12]. When a (regularized) maximum likelihood (ML) approach is taken, the biggest divergence from the truth occurs for underlying Poisson sources with small mean parameter. This is caused by the abundance of small observed counts and the highly peaked nature of ML-estimated Poisson models at such low counts.

We analytically approximated the expected K-L divergence from the true unknown Poisson distribution to the ML estimates. The analytical approximation closely tracks the expected divergence in the critical region of small mean parameters of the underlying Poisson source. We also showed that it pays off to regularize the ML estimation by inferring Poisson model with mean 1, even though the observed count is 0.

In the future work, the theoretical study presented here will be complemented with large scale experiments using realistic data that is not necessarily strictly Poisson distributed. We will also study and verify more involved regularization schemes, e.g. using gamma prior concentrated on small values of the mean parameter instead of the flat improper prior used in the A-C statistic.

References

1. Varuzza, L., Gruber, A., Pereira, C.A.B.: Significance tests for comparing digital gene expression profiles. Nature Precedings npre.2008.2002.3 (2008)
2. Velculescu, V., Zhang, L., Vogelstein, B., Kinzler, K.: Serial analysis of gene expression. Science 270, 484–487 (1995)
3. Audic, S., Claverie, J.: The significance of digital expression profiles. Genome Res. 7, 986–995 (1997)
4. Ruijter, J., Kampen, A.V., Baas, F.: Statistical evaluation of SAGE libraries: consequences for experimental design. Physiol. Genomics 11, 37–44 (2002)

5. Ge, N., Epstein, C.: An empirical Bayesian significance test of cDNA library data. Journal of Computational Biology 11, 1175–1188 (2004)
6. Stekel, D., Git, Y., Falciani, F.: The comparison of gene expressiom from multiple cDNA libraries. Genome Research 10, 2055–2061 (2000)
7. Medina, C., Rotter, B., Horres, R., Udupa, S., Besser, B., Bellarmino, L., Baum, M., Matsumura, H., Terauchi, R., Kahl, G., Winter, P.: SuperSAGE: the drought stress-responsive transcriptome of chickpea roots. BMC Genomics 9, 553 (2008)
8. Kim, H., Baek, K., Lee, S., Kim, J., Lee, B., Cho, H., Kim, W., Choi, D., Hur, C.: Pepper EST database: comprehensive in silico tool for analyzing the chili pepper (Capsicum annuum) transcriptome. BMC Plant Biology 8, 101–108 (2008)
9. Morin, R., OConnor, M., Griffith, M., Kuchenbauer, F., Delaney, A., Prabhu, A., Zhao, Y., McDonald, H., Zeng, T., Hirst, M., Eaves, C., Marra, M.: Application of massively parallel sequencing to microRNA profiling and discovery in human embryonic stem cells. Genome Research 18, 610–621 (2008)
10. Cervigni, G., Paniego, N., Pessino, S., Selva, J., Diaz, M., Spangenberg, G., Echenique, V.: Gene expression in diplosporous and sexual Eragrostis curvula genotypes with differing ploidy levels. BMC Plant Biology 67, 11–23 (2008)
11. Miles, J., Blomberg, A., Krisher, R., Everts, R., Sonstegard, T., Tassell, C.V., Zeulke, K.: Comparative transcriptome analysis of in vivoand in vitro-produced porcine blastocysts by small amplified RNA-serial analysis of gene expression (SAR-SAGE). Molecular Reproduction and Development 75, 976–988 (2008)
12. Tiňo, P.: Basic properties and information theory of audic-claverie statistic for analyzing cDNA arrays. BMC Bioinformatics 10, 308–310 (2009)

Simultaneous Model Selection and Feature Selection via BYY Harmony Learning

Hongyan Wang and Jinwen Ma*

Department of Information Science, School of Mathematical Sciences & LMAM
Peking University, Beijing, 100871, P.R. China

Abstract. Model selection for Gaussian mixture learning on a given dataset is an important but difficulty task and also depends on the feature or variable selection in practical applications. In this paper, we propose a new kind of learning algorithm for Gaussian mixtures with simultaneous model selection and variable selection (MSFS) based on the BYY harmony learning framework. It is demonstrated by simulation experiments that the proposed MSFS algorithm is able to solve the model selection and feature selection problems of Gaussian mixture learning on a given dataset simultaneously.

Keywords: Gaussian mixtures, Baysian Ying-Yang (BYY) Harmony learning, Model selection, Feature selection, Clustering analysis.

1 Introduction

Finite mixture models [1] are flexible and powerful statistical tools for data analysis and information processing. In fact, they been extensively used in a variety of practical applications such as clustering analysis, image segmentation and speech recognition. Among these applications, the Gaussian mixture model is very popular and very important in theory and practice. In order to solve the problem of Gaussian mixture modeling, several statistical learning methods have been established, such as the EM algorithm [2]-[3]. However, the conventional learning algorithm cannot solve the model selection problem, i.e., to determine the number of Gaussians for a given dataset. When the Gaussian mixture model is applied to clustering analysis, the model selection problem is just to determine the number of clusters for a dataset. Since the number of Gaussians or clusters is not available in the general cases, model selection must be made with the parameter estimation, which is a rather complicated and difficult task [4].

The other crucial problem on Gaussian mixture learning is feature selection. In principle, the more information we have about each individual, the better a learning method is expected to perform. But in practice, some features are noises and may degrade the learning performance, especially in high-dimension circumstances. A genic dataset usually has a limited number of observations with thousands of features. Actually, there are a large number of features which are

* Corresponding author, jwma@math.pku.edu.cn

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 47–56, 2011.

irrelevant to the learning or classification problem. So, feature selection is necessary. In fact, feature selection has been investigated in the context of supervised learning scenarios [5]-[8]. It was shown in [9] that feature selection can improve the performance of a supervised classifier on learning from a limited number of data points. But for unsupervised learning or clustering analysis, because of the lack of labels as guidance, it is rather difficult for a learning method to achieve the feature or variable selection together with the parameter learning.

As the model selection is related to the feature selection on Gaussian mixture learning, it is reasonable to consider the two selection problems simultaneously under a unified framework. In fact, there have been two investigations on this aspect directly for clustering analysis. Martin et al. [10] proposed a simultaneous feature selection and clustering method using mixture models through the concept of feature saliency and the EM algorithm. On the other hand, Li et al. [11] proposed a simultaneous localized feature selection and model detection for Gaussian mixtures by Bayesian variational learning. Now, we try to propose a a simultaneous model selection and variable selection (MSFS) algorithm for Gaussian mixtures based on the Bayesian Ying-Yang (BYY) harmony learning system and theory [12]-[13].

The remainder of this paper is organized as follows. We begin with a brief description of related works on model selection and feature selection in Section 2. In Section 3, we present our simultaneous model selection and feature selection algorithm for Gaussian mixtures. Section 4 contains the experimental results. Finally, we conclude briefly in Section 5.

2 Related Works

2.1 Feature Selection

Feature selection algorithms can be broadly divided into two categories: filters and wrappers. The so-called "filter" approaches select proper features before the learning process or clustering analysis. They evaluate the relevance of each feature to the learning problem using the dataset alone [14]-[15]. Independent selection of the features may influence the effect of learning or clustering. On the other hand, the so-called "wrapper" approaches combine the learning or clustering algorithm with evaluating the quality of each feature. Specifically, a learning algorithm (distance-based [16]-[17] or model-based [18]-[19]) can be implemented for each feature subset. Then this feature subset is evaluated by the performance of learning or clustering. From this point of view, the "wrappers" approaches are usually more computationally demanding since they evaluate all feature subsets.

Intuitionally, feature selection is choosing relevant features, and there are many definitions of feature irrelevancy for supervised learning, such as the correlation or mutual information. Here, we adopt such a definition of feature irrelevancy for unsupervised learning that the i-th variable is irrelevant if its distribution is independent of the class labels. This means that the i-th variable

is irrelevant when it comes from a common distribution $p(y_l|\lambda_l)$ which is independent with labels. By contrast, we define the density of a relevant feature l by $p(y_l|\theta_{jl})$ for j-th component of the mixture model. Based on these definitions, if we assume that these variables are independent, the likelihood function can be written as the following form (refer to [10],[11]):

$$p(y|\theta) = \sum_{j=1}^{k} \alpha_j p(y|\theta_j) = \sum_{j=1}^{k} \alpha_j \prod_{l=1}^{D} (\rho_l p(y_l|\theta_{jl}) + (1-\rho_l) q(y_l|\lambda_l)), \quad (1)$$

where ρ_l is the probability that i-th feature is relevant and θ_{jl} and λ_l are the parameters.

2.2 Model Selection

The traditional approaches to solving the compound Gaussian mixture modeling problem of model selection and parameter learning or estimation are to choose an optimal number k^* of Gaussians as the clusters in the dataset via one of the information, coding and statistical selection criteria such as the famous Akaike's Information Criterion [20], Bayesian Inference Criterion (BIC) [21], Minimum Description Length (MDL) [22], and Minimum Message Length (MML) [23]. Among them, Akaike's information criterion (AIC) and the MML criterion are often used. However, the validating processes of these approaches are computationally expensive because we need to repeat the entire parameter learning process at a large number of possible values of k, i,e, the number of Gaussians in the mixture. Moreover, these existing selection criteria have their limitations.

Since the 1990s, there have appeared some statistical learning approaches to solving this compound modeling problem. The first approach is to utilize certain stochastic simulations to infer the optimal mixture model. Two typical implementations are the methods of Dirichlet processes [24] and reversible jump Markov chain Monte Carlo (RJMCMC) [25]. These stochastic simulation methods generally require a large number of samples through different sampling rules. The second approach is the Bayesian model search based on optimizing the variational bounds [26]-[27]. This approach implements a new selection criterion with the Bayesian variation bound. The third approach is unsupervised learning [28] on finite mixtures (including Gaussian mixture as a particular case) which introduces certain competitive learning mechanism into the EM algorithm such that the model selection can be made adaptively during parameter learning by annihilating the components with very small mixing proportions via the MML criterion. Recently, the Bayesian Ying-Yang (BYY) harmony learning system and theory [12]-[13] haven been developed as a unified statistical learning framework and provided a new statistical learning mechanism that makes model selection adaptively during parameter learning for Gaussian mixtures [29]-[32]. In the following, we will use the BYY harmony learning system as our unsupervised learning framework for Gaussian mixtures.

3 Simultaneous Model Selection and Feature Selection

3.1 BYY Harmony Learning for Gaussian Mixtures

A BYY system describes each observation $x \in \mathcal{X} \subset \Re^n$ and its corresponding inner representation $y \in \mathcal{Y} \subset \Re^m$ via the two types of Bayesian decomposition of the joint density: $p(x, y) = p(x)p(y|x)$ and $q(x, y) = q(y)q(x|y)$, which are called Yang machine and Ying machine, respectively. Given a sample dataset $D_x = \{x_t\}_{t=1}^N$ from the Yang or observable space, the goal of harmony learning on a BYY system is to extract the hidden probabilistic structure of x with the help of y from specifying all aspects of $p(y|x)$, $p(x)$, $q(x|y)$ and $q(y)$ via a harmony learning principle implemented by maximizing the following functional:

$$H(p||q) = \int p(y|x)p(x) \ln[q(x|y)q(y)]dxdy. \tag{2}$$

For the Gaussian mixture model with a given sample dataset $D_x = \{x_t\}_{t=1}^N$, we can utilize the following specific Bi-architecture of the BYY learning system. The inner representation y is discrete in $\mathcal{Y} = \{1, 2, \cdots, k\}$ (i.e., with $m = 1$), while the observation x is continuous from a Gaussian mixture distribution. On the Ying space, we let $q(y = j) = \pi_j \geq 0$ with $\sum_{j=1}^k \pi_j = 1$. On the Yang space, we suppose that $p(x)$ is a latent probability density function (pdf) of Gaussian mixture, with a set of sample data D_x being generated from it. Moreover, in the Ying path, we let each $q(x|y = j) = q(x|m_j, \Sigma_j)$ be a Gaussian probability density with the mean vector m_j and the covariance matrix Σ_j, while the Yang path is constructed under the Bayesian principle by the following parametric form:

$$p(y = j|x) = \frac{\pi_j q(x|m_j, \Sigma_j)}{q(x|\Theta_k)}, \qquad q(x|\Theta_k) = \sum_{j=1}^k \pi_j q(x|m_j, \Sigma_j), \tag{3}$$

where $\Theta_k = \{\pi_j, m_j, \Sigma_j\}_{j=1}^k$ and $q(x|\Theta_k)$ is just a Gaussian mixture model that will approximate the true Gaussian mixture model $p(x)$ hidden in the sample data D_x via the harmony learning on the BYY learning system.

With all these component densities into Eq.(2), we get an estimate of $H(p||q)$ as the following harmony function for Gaussian mixtures with the parameter set Θ_k:

$$J(\Theta_k) = \frac{1}{N} \sum_{t=1}^N \sum_{j=1}^k \frac{\pi_j q(x_t|m_j, \Sigma_j)}{\sum_{i=1}^k \pi_i q(x_t|m_j, \Sigma_j)} \ln[\pi_j q(x_t|m_j, \Sigma_j)]. \tag{4}$$

According to theoretical and experimental results on the BYY harmony learning on the BI-architecture for Gaussian mixtures [29]-[30],[32]-[33], the maximization of the harmony function $J(\Theta_k)$ is able to make model selection adaptively during parameter learning when the actual Gaussians in the sample data are separated in a certain degree. That is, in such a situation, if we set k to be larger than the number k^* of actual Gaussians in the sample data, the maximization of the harmony function can make k^* Gaussians from the estimated mixture match the actual Gaussians, respectively, and force the mixing proportions of $k - k^*$ extra Gaussians to attenuate to zero.

3.2 Proposed BYY Harmony Learning Algorithm

By a transformation, $J(\Theta_k)$ can be divided into two parts:

$$J(\Theta_k) = L(\Theta_k) - O_N(p(y|x)), \tag{5}$$

where the first part is just the log-likelihood function:

$$L(\Theta_k) = \frac{1}{N}\sum_{t=1}^{N} \ln(\sum_{j=1}^{k}(\pi_j q(x_t|m_j, \Sigma_j))), \tag{6}$$

while the second part is the average Shannon entropy of the posterior probability $p(y|x)$ over the sample dataset $\mathcal{D} = \{x_t\}_{t=1}^N$:

$$O_N(p(y|x)) = -\frac{1}{N}\sum_{t=1}^{N}\sum_{j=1}^{k} p(j|x_t)\ln p(j|x_t). \tag{7}$$

According to Eq.(5), if $-O_N(p(y|x))$ is considered as a regularization term, the BYY harmony learning, i.e., maximizing $J(\Theta_k)$, is a kind of regularized ML learning. This regularization term contributes to avoiding over-fitting and achieving model selection.

If we replace the Likelihood part with (1) and assume that these variables are independent, the maximization of $J(\Theta_k)$ will be able to make model selection and feature selection simultaneously.

$$\begin{aligned} J(\Theta_k) = &\frac{1}{N}\sum_{t=1}^{N} \log(\sum_{j=1}^{k} \alpha_j \prod_{l=1}^{D}(\rho_l p(x_{tl}|\theta_{jl}) + (1-\rho_l)q(x_{tl}|\lambda_l))) \\ &+\frac{1}{N}\sum_{t=1}^{N}\sum_{j=1}^{k} p(j|x_t)\log p(j|x_t), \end{aligned} \tag{8}$$

where θ_{jl}, λ_l are the parameters of Gaussian densities and

$$p(j|x_t) = \frac{\alpha_j \prod_{l=1}^{D}(\rho_l p(x_{tl}|\theta_{jl}) + (1-\rho_l)q(x_{tl}|\lambda_l))}{\sum_{i=1}^{k}\alpha_i \prod_{l=1}^{D}(\rho_l p(x_{tl}|\theta_{il}) + (1-\rho_l)q(x_{tl}|\lambda_l))}. \tag{9}$$

Actually, there have been many learning algorithms to maximize $J(\theta)$. Here, we adopt the fixed-point learning paradigm (refer to [32]) and the learning algorithm can be derived as follows.

Define:

$$\gamma_j(t) = 1 + \log p(j|x_t) - \sum_{i=1}^{k} p(i|x_t)\log p(i|x_t) \tag{10}$$

$$u_{tjl} = \frac{\rho_l p(x_{tl}|\theta_{jl})}{\rho_l p(x_{tl}|\theta_{jl}) + (1-\rho_l)q(x_{tl}|\lambda_l)}; \qquad v_{tjl} = 1 - u_{tjl} \tag{11}$$

By derivation, we have the derivatives of $J(\Theta_k)$ with respect to α_j, θ_{jl} and λ_l, respectively. Letting these derivatives be equal to zero, we get the following fixed-point equations:

$$\hat{\alpha_j} = \frac{\sum_{t=1}^{N} p(j|x_t)\gamma_j(t)}{\sum_{j=1}^{k}\sum_{t=1}^{N} p(j|x_t)\gamma_j(t)} \tag{12}$$

$$\widehat{mean\ in}\ \theta_{jl} = \frac{\sum_{t=1}^{N} p(j|x_t)\gamma_j(t)u_{tjl}x_{tl}}{\sum_{t=1}^{N} p(j|x_t)\gamma_j(t)u_{tjl}} \tag{13}$$

$$\widehat{var\ in}\ \theta_{jl} = \frac{\sum_{t=1}^{N} p(j|x_t)\gamma_j(t)u_{tjl}(x_{tl} - \widehat{mean\ in}\ \theta_{jl})^2}{\sum_{t=1}^{N} p(j|x_t)\gamma_j(t)u_{tjl}} \tag{14}$$

$$\widehat{mean\ in}\ \lambda_l = \frac{\sum_{t=1}^{N}\sum_{j=1}^{k} p(j|x_t)\gamma_j(t)v_{tjl}x_{tl}}{\sum_{t=1}^{N}\sum_{j=1}^{k} p(j|x_t)\gamma_j(t)v_{tjl}} \tag{15}$$

$$\widehat{var\ in}\ \lambda_l = \frac{\sum_{t=1}^{N}\sum_{j=1}^{k} p(j|x_t)\gamma_j(t)v_{tjl}(x_{tl} - \widehat{mean\ in}\ \lambda_l)^2}{\sum_{t=1}^{N}\sum_{j=1}^{k} p(j|x_t)\gamma_j(t)v_{tjl}} \tag{16}$$

As for ρ_l, since the gradient and Hessian of $J(\Theta_k)$ are very complicated (refer to the appendix), we can use the constrained nonlinear optimization software to find their optimal values in [0,1].

Summarizing the upper results, we obtain a two-step iterated optimization algorithm:

Step 1: Fix ρ_l and estimate α_j, θ_{jl} and λ_l according (12)-(16);

Step 2: Fix α_j, θ_{jl} and λ_l and obtain the optimized ρ_l using the constrained nonlinear optimization software.

In our learning paradigm, k is flexible. However, it should be larger than the true number k^* of Gaussians in the dataset. Then when we repeat these two steps to make the parameter estimation, we cut very weak components (with $\alpha_j < 0.05$) synchronously.

4 Experimental Results

4.1 Three Synthetic Datasets

The first dateset consists of 800 8-dimensional data or points from two classes whose distributions are subject to $\mathcal{N}(m_i, 2.25{*}I)$, $i = 1, 2$, where I is the identity matrix, $m_1 = (5, 5, 5, 5, 0, 0, 0, 0)$ and $m_2 = (-5, -5, -5, -5, 0, 0, 0, 0)$. In this situation, it is clear that the first four features are informative and the rest four are irrelevant or noises. We check whether our proposed algorithm can distinguish the informative features from the irrelevant ones. The second dataset consists of 1000 10-dimensional data from two classes. For the two classes, their i-th features come from $\mathcal{N}(1/i, 1)$ and $\mathcal{N}(-5/i, 1)$ respectively, for $i = 1, 2, \cdots, 10$. Note that the features are arranged in a descending order of relevance. This design is more challenging for testing the feature selection. The third dataset is an imbalanced one containing three classes with 200, 400 and 200 samples, respectively.

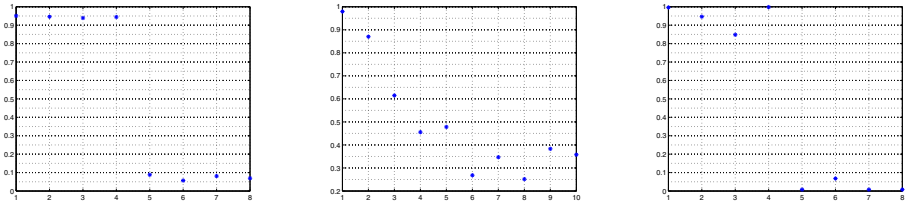


Fig. 1. The sketches of estimated ρ_l for (a). Dataset 1, (b). Dataset 2, (c). Dataset 3.

The Distributions of the three classes are $\mathcal{N}(m_i, 1.414 * I)$, $i = 1, 2, 3$, where I is still the identity matrix, $m_1 = (5, 5, 5, 5, 0, 0, 0, 0)$, $m_2 = (0, 0, 0, 0, 0, 0, 0, 0)$ and $m_3 = (-5, -5, -5, -5, 0, 0, 0, 0)$.

4.2 Simulation Results

We repeat our proposed model selection and feature selection algorithm for Gaussian mixtures on each of these three datasets 50 times with the parameters being randomly initialized. Here, our attention is just focused on the feature selection. So, we only show the average value of ρ_l over 50 experimental results in Fig. 1. It can be seen that our proposed algorithm can successfully distinguish the informative features from the noises, especially for the first dataset on which the last four irrelevant features are found out exactly. The average values of ρ_l are in a descending order just as those features in the second dataset are designed in a descending order of relevance. It can be also noticed that there are some fluctuations along the downtrend. This may be caused by the local optimization of the modified harmony function and can be solved by some global optimization technique.

As for model selection, when k is set to be $2k^*$ (k^* is the true number of Gaussians or classes in the dataset), our proposed algorithm achieves the classification accuracy rates of 66% ,82% and 76% over the three datasets, respectively. Clearly, the model selection result on the first dataset is not so satisfied. In fact, the structure of the first dataset is indeed complicated. For comparison, we implement the RPCL algorithm [34] on the first dataset and generally get a poor clustering result. As for the third dataset, 36 tries out of 50 make the correct model selection and the rest 14 tries lead to 4 clusters. Actually, the largest component is split into two clusters.

5 Conclusions

We have investigated the problem of simultaneous model selection and feature selection for Gaussian mixtures and proposed a new BYY harmony learning algorithm for solving it. The proposed algorithm is constructed in the fixed-point learning paradigm. It is demonstrated by the simulation experiments that the

proposed algorithm can simultaneously detect the number of actual Gaussians in the dataset and recognize the informative features accurately.

Acknowledgements

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

References

1. McLachlan, G.J., Peel, D.: Finite Mixture Models. Wiley, New York (2000)
2. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society B 39, 1–38 (1977)
3. Render, R.A., Walker, H.F.: Mixture densities, maximum likelihood and the EM algorithm. SIAM Review 26(2), 195–293 (1984)
4. Hartigan, J.A.: Distribution problems in clustering. In: Garrett, J. (ed.) Classification and Clustering, pp. 45–72. Academic Press, New York (1977)
5. Blum, A., Langley, P.: Selection of Relevant Features and Examples in Machine Learning. Artificial Intelligence 97(1-2), 245–271 (1997)
6. Kohavi, R., John, G.H.: Wrapper for feature subset selection. Articial Intelligence 97, 273–324 (1997)
7. Jain, A., Zongker, D.: Feature Selection: Evaluation, Application, and Small Sample Performance. IEEE Trans. Pattern Analysis and Machine Intelligence 19(2), 153–157 (1997)
8. Koller, D., Sahami, M.: Toward optimal feature selection. In: Proc. 13th Int'l Conf. Machine Learning, pp. 284–292 (1996)
9. Raudys, S.J., Jain, A.K.: Small sample size effects in statistical pattern recognition: recommandations for practitioners. IEEE Trans. Pattern Analysis and Machine Intelligence 13(3), 252–264 (1991)
10. Law, M.H.C., Figueiredo, M.A.T., Jain, A.K.: Simultaneous Feature Selection and Clustering Using Mixture Models. IEEE Trans. on Pattern Analysis and Machine Intelligence 26(9), 1154–1166 (2004)
11. Li, Y., Dong, M., Hua, J.: Simultaneous Localied Feature Selection and Model Detection for Gaussian Mixtures. IEEE Trans. on Pattern Analysis and Machine Intelligence 31(5), 953–960 (2009)
12. Xu, L.: Best harmony, unified RPCL and automated model selection for unsupervised and supervised learning on Gaussian mixtures, three-layer nets and ME-RBF-SVM models. International Journal of Neural Systems 11, 43–69 (2001)
13. Xu, L.: BYY harmony learning, structural RPCL, and topological self-organizing on mixture modes. Neural Networks 15, 1231–1237 (2002)
14. Dash, M., Choi, K., Scheuermann, P., Liu, H.: Feature Selection for Clustering - A Filter Solution. In: Second IEEE International Conference on Data Mining (ICDM 2002), p. 115 (2002)
15. Jouve, P.-E., Nicoloyannis, N.: A Filter Feature Selection Method for Clustering. In: Hacid, M.-S., Murray, N.V., Raś, Z.W., Tsumoto, S. (eds.) ISMIS 2005. LNCS (LNAI), vol. 3488, pp. 583–593. Springer, Heidelberg (2005)

16. Fowlkes, E.B., Gnanadesikan, R., Kettering, J.R.: Variable selection in clustering. Journal of Classification 5, 205–228 (1988)
17. Devaney, M., Ram, A.: Eficient feature selection in conceptual clustering. Machine Learning. In: Proceedings of the Fourteenth International Conference, Nashville, TN, pp. 92–97 (1997)
18. Tadesse, M.G., Sha, N., Vannucci, M.: Bayesian Variable Selection in Clustering High-Dimensional Data. Journal of the American Statistical Association 100, 602–617 (2005)
19. Kim, S., Tadesse, M.G., Vannucci, M.: Variable selection in clustering via Dirichlet process mixture models. Biometrika 93, 321–344 (2006)
20. Akaike, H.: A new look at statistical model identification. IEEE Transactions on Automatic Control AC-19, 716–723 (1974)
21. Scharz, G.: Estimating the dimension of a model. The Annals of Statistics 6, 461–464 (1978)
22. Rissanen, J.: Modeling by shortest data description. Automatica 14, 465–471 (1978)
23. Wallace, C., Dowe, D.: Minimum Message Length and Kolmogorov Complexity. Computer Journal 42(4), 270–283 (1999)
24. Escobar, M.D., West, M.: Bayesian density estimation and inference using mixtures. Journal of the American Statistical Association 90(430), 577–588 (1995)
25. Recgardson, S., Green, P.J.: On Bayesian analysis of mixtures with an unknown number of components. Journal of the Royal Statistical Soceity B 59(4), 731–792 (1997)
26. Ueda, N., Ghahramani, Z.: Bayesian model search for mixture models based on optimizing variational bounds. Neural Networks 15(10), 1123–1241 (2002)
27. Constantinopoulos, C., Likas, A.: Unsupervised learning of Gaussian mixtures based on variational component splitting. IEEE Trans. on Neural Networks 18(3), 745–755 (2007)
28. Figueiredo, M.A.T., Jain, A.K.: Unsupervised learning of finite mixture models. IEEE Trans. on Pattern Analysis and Machine Intelligence 24(3), 381–396 (2002)
29. Ma, J., Wang, T., Xu, L.: A gradient BYY harmony learning rule on Gaussian mixture with automated model selection. Neurocomputing 56, 481–487 (2004)
30. Ma, J., Wang, L.: BYY harmony learning on finite mixture: adaptive gradient implementation and a floating RPCL mechanism. Neural Processing Letters 24(1), 19–40 (2006)
31. Ma, J., Liu, J.: The BYY annealing learning algorithm for Gaussian mixture with automated model selection. Pattern Recognition 40, 2029–2037 (2007)
32. Ma, J., He, X.: A fast fixed-point BYY harmony learning algorithm on Gaussian mixture with automated model selection. Pattern Recognition Letters 29(6), 701–711 (2008)
33. Ma, J.: Automated model selection (AMS) on finite mixtures: a theoretical analysis. In: Proc. 2006 International Joint Conference on Neural Networks (IJCNN 2006), Vancouver, Canada, July 16-21, pp. 8255–8261 (2006)
34. Ma, J., Wang, T.: A cost-function approach to rival penalized Competitive learning (RPCL). IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics 36(4), 722–737 (2006)

Appendix

Define:

$$h_{tjl} = \frac{p(x_{tl}|\theta_{jl}) - q(x_{tl}|\lambda_l)}{\rho_l p(x_{tl}|\theta_{jl}) + (1-\rho_l) q(x_{tl}|\lambda_l)}.$$

The gradient and Hessian of $J(\Theta_k)$ with respect to ρ_l are:

$$\frac{\partial J(\theta)}{\partial \rho_l} = \frac{1}{N} \sum_{t=1}^{N} \sum_{j=1}^{k} p(j|x_t) \gamma_j(t) h_{tjl};$$

if $l \neq m$,

$$\frac{\partial^2 J(\theta)}{\partial \rho_l \partial \rho_m} = \frac{1}{N} \sum_{t=1}^{N} \sum_{j=1}^{k} p(j|x_t) \Big[h_{tjm} h_{tjl} \gamma_j(t) + h_{tjm} h_{tjl} - \sum_{i=1}^{k} p(i|x_t) h_{tim} h_{tjl} \gamma_j(t) - \sum_{i=1}^{k} p(i|x_t) h_{til} \gamma_j(t) h_{tjm} \Big].$$

if $l = m$,

$$\frac{\partial^2 J(\theta)}{\partial \rho_l^2} = \frac{1}{N} \sum_{t=1}^{N} \sum_{j=1}^{k} p(j|x_t) \Big[h_{tkl} - 2 \sum_{i=1}^{k} p(i|x_t) h_{til} \gamma_i(t) \Big] h_{til}.$$

The Characteristics Study on LBP Operator in Near Infrared Facial Image

Qiang Chen and Weiqing Tong

Department of Computer Science and Technology,
East China Normal University,
500 Dongchuan Road, Shanghai 200241, P.R. China
qzschen@gmail.com, wqtong@cs.ecnu.edu.cn

Abstract. Serving as an effective texture description operator, local binary pattern (LBP) has been applied in the visible face recognition successfully. In order to enhance the recognition performance, the technology of face recognition depending on the near infrared (NIR) image has attracted extensive attention in the recent years. Although the characteristics of LBP have been researched in optical image thoroughly, the development of these do not catch enough attention in the NIR image at all. Therefore, in our paper, we study the characteristics of LBP from the following two aspects in NIR image. On one hand, we come to the probability distribution of various patterns of LBP in the NIR facial image. On the other hand, we discuss the influence to LBP caused by the illumination change in facial image.

Keywords: Local binary pattern, Face recognition, Near infrared facial image.

1 Introduction

Local binary pattern (LBP) [1]is a kind of effective texture description operator, which has excellent robustness to the texture image rotation and the gray variation. On the basis of above properties, LBP operator is widely applied in the texture classification [2,3], texture segmentation [4] and facial image analysis [5,6,7,8].

Generally speaking, we can use symbol $\mathrm{LBP}_{P,R}$ represent the LBP operator in the domain with a radius R of P pixels. As a result, $\mathrm{LBP}_{P,R}$ can produce 2^p kinds of binary patterns. At the same time, with the increasing of parameter P, the number of binary patterns will increase greatly. Due to the reason above, the usage of a large P in LBP operator is disadvantageous to the texture extraction, identification, classification or information access obviously. Consequently, the operator $LBP_{8,1}$ [2] is always used basically in texture analysis and recognition.

As Ojala [2] mentioned, although LBP is an useful texture description operator, it can take a greatly different effect on various texture images. Moreover, face recognition is a challenging field both in theoretical level or technical level and the problem of how to reduce the influence caused by illumination in face recognition has troubled researchers continuously. However, recently, the researchers

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 57–65, 2011.

have made use of active near infrared imaging technology [9,10,11,12] to solve the problem successfully. For the purpose of using LBP operator in face recognition with NIR more efficiently, it is necessary for us to research its basic characteristics according to this kind of images.

We firstly design an NIR facial image capturing camera, which consists by 36 LEDs ($850nm$), visible filter and camera etc. This kind of camera can shield off visible and only shoot $850nm$ band near infrared images. Then, we use it to establish the libraries of NIR facial images which contains 34160 images of 854 persons in more positive direction. Through these library, we study the characteristics of operator $LBP_{8,1}$ in NIR facial image in the following two points. The first point is the probability distribution of various patterns about LBP in the NIR facial image. The second one is the influence to LBP caused by the illumination change in facial image.

The remainder of this paper is organized as follows. In Section 2, we introduce LBP operator. Following this, in Section 3 we introduce the NIR facial image capturing camera and facial image libraries. Then, the experiment and research in characteristics of LBP are introduced in Section 4 followed by the conclusions in Section 5.

2 LBP Operator

In Fig. 1, the calculation principle of operator is indicated and details about it can be seen as following. At first, we select a pixel as the center and regard the grey value of the mid-point as the threshold. After comparing the gray value of pixel with its neighborhood, if it is greater than the threshold, we set point-value as 1, or it is 0. Thus, we can obtain a series of binary yards. This group of yards can get a LBP value from the formula below.

$$\mathrm{LBP}_{P,R}(x_c, y_c) = \sum_{p=0}^{7} s(g_p - g_c)2^p, where \;\; s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} . \qquad (1)$$

Here, x_c and y_c are the coordinates of the pixel. g_c is the gray value of the center-point, g_p is the gray value of each pixel in 3×3-neighborhood.

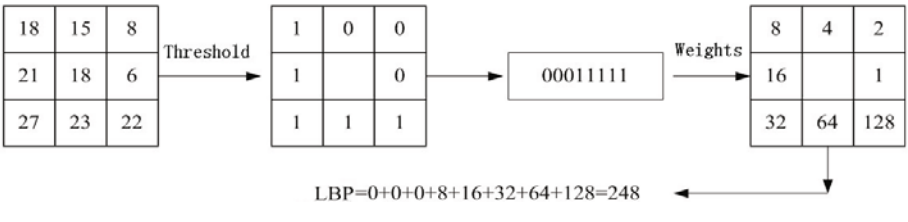


Fig. 1. The example of LBP operator

The value of $LBP_{8,1}$ ranges from 0 to 255, which totally contains 256 kinds of binary patterns. According to the number of spatial transitions (bitwise 0/1 changes) in the patterns, we can divide them into five types. In this paper, we assume that if this kind of value is equal to $0\,, 2\,, 4\,, 6\,, 8$ respectively, we call it type $U_0\,, U_2\,, U_4\,, U_6\,, U_8$ correspondingly.

3 The NIR Facial Image Capturing Camera and Facial Image Libraries

3.1 Process of Designing NIR Facial Image Capturing Camera

In order to establish the libraries of near infrared facial images, we need design NIR facial image capturing camera on the basis of following requirements. Firstly, it is essential to put LED around the camera so as to emit positive light. Secondly, we can filter out the visible light. Then, the requirement of imaging only in $850nm$ band spot needs to be reached. Finally, it is important to minimize the influence owing to the existence of $850nm$ band in the ambient light.

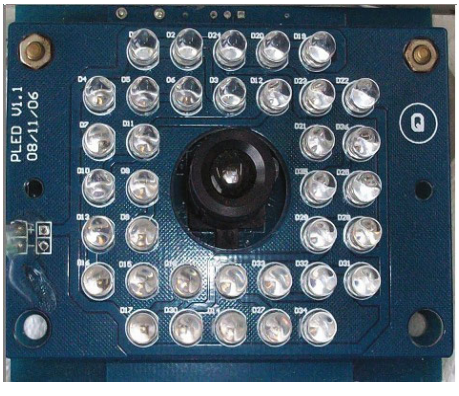
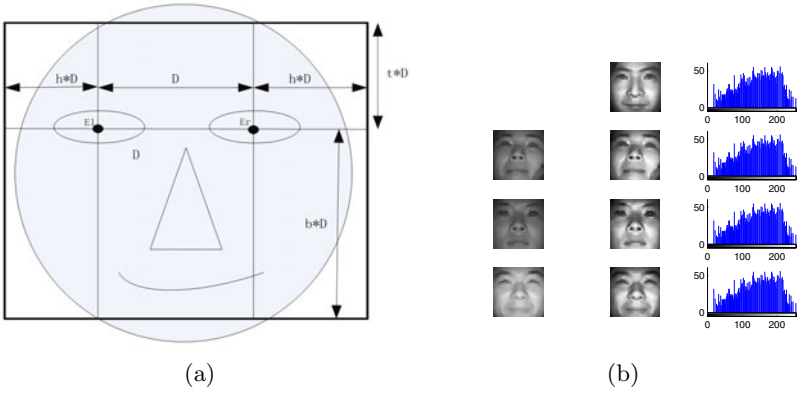


Fig. 2. The NIR facial image capturing camera

According to the technical requirements above, we make the NIR facial image capturing camera, which can be seen in Fig. 2. The imaging device which can shield off visible light and only catch the $850 \pm 5nm$ band near infrared is composed of 36 LEDs ($850nm$), visible filter and camera etc.

3.2 Facial Image Libraries

Owing to the fact that the shooting distance and the difference of shooting angle can cause inconsistency of face size, we need to normalize the facial images to explore LBP properties. The proportion about the normalization of face is shown in Fig. 3(a). In this figure, E_l and E_r represent the left and right pupils position



(a) (b)

Fig 3. **(a)**The ratio figure of face normalization. **(b)**The example of histogram matching.

Table 1. The experimental image libraries

Image library	Sum of person	Images per person	Sum
$Set1$	854	40	34160
$Set2$	854	1	854
$Set1\_Q$	854	40	34160
$Set2\_Q$	854	1	854
$Set1\_M$	854	40	34160
$Set2\_M$	854	1	854

Note: The column three indicates the sum of images per person

separately and D represents pupils spacing. In addition, t and b represent for the proportion from pupil to the top and bottom in the face rectangle respectively. Moreover, h is the proportion for left and right pupils to its edge. In our study, $t = 0.43, b = 1.33, h = 0.53$ and the size of face normalization is 64×64.

We use the above camera to capture the images of 854 Chinese (40 images for every person) and generate six facial image libraries (see $Table\,1$). In these libraries, the angle range of left-right rotation and upper-lower rotation are respectively $-40° \sim 40°$ and $-30° \sim 30°$. The image size is 640×480. After that, we can create the following facial image libraries. According to the above face normalized proportion ,we normalize all the images in NIR face libraries to 64×64 size-scale images as $Set1$. Then, we pick an image from every person in $Set1$ randomly to come to $Set2$ which has 854 images. Thirdly, we do histogram equalization processing for $Set1$ and $Set2$ respectively to generate $Set1\_Q$ and $Set2\_Q$. Finally, we also do histogram matching processing for $Set1$ and $Set2$ to generate $Set1\_M$ and $Set2\_M$.

Fig. 4. (a) Parts of the original NIR facial images. **(b)** the corresponding normalized facial images of (a).

We can see the histogram matching processed image in Fig. 3(b) clearly. The second and third column in the first row are the object histogram matching images and their histograms. In row2 to row4, the first column are original images and the second are their images after histogram matching with their corresponding histograms in third column. Fig. 4(a) shows a part of the original facial images from $Set1$ and their normalized images are shown in Fig. 4(b).

4 Experiments and Results

4.1 Probability Distribution of LBP in the NIR Facial Image Libraries

Operator $\mathrm{LBP}_{8,1}$ has 2^8 kinds of binary patterns. Obviously, the probability distribution of 256-*patterns* is closely related to the texture images, which means

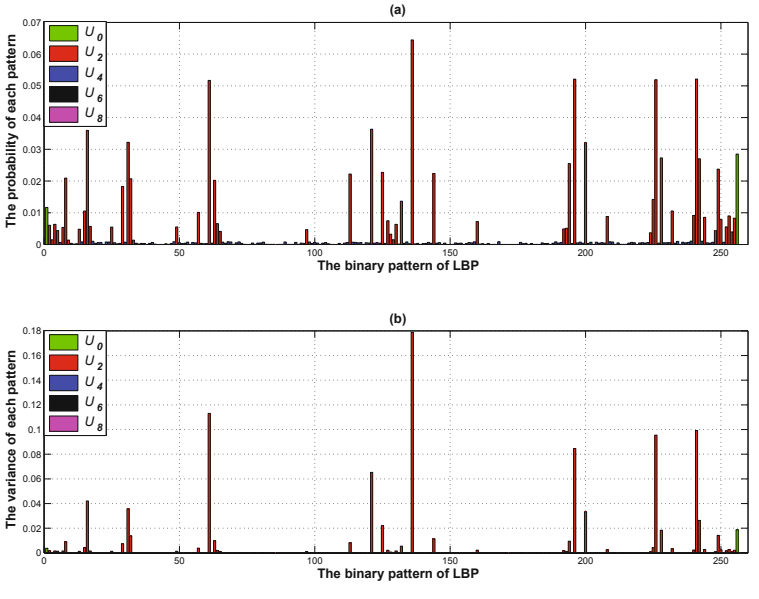


Fig. 5. (a) The scattergram of average probability of 256-*patterns* in LBP. **(b)** The variogram of average probability of 256-*patterns* in LBP.

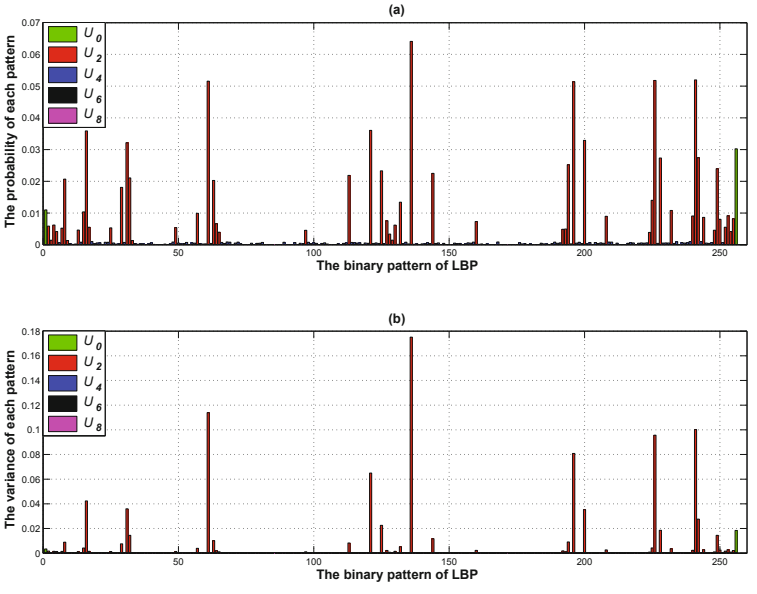


Fig. 6. (a) The scattergram of average probability of 256-*patterns* in LBP after histogram equalization. **(b)** The variogram of average of 256-*patterns* in LBP after histogram equalization.

that different texture images have different probability distribution. In the experiment, we use $Set1$ to research the probability distribution of LBP in such specific texture images. Firstly, we transform all NIR images into LBP images by calculating LBP value which ranges from 0 to 255. Secondly, on the basis of all binary patterns in $\mathrm{LBP}_{8,1}$, we get the probability distribution for every image through calculating the normalization histogram of corresponding LBP images respectively. Then, we get the average probability distribution and its corresponding probability variance of 256-*patterns* in $Set1$.

Fig. 5(a) shows the average probability distribution of 256-*patterns* of $\mathrm{LBP}_{8,1}$ in $Set1$ and (b) is its corresponding probability variance. After taking knowledge about these figures, we can draw the following important conclusions about LBP. Firstly, the probability of the patterns in U_0 and U_2 accounts for only 23%, which means type U_4, U_6 and U_8 account for 76% in all 256 patterns. Nevertheless, the probability of occurrence about U_0 and U_2 is up to 93% and the rest three types only contain 7% at all. Secondly, in one facial image, the maximum probability of occurrence of the patterns in U_4, U_6 and U_8 is 0.1% and the average one is just 0.036%, which indicates that the probability of occurrence of five bitwise types except U_0 and U_2 is less than 1.5 pixels in a LBP image with size-scale 64×64. In addition, an interesting point here is that we find probability variance of U_0 and U_2 is fairly large and the rests are very smaller. This suggests that slight

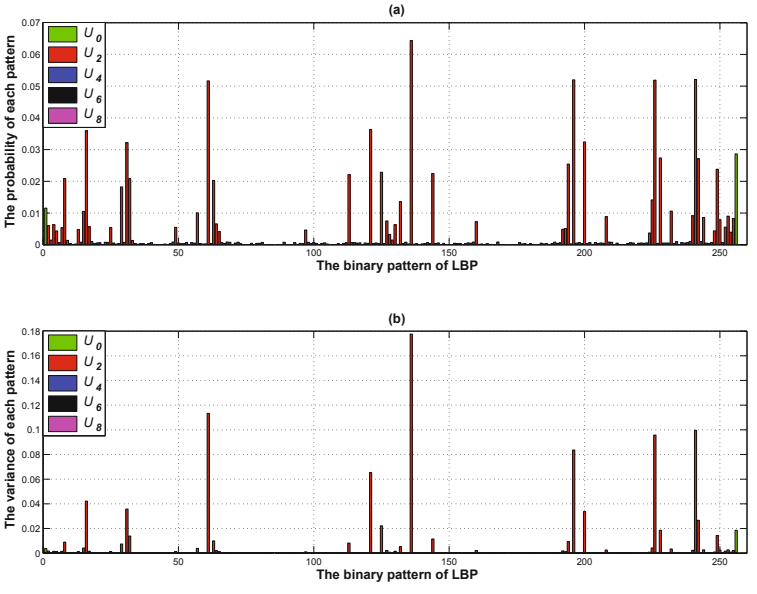


Fig. 7. (a) The scattergram of average probability of 256-*patterns* in LBP after histogram matching. **(b)** The variogram of average probability of 256-*patterns* in LBP after histogram matching.

difference between different faces can be reflected mainly by the patterns in U_0 and U_2. What is more, the fact about the larger difference of variance illustrates that some patterns in U_0 and U_2 are more sensitive to identify the details in face recognition.

4.2 The Influence to LBP Resulting from Illumination Changing

Owing to the fact that the shooting distance and the difference of shooting angle can cause the illumination change when we capture images. In this section, we analyze the performance of LBP operator under the changing non-monotone illumination circumstance.

The purpose of histogram equalization[13] is to reach homogeneous distribution in the gray image. Furthermore, the purpose of histogram matching[13] is to make use of gray transformation to get one image which contains similar gray distribution corresponding to the specified image. We can regard these two methods as gray normalization and use them to promote the image quality through avoiding the effect caused by illumination changing. $Set1\_Q$ and $Set2\_Q$ are NIR normalized image libraries after histogram equalization. $Set1\_M$ and $Set2\_M$ are NIR normalized image libraries after histogram matching. At the same time, $Set1$ and $Set2$ are two image libraries without gray normalization. Hence, we can

think that *Table* 1 contains three non-monotone illumination changes of varying degrees.

According to the process of *Experiment* 4.1, we come to the experimental results shown in Fig. 6 of $Set1\_Q$ firstly. Following is the experimental results about $Set1\_M$ shown in Fig. 7.

From the experiments mentioned above, we find that no change in LBP probability distribution has been caused by histogram equalization in facial images according to Fig. 6 with Fig. 7. Next, after comparing Fig. 5 with Fig. 6, the same condition comes to the histogram matching as well. At last, we also get the similar result when comparing Fig. 6 with Fig. 7. In sum, LBP has excellent robustness to the non-monotone illumination changes in the NIR facial image.

4.3 Experimental Verification

In the Section 4.1 and 4.2, we make a research on LBP characteristics in three large facial libraries including $Set1$, $Set1\_Q$ and $Set1\_M$. In this section, according to the experimental process in Section 4.1 and 4.2, we do the same experiments again in three smaller image libraries, $Set2$, $Set2\_Q$ and $Set2\_M$ which have the quarter samples of $Set1$, $Set1\_Q$ and $Set1\_M$. The experimental results show that they have exactly the same conclusions in $Set2$, $Set2\_Q$ and $Set2\_M$ as in $Set1$, $Set1\_Q$ and $Set1\_M$.

5 Conclusions

Due to the fact of lacking the study about infrared image, in recent years, more attention has been paid to the usage of near infrared image in face recognition. Simultaneously, researching the characteristic of LBP in that kind of images turns to be more meaningful. In the paper, after doing various experiments in NIR facial libraries designed by us, we come to some useful outcomes. Firstly, there are 58 kinds of patterns totally in U_0 and U_2, but they cover the image information with 93%. On the contrary, the total kinds of patterns in U_4, U_6 and U_8 are 198, but they cover the image information with only 7%. That is to say, it is more efficient to describe the texture image with these 58 kinds of patterns than others. Secondly, we find that the probability variance of U_0 and U_2 is fairly large, but of the rests is very smaller. This means that slight difference between different faces is reflected mainly by the patterns in U_0 and U_2. In addition, the difference of the variance is larger in U_0 and U_2, which suggests that the patterns in it have different contributions in describing the texture structure. Finally, we find that the operator $\mathrm{LBP}_{8,1}$ has excellent robustness to the non-monotone illumination change in the NIR facial image.

References

1. Ojala, T., Pietikainen, M., Harwood, D.: A comparative study of texture measures with classification based on featured distributions. Pattern Recognition 29, 51–59 (1996)

2. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 971–987 (2002)
3. Ojala, T., Valkealahti, K., Oja, E., Pietikainen, M.: Texture discrimination with multidimensional distributions of signed gray-level differences. Pattern Recognition 34, 727–739 (2001)
4. Qing, X., Jie, Y., Siyi, D.: Texture segmentation using LBP embedded region competition. Electronic Letters on Computer Vision and Image Analysis 5, 41–47 (2005)
5. Ma, L., Zhu, L.: Integration of the optimal Gabor filter design and local binary patterns for texture segmentation. In: IEEE International Conference on Integration Technology, ICIT 2007, pp. 408–413 (2007)
6. Jin, H., Liu, Q., Lu, H., Tong, X.: Face detection using improved LBP under bayesian framework. In: Proceedings of Third International Conference on Image and Graphics, 2004, pp. 306–309 (2004)
7. Ahonen, T., Hadid, A., Pietikäinen, M.: Face recognition with local binary patterns. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 469–481. Springer, Heidelberg (2004)
8. Ahonen, T., Hadid, A.: Face description with local binary patterns: application to face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 2037–2041 (2006)
9. Zhang, G., Huang, X., Li, S.Z., Wang, Y., Wu, X.: Boosting local binary pattern (LBP)-based face recognition. In: Li, S.Z., Lai, J.-H., Tan, T., Feng, G.-C., Wang, Y. (eds.) SINOBIOMETRICS 2004. LNCS, vol. 3338, pp. 179–186. Springer, Heidelberg (2004)
10. Li, S.Z., Jain, A.K.: NetLibrary, Inc. Handbook of Face Recognition. Springer, Heidelberg (2005)
11. Li, S.Z., Chu, R.F., Liao, S.C., Zhang, L.: Illumination invariant face recognition using near-infrared images. IEEE Transactions on Pattern Analysis and Machine Intelligence 29, 627–639 (2007)
12. Li, S.Z., Chu, R.F., Ao, M., Zhang, L., He, R.: Highly accurate and fast face recognition using near infrared images. In: Zhang, D., Jain, A.K. (eds.) ICB 2005. LNCS, vol. 3832, pp. 151–158. Springer, Heidelberg (2005)
13. Diniz, P.S.R.: Digital Image Processing. Cambridge University Press, Cambridge (2010)

Fault Diagnosis for Smart Grid with Uncertainty Information Based on Data*

Qiuye Sun[1,2], Zhongxu Li[1], Jianguo Zhou[1], and Xue Liang[1]

[1] School of Information Science and Engineering, Northeastern University, Shenyang, 110819, P.R. China
[2] Rongxin Power Electronic Co., Ltd, Anshan, 114051, P.R. China
sunqiuye@mail.neu.edu.cn

Abstract. The concept of Smart Grid has gained significant acceptance during the last several years due to the high cost of energy, environment concerns, and major advances in distributed generation (DG) technologies. Distribution systems have traditionally been designed as radial systems, and time coordination of protection devices at the distribution level, the main characteristic for fault diagnosis, is a standard practice used by the utilities. However, when Smart Grid occurs fault, the certainty and integrity of information will be damaged by many causations. In order to improve the accuracy and rapidity of fault diagnosis, it is necessary to discover a new method that has high fault tolerant and can compress data space and filtrate error data. To deal with the uncertainty and deferent structures of the causation, rough sets and intuitionistic fuzzy sets are introduced. Based on them, intuitionistic uncertainty-rough sets are proposed and the reduction algorithm is improved. The rule reliability is deduced using intuitionistic fuzzy sets and probability. The worked example for Xigaze power system in China's Tibet shows the effectiveness and usefulness of the approach.

Keywords: Smart Grid, Fault diagnosis, Rough sets, Intuitionistic uncertainty sets.

1 Introduction

The concept of "IntelliGrid" [1][2] and "Smart Grid" [3]are put forward by the power research organizations in Europe and America, respectively. In recent years, Smart Grid has emerged in China due to its obvious advantages in reliable and economic operation, energy-saving and environment protection. Until now there has been no unified definition of Smart Grid, but some characteristics of Smart Grid are identified home and abroad[1]-[4]: ①Self-healing: Online self-assessment of the grid operation state, able to detect fault quickly without or with little manual intervention to avoid large area blackout. ② Interaction: able to incorporate consumer equipment and behavior in the design and operation of the grid. ③Optimization: able to optimize its

* Projects(60904101, 60972164) supported by the National Natural Science Foundation of China; Project(N090404009) supported by the Fundamental Research Funds for the Central Universities; and Project(20090461187)supported by China Postdoctoral Science Foundation.

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 66–75, 2011.

capital assets while minimizing operation and maintenance costs in the whole life cycle, and reduce the loss of power grid. ④Compatibility: able to accommodate a wide variety of distributed generation and storage options. ⑤Integration: assistant decision-making system based on the integration of information. It can be seen from above that safety and sharing are regarded as the core advantage and key technical problem to be tackled of Smart Grid. However, precise and real-time fault diagnosis has become one of the key technologies in the application of Smart Grid due to its impact on the power grid safety[5].

The networking mode of Smart Grid is similar to that of MicroGrid[6] which is quite different from traditional distribution network. Because of the connection of DG (Distributed Generation), there are many power sources in the system, so the power flow is not unidirectional and the fault point may be not at the point with lowest voltage. Traditional fault diagnosis methods are no longer suitable and IEEE has made guide standard for twice[7] [8]. The factors that influence the accuracy and real-time performance of fault diagnosis are as follows: ①Currently, fault diagnosis mainly depends on the information of the circuit breaker and relay protection. But maloperation or miss trip sometimes occur on these devices ②Traditional discrete methods are harsh to the boundary of the continuous values of voltage and current. It cannot adapt to the diversity of fault information after the connection of DG. ③Some faults are not included in the expert database and there's no inversion record, which makes it hard for operators to handle the fault in time. ④There's no consideration of the randomness of the fault. However, large amount of information is not fully utilized in the fault diagnosis of Smart Grid, like backup protection, fault voltage, current and waveform. If these information is effectively used, the precision of fault diagnosis will be largely improved.

Rough sets are applied for the fault diagnosis of Smart Grid in this paper, considering their ability of reduction to mass data and good performance in the fault diagnosis of distribution network[9]. Based on the theory of fuzzy-rough sets[10] and intuitionistic fuzzy sets[11], and also considering the random fault factors and actual situation of Smart Grid, the intuitionistic uncertainty sets are established to perform the fault diagnosis. The worked example for Xigaze power system in China's Tibet shows the effectiveness and usefulness of the approach.

2 Mechanism Analysis of the Impact of Distributed Generators on Fault Diagnosis

2.1 Mechanism Analysis of Fault Diagnosis with Distributed Generators

If a DG is connected to a feeder with traditional relay protection, the relay protection should be installed after DG, as shown in Fig.1. When fault occurs in the feeder with DG, the DG will send short circuit current to the fault point, which reduces the sensitivity of the relay protection.

In Fig.1, Z_s is the equivalent impedance between the source and the location of the relay protection; Z_l is the impedance of feeder; Z_{DG} is the impedance of DG and transformers; l is the distance between the short circuit point and the end of feeder; x

is the distance between the short circuit point and DG. In order to protect the whole feeder, current instantaneous trip protection with time limit should provide enough sensitivity when phase to phase short circuit fault occurs at the end of feeder under minimum operation mode. Generally, sensitivity coefficient is used to check the sensitivity.

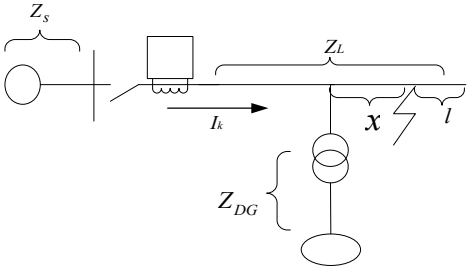


Fig. 1. Impact of distributed generation on the protection

Suppose $Z_{DG} = \alpha_1 Z_S$, $Z_l = \alpha_2 Z_S$, the setting value of instantaneous trip protection I_{op1} is set under the situation that phase to phase short circuit occurs at point k at the end of feeder. Reliable coefficient is K_k .The setting value of over current protection I_{op2} is set according to maximum load current. According to the relation of phase to phase and three-phase short circuit, the rate of the setting value of instantaneous trip protection is 0.5. Feeder sensitivity coefficient is K_{sen} . When phase to phase short circuit occurs on feeder without DG, the detected fault current by instantaneous trip protection device is I_{K1} ; when with DG for the same situation, the detected fault current is I_{K2} . So

$$
\begin{aligned}
I_{op1} &= \frac{I_{k\min}^{(2)}}{K_{\min}} = \frac{\sqrt{3}}{2(1+\alpha_2)ZsK_{sen}} \\
I_{op2} &= \frac{I_{op1}}{2} \\
I_{K1} &= \frac{\sqrt{3}}{2\{1+\alpha 2(1-l)\}Zs} \\
I_{K2} &= \frac{\alpha 1}{1+\alpha 1+(1-x-l)\alpha 2} \times \frac{\sqrt{3}}{2\left\{x\alpha 2+\frac{\alpha 1[1+\alpha 2(1-x-l)]}{1+\alpha 1+\alpha 2(1-x-l)}\right\}ZS}
\end{aligned}
\tag{1}
$$

It can be seen that traditional fault diagnosis methods will cause fault report and missing report in the Smart Grid with DG, which attracts lots of research interests. A fault handling strategy based on voltage control was proposed in [12] In [13], a method combining voltage and frequency control was used to detect isolated islands. In [14] history records were used to reason based on causality. Then a set of plans was obtained to detect and handle isolated islands. Based on different DG models, the impact of DG on feeder voltage and frequency under fault was studied in [15]. In most researches

nowadays, the continuous signals like voltage, current, and frequency were considered as important fault diagnosis properties, which was rare in traditional fault diagnosis for distribution network. However, methods to handle incomplete information or fuzzy, random cases are not solved in the papers above, which is the key problem of fault diagnosis application.

2.2 Theory of Rough Sets

Rough set [16] introduced by Pawlak in 1982 is a veryimportant mathematical tool to deal with vagueness and uncertainty.

Let U denote a finite and nonempty set called the universe. Suppose $R \subseteq U \times U$ is an equivalence relation on, i.e., R is reflexive, symmetric, and transitive. The equivalence relation partitions the set U into disjoint subsets. Elements in the same equivalence classes of R are said to be indistinguishable. Equivalence classes of are called elementary sets. Every union of elementary sets is called a definable set. The empty set is considered to be a definable set, thus all the definable sets form a Boolean algebra. (U, R) is called an approximation space. Given an arbitrary set $X \subseteq U$, one can characterize X by a pair of lower and upper approximations. The lower approximation $\underline{apr}_R X$ is the greatest definable set contained in X , and the upper approximation $\overline{apr}_R X$ is the least definable set containing X . They can be computed by two equivalent formulas

$$\begin{aligned} &\underline{apr}_R X = \{x : [x]_R \subseteq X\} \\ &\overline{apr}_R X = \{x : [x]_R \cap X \neq \varnothing\} \\ &\underline{apr}_R X = \bigcup\{[x]_R : [x]_R \subseteq X\} \\ &\overline{apr}_R X = \bigcup\{[x]_R : [x]_R \cap X \neq \varnothing\} \end{aligned} \tag{2}$$

The lower approximation $\underline{apr}_R X$ and upper approximation $\overline{apr}_R X$ satisfy the following properties:

1) $\underline{apr}_R U = U$, $\overline{apr}_R \varnothing = \varnothing$
2) $\underline{apr}_R (X \cap Y) = \underline{apr}_R X \cap \underline{apr}_R Y$, $\overline{apr}_R (X \cup Y) = \overline{apr}_R X \cup \overline{apr}_R Y$
3) $\underline{apr}_R X^c = (\overline{apr}_R X)^c$, $\overline{apr}_R X^c = (\underline{apr}_R X)^c$
4) $\underline{apr}_R X \subseteq X$, $X \subseteq \overline{apr}_R X$
5) $X \subseteq \underline{apr}_R (\overline{apr}_R X)$, $\overline{apr}_R (\underline{apr}_R X) \subseteq X$
6) $\underline{apr}_R X = \underline{apr}_R (\underline{apr}_R X)$, $\overline{apr}_R X = \overline{apr}_R (\overline{apr}_R X)$

From these six properties, one can obtain many properties of rough sets, we only list these six properties because they can be treated as axiomatic characteristics of rough sets.

Based on the definition of intuitionistic fuzzy set sintuitionistic uncertainty rough sets is described as follows: universe $U=\{x_i \mid i=1,\cdots,n\}$ is described by the discretization attributes $\{P_1,P_2,\cdots,P_p\}$. Each attribute measures some important feature of and is limited to linguistic terms $A(P_i)=\{F_{ik} \mid k=1,\cdots,C_i\}$. Each object $x_i \in U$ is classified by a set of classes $A(Q)=\{F_l \mid l=1,\cdots,C_Q\}$.Each $F_l \in A(Q)$ may be a crisp or membership function and Q is decision attribute. The set $U/P=\{F_{ik} \mid i=1,\cdots,p;k=1,\cdots,C_i\}$ can be regarded as a kind of partitions of U by a set of attributes P using uncertainty model.

The lower and upper approximation membership function and nonmembership function are defined as follows:

$$\mu_{\underline{A}}(F_{ik})=\begin{cases}\inf\limits_{x\in U}\left\{\max\left[1-\mu_{F_{ik}}(x),\mu_A(x),\alpha\right]\right\} \\ \text{where } D_{\underline{A}}(F_{ik})\neq\varnothing \\ 1 \\ \text{where } D_{\underline{A}}(F_{ik})=\varnothing\end{cases} \tag{3}$$

$$\chi_{\underline{A}}(F_{ik})=\begin{cases}\sup\limits_{x\in U}\left\{\min\left[1-\chi_{F_{ik}}(x),\chi_A(x),\alpha\right]\right\} \\ \text{where } B_{\underline{A}}(F_{ik})\neq\varnothing \\ 1 \\ \text{where } B_{\underline{A}}(F_{ik})=\varnothing\end{cases} \tag{4}$$

$$\mu_{\overline{A}}(F_{ik})=\begin{cases}\sup\limits_{x\in U}\left\{\max\left\{\min\left[\mu_{F_{ik}}(x),\mu_A(x)\right],\beta\right\}\right\} \\ \text{where } D_{\overline{A}}(F_{ik})\neq\varnothing \\ 1 \\ \text{where } D_{\overline{A}}(F_{ik})=\varnothing\end{cases} \tag{5}$$

$$\chi_{\overline{A}}(F_{ik})=\begin{cases}\sup\limits_{x\in U}\left\{\max\left\{\min\left[\chi_{F_{ik}}(x),\chi_A(x)\right],\beta\right\}\right\} \\ \text{where } B_{\overline{A}}(F_{ik})\neq\varnothing \\ 1 \\ \text{where } B_{\overline{A}}(F_{ik})=\varnothing\end{cases} \tag{6}$$

Where $0\le\beta<\alpha\le 1$ are lower and upper limits in probability.

3 Establishment of the Fault Diagnosis System for Smart Grid

As to the fault diagnosis for Smart Grid, we discuss in two aspects: discrete data and continuous data. Discrete data mainly consists of breakers and relay protection devices. Continuous data mainly consists of the current, voltage, frequency, and power factor. Discrete property can be handled by rough sets so we only need to discuss the reliability of devices.

The fault rate of breakers λ_{Qi} can be calculated by (1)

$$\lambda_{Qi} = \lambda_Q + \lambda_L \frac{L}{100} + \lambda \tag{7}$$

where λ_Q is its own fault rate, λ_L is the impact rate of feeders, λ is the impact rate of buses, L is the length of feeder. λ_Q is only relative to itself and can been considered as random perturbation. λ_L and λ are relative to the distance between the device and bus, DG, load, and fault point. They can be given by expert experience. The reliability of relay protection devices is lower than switchgear, so we set the parameters λ_R in λ_{MR}, λ_{SR}, and λ_{FR} larger than the parameters of breakers by 0.3%、0.2%、0.1%. Intuitionistic uncertainty membership and nonmembership functions of discrete property are as follows:

$$\mu_{Di} = \mu_i\left(\lambda_L, \lambda\right) + \sigma_i\left(\lambda_Q, \lambda_R\right) \tag{8}$$

$$\chi_{Di} = \chi_i\left(\lambda_L, \lambda\right) - \sigma_i\left(\lambda_Q, \lambda_R\right) \tag{9}$$

In continuous data, the current error is mainly caused by CT and communication interference. The later is a random event and can be expressed by a random disturbance; the former is caused by the error of CT, which is defined as follows:

$$\Delta I\% = (KI_2 - I_1)/I_1 \times 100\% = \varepsilon\% + \lambda_{i1}\Phi + \lambda_{i2} I_d \tag{10}$$

where Φ is flux of CT, I_d is the short circuit current, $K = I_{1N}/I_{2N}$ is the transformation ratio, I_1 and I_2 are measured values on the primary side and secondary side of CT, λ_{i1} and λ_{i2} are the coefficient of flux and short circuit current, respectively. $\varepsilon\%$ is the error level of CT defined as follows:

$$\varepsilon\% = \frac{100}{I_1} \times \sqrt{\frac{1}{T}\int_0^T (K_I i_2 - i_1)^2 \mathrm{d}t} \tag{11}$$

where i_1 and i_2 are the short circuit current of the primary side and secondary side of CT. T is the period of short circuit current. Definitions of measurement errors of voltage, frequency, and power factor are similar.

In fault cases, short circuit relation of Smart Grid is described as follows:

$$\begin{bmatrix} \dot{V}_a \\ \dot{V}_b \\ \dot{V}_c \end{bmatrix} = l \begin{bmatrix} \dot{Z}_{aa} & \dot{Z}_{ab} & \dot{Z}_{ac} \\ \dot{Z}_{ba} & \dot{Z}_{bb} & \dot{Z}_{bc} \\ \dot{Z}_{ca} & \dot{Z}_{cb} & \dot{Z}_{cc} \end{bmatrix} \begin{bmatrix} \dot{I}_a \\ \dot{I}_b \\ \dot{I}_c \end{bmatrix} + R \begin{bmatrix} \dot{I}_{fa} \\ \dot{I}_{fb} \\ \dot{I}_{fc} \end{bmatrix} \tag{12}$$

where $\dot{V}_a$ is the voltage of phase A, $\dot{I}_a$ is the rated current of phase A; $\dot{I}_{fa}$ is the short circuit current of phase A; $\dot{I}_{La}$ is the current of phase A in normal operation. l is the distance between the fault point and measurement point. R is the resistance of the fault point to the ground. $\dot{Z}$ is the line impedance matrix. There are two unknown variables in equation (12): l and $\dot{R}$. Equation of voltage and current including uncertainty is shown in (13):

$$\dot{V}_a = (l\dot{Z} + R)\dot{I}_a - R\dot{I}_{La} \tag{13}$$

where $\dot{V}_a$, $\dot{I}_a$, $\dot{I}_{La}$, and $\dot{Z}$ is known and l and $\dot{R}$ are random variable. Because $\dot{I}_a \gg \dot{I}_{La}$, $R\dot{I}_{La}$ equals to σ_i in (10), so $(l\dot{Z} + R)\dot{I}_a$ can be seen as a combination of a certain variable and a fuzzy variable. In fault cases, membership and nonmembership functions of the fault point in Smart Grid are expressed as follows:

$$\mu_{Ci} = \mu_i(\Delta I\%) + \mu_i(\dot{I}_a) + \sigma_i(T) + \sigma_i(\dot{I}_{La}) \tag{14}$$

$$\chi_{Ci} = \chi_i(\Delta I\%) + \chi_i(\dot{I}_a) - \sigma_i(T) - \sigma_i(\dot{I}_{La}) \tag{15}$$

where $\mu_i(\Delta I\%)$ and $\mu_i(\dot{I}_a)$ is the current measurement error by experts and current estimation of the fault point to ground, respectively. $\sigma_i(T)$ and $\sigma_i(\dot{I}_{La})$ are random mutation and random charging current of the fault point to ground, respectively.

Now we discuss the reduction algorithm based on the diagnosis model of Smart Grid above. According to the definition of dependency degree and non dependency degree: when $P \Rightarrow Q$,the positive region of intuitionistic uncertainty set U/Q covers elements $\gamma_P(Q) \times 100\%$ in knowledge base $K = \{U, P\}$, and impossible to cover elements $\kappa_P(Q) \times 100\%$ in knowledge base $K = \{U, P\}$. So coefficient $\gamma_P(Q)$ and $\kappa_P(Q)$ can be considered as dependency relation and non dependency relation between P and Q .

4 Trial Operation Analysis of Fault Diagnosis on Xigaze Power System in China's Tibet

There has been no unified power grid in Tibet. A power grid forms among Lhasa, Rikaze, Shannan, and Naqu in the central Tibet. Power grids in Linzi and Changdu are isolated. There's no power grid in Ali.

Until the end of 2004, installed capacity of city-level power grid in Tibet has reached 429.8MW, in which hydropower accounts for 179.49MW, pumped storage power station accounts for 112.5 MW, geothermal power plant accounts for 24.18 MW, thermal power accounts for 11.78MW, and new energy accounts for 101.85 MW. The percentage of the power sources above is 44.7%, 24.3%, 5.4%, 3.6%, and 22%, respectively. In this paper, Lazi distribution system is researched. The grid structure is shown in Fig.2 with voltage level 10KV.

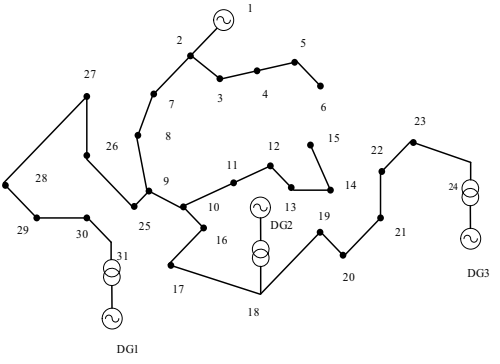


Fig. 2. Structure of Lazi distribution system

Maximum load of each line node of Lazi distribution system is shown in Table.1, which are important references for experts to determine the fault electric quantity.

Table 1. Max load parameters of line node (KW, KVar)

Node	Real load	Reactive load	Node	Real load	Reactive load
1	0	0	18	300	150
2	90	30	19	90	60
3	100	70	20	150	100
4	70	50	21	60	40
5	120	90	22	150	100
6	120	80	23	150	100
7	95	70	24	200	100
8	150	100	25	130	70
9	130	80	26	80	60
10	100	30	27	130	80
11	70	40	28	120	80
12	100	70	29	90	60
13	90	60	30	200	100
14	95	70	31	200	100
15	100	70	DG1	369.96	225.60
16	130	100	DG2	92.15	65.36
17	150	120	DG3	200.20	160.48

Generally, more than 90% of the faults in power system are single fault, double fault, and single fault with device's abnormal action. Based on the three cases above, the fault diagnosis rule base is established.

In the distribution network in Fig.4, there're 30 branches in total and some branches install breakers and current instantaneous trip protection bi-directionally. Corresponding binary attributes reaches 84. There're 28 branches except the branches connected with nodes in ends of line, so the corresponding distance protection is 28. There's 1 bus main protection in mail power source (Node 1). One under frequency load shedding and one under voltage load shedding protection are installed on each bus with DG. So there are 118 binary attributes in the fault diagnosis system. Measurable voltage and current in the branches reach 68. The frequency and power factor of the mail power source and DGs are known. So there are 76 continuous attributes. Considering the three cases: single fault, double fault, and single fault with abnormal action devices, there are 3 decision attribute totally. The following 318 records are included in the expert database: 68 records of single fault without device's abnormal action, 200 records of double fault, and 50 records of single fault with device's abnormal action. With additional 32 records of non fault operation, up to 350 records build the original rule base. As to the generated rule base, 30 records of single fault without switches and with abnormal action (Fault Type Ⅰ), 40 records of double fault without switches and with abnormal action (Fault Type Ⅱ), and 30 records of single fault with switches and with abnormal action (Fault Type Ⅲ) are selected as inspection data of the knowledge base. The experiment result is shown in Tab.2.

Table 2. Comparison of different system experiment data

Fault Type	Amount of records	Correct records of Fault 1		Correct records of Fault 2		Correct rate	
		Rules before reduction	Rules after reduction	Rules before reduction	Rules after reduction	Rules before reduction	Rules after reduction
Type Ⅰ	30	30	30			100%	100%
Type Ⅱ	40	39	40	6	37	15%	87.5%
Type Ⅲ	30	13	28			43.3%	93.3%

It can be seen from Tab.2 that recognition rate for fault type Ⅰ is 100%. For fault type Ⅱ, at least one fault can be recognized by system. In the tested 40 records, only 3 records cannot recognize the fault correctly because of the incomplete information of original records. For system without reduction, only 6 records can recognize the two faults. For fault type Ⅲ, the anti-interference ability of reduced rules is largely improved due to the decrease of redundancy attributes. The two unrecognized records is because of the missing information of key attributes. For original records without reduction, only 13 records can be recognized correctly.

5 Conclusions

A reduction algorithm based on intuitionistic uncertainty rough sets is proposed in this paper according to the special need of fault diagnosis in Smart Grid. The main problems which interfere the accuracy of fault diagnosis are effectively solved by the proposed algorithm. For handling the uncertain information, the proposed algorithm has better performance than the present fault diagnosis methods of power system. The worked example for Xigaze power system in China's Tibet shows the effectiveness of the algorithm for fault diagnosis of the practical and complex Smart Grid, which largely extends the application scope of Smart Grid.

References

1. Power delivery system and electricity markets of the future. J. EPRI Journal 2, 31–36 (2003)
2. Haase, P.: IntelliGrid: a smart network of power. J. EPRI Journal, 17–25 (Summer 2005)
3. Momoh, J.A.: Smart grid design for efficient and flexible power networks operation and control. In: IEEE/PES Power Systems Conference and Exposition, pp. 1–8. IEEE Press, Los Alamitos (2009)
4. Yu, Y., Luan, W.: Smart Grid. J. Power and Electrical Engineering 2(5), 13–17 (2008)
5. Walling, R.A., Saint, R., Dugan, R.C., Burke, J., Kojovic, L.A.: Summary of Distributed Resources Impact on Power Delivery Systems. J. IEEE Transactions on Power Delivery 23(3), 1636–1644 (2008)
6. Brown, R.E.: Impact of Smart Grid on Distribution System Design. In: IEEE Power and Energy Society General Meeting, pp. 1–4. IEEE Press, Pittsburgh (2008)
7. IEEE Std. 929-2000. IEEE recommended practice for utility interface of photovoltaic (PV) system. IEEE P, Iscataway, NJ, USA (2000)
8. IEEE Std. 1547. IEEE standard for interconnecting distributed resources with electric power systems. IEEE P, Iscataway, NJ, USA (2003)
9. Sun, Q., Zhang, H., Dai, J.: On-line Fault Diagnose of Distribution System Based on Modified Rough Sets Reduction Algorithm. J. Proceedings of the CSEE 27(7), 58–64 (2007)
10. Jensen, R., Shen, Q.: Fuzzy-rough attribute reduction with application to web categorization. J. Fuzzy Sets and System 141(3), 469–485 (2004)
11. Deschrijver, G., Cornelis, C., Kerre, E.E.: On the representation of intuitionistic fuzzy t-norms and t-conorms. J. IEEE Transactions on Fuzzy Systems 12(1), 45–61 (2004)
12. Fang, G., Iravani, M.R.: A Control Strategy for a Distributed Generation Unit in Grid-Connected and Autonomous Modes of Operation. J. IEEE Transactions on Power Delivery 23(2), 850–859 (2008)
13. Menon, V., Nehrir, M.H.: A Hybrid Islanding Detection Technique Using Voltage Unbalance and Frequency Set Point. J. IEEE Transactions on Power Systems 22(1), 442–448 (2007)
14. Jayaweera, D., Galloway, S., Burt, G., McDonald, J.R.: A Sampling Approach for Intentional Islanding of Distributed Generation. J. IEEE Transactions on Power Systems 22(2), 514–521 (2007)
15. Quinonez-Varela, G., Cruden, A.: Development of a Small-Scale Generator Set Model for Local Network Voltage and Frequency Stability Analysis. J. IEEE Transaction on Energy Conversion 22(2), 368–375 (2007)
16. Zeshui, X.: Intuitionistic Fuzzy Aggregation Operators. J. IEEE Transactions on Fuzzy Systems 15(6), 1179–1187 (2007)

Sparse Kernel Regression for Traffic Flow Forecasting

Rongqing Huang, Shiliang Sun, and Yan Liu

Department of Computer Science and Technology, East China Normal University
500 Dongchuan Road, Shanghai 200241, P.R. China
rqhuang09@gmail.com, slsun@cs.ecnu.edu.cn, yliu@cc.ecnu.edu.cn

Abstract. In this paper, a new kernel regression algorithm with sparse distance metric is proposed and applied to the traffic flow forecasting. The sparse kernel regression model is established by enforcing a mixed $(2, 1)$-norm regularization over the metric matrix. It learns a mahalanobis metric by a gradient descent procedure, which can simultaneously remove noise in data and lead to a low-rank metric matrix. The new model is applied to forecast short-term traffic flows to verify its effectiveness. Experiments on real data of urban vehicular traffic flows are performed. Comparisons with two related kernel regression algorithms under three criterions show that the proposed algorithm is more effective for short-term traffic flow forecasting.

Keywords: Traffic flow forecasting, Kernel regression, Sparse distance metric learning, Mixed norm regularization, Gradient descent algorithm.

1 Introduction

Short-term traffic flow forecasting is one of the most important and fundamental problems in intelligent transportation systems (ITS). It contributes a lot to traffic signal control and congestion avoidance. The benefits of ITS cannot be realized without the ability to forecast traffic condition in the next time interval, for example, 5 minutes to half an hour. A good traffic condition forecasting model will provide this ability and make traffic management more efficient [1]. To alleviate the increasingly serious urban traffic condition, traffic flow forecasting, which is an important aspect of traffic condition, has already evoked great interest of the researchers in recent years.

Up to the present, there are a variety of methods proposed for short-term traffic flow forecasting such as Markov chain models [1], time series models [2], Kalman filter theory [3], Bayesian networks [4], and support vector machines [5]. All these methods are based on the fact that historical data, especially the current and most recent data, can provide information for predicting the future data. In addition, kernel regression [6,7], which is also a classical and important method for traffic flow forecasting, is our concern in this paper.

The traditional kernel regression (KR) combines Euclidean distance metrics with the Gaussian kernel, which decay exponentially with squared distance

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 76–84, 2011.

rescaled by a kernel width factor. KR is the simplest kernel regression method for not needing to learn a metric matrix. However, Euclidean distance metric has its obvious defect of treating all the features the same. It neglects the fact that features of an input vector may play different roles in a specific task and should be assigned different weights.

Recently, it has been shown that even a simple linear transformation of the input features can lead to significant improvements for machine learning algorithms involving distance metric learning. Metric learning for kernel regression (MLKR) is the first outcome to combine kernel regression and distance metric learning [9]. It learns a Mahalanobis metric by a gradient descent procedure to minimize the training error. The application of MLKR in several domains for regression also shows promising results. However, the observed data, especially high-dimensional datasets, are probably noisy [11]. If the data can be preprocessed to remove the irrelevant features, the efficiency and effectiveness of existing kernel regression algorithms are likely to increase largely. Based on this rationality, we propose a sparse kernel regression algorithm (SMLKR) in this paper and apply it to forecast short-term traffic flows. Experiments of SMLKR on real traffic flow datasets show that it can learn a good metric and simultaneously conduct dimensionality reduction as well.

The rest of this paper is organized as follows. Section 2 thoroughly introduces the sparse kernel regression model. Section 3 reports our experimental results on real data of urban vehicular traffic flows, including comparisons with other two related methods. Finally, Section 4 concludes this paper and gives future research direction.

2 The Proposed Sparse Kernel Regression Model

2.1 Basic Notations

Let $(\boldsymbol{x}, y)$ represent an example with input $\boldsymbol{x} = (x_1, x_2, \ldots, x_d) \in R^d$ and $y \in R$ be the corresponding target value. A dataset with n examples is denoted by $Z = \{(\boldsymbol{x}_i, y_i)_{i=1}^n\}$. The space of symmetric d by d matrices is denoted by S^d. If $S \in S^d$ is positive semi-definite, we write it as $S \geq 0$. The cone of positive semi-definite matrices is denoted by S_+^d and we denote the set of d by d orthonormal matrices by O^d. The trace operation for matrices is denoted by $Tr(\cdot)$, which is the sum of all the diagonal elements of a matrix [8]. In addition, any d by d diagonal matrix is denoted by $diag(D_{11}, D_{22}, \ldots, D_{dd})$, where $D_{11}, D_{22}, \ldots, D_{dd}$ are the diagonal elements of the matrix.

In mathematics, the Euclidean distance is the "ordinary" distance between two points that one would measure with a ruler and is given by the following formula:

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\| = \sqrt{\sum_{r=1}^{d} (x_{ir} - x_{jr})^2}, \tag{1}$$

where $\boldsymbol{x}_i, \boldsymbol{x}_j \in R^d$ and $i, j \in \{1, 2, \ldots, n\}$. The associated norm is called the Euclidean norm, which is represented as $\|\cdot\|$ [13]. For simplification, we denote the difference vector of two vectors by $\boldsymbol{x}_{ij} = \boldsymbol{x}_i - \boldsymbol{x}_j$.

2.2 Kernel Regression

Given a training set of (possibly noisy) examples $\{(\boldsymbol{x}_i, y_i)_{i=1}^n\}$, the standard regression task is to estimate an unknown function $f : R^d \to R$, so that $y_i = f(\boldsymbol{x}_i) + \varepsilon$, with $\hat{y}_i \approx f(\boldsymbol{x}_i)$, the estimation of y_i, at the minimum loss:

$$L = \sum_i (y_i - \hat{y}_i)^2, \tag{2}$$

In short-term traffic flow forecasting model, generally the current traffic flow is closely related to flows of past time. The previous traffic flows contribute differently to the prediction of the current flow. Therefore, the corresponding weights of past flows are different from each other. With the assumption that the relationship between past flows and the predicted flow is linear, then the current flow on a certain spot can be forecasted using its previous flows as [7]:

$$\hat{y}_i = a_1.y_{i-1} + a_2.y_{i-2} + \ldots + a_m.y_{i-m}, \tag{3}$$

where $\hat{y}_i$ is the predicted flow and $y_{i-1}, y_{i-2}, \ldots, y_{i-m}$ are m flows of past time. Actually, the estimation of y_i is usually obtained by the weighted average method. If the weights are determined by a kernel function, then (3) can be reformulated in terms of kernel:

$$\hat{y}_i = \frac{\sum_{j=i-m}^{i-1} y_j.k_{ij}}{\sum_{j=i-m}^{i-1} k_{ij}}, \tag{4}$$

where k_{ij} is a kernel function based on the squared distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. In traditional kernel regression, Gaussian kernel is usually adopted and its concrete formulation is as follows:

$$k_{ij} = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{d(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\sigma^2}}, \tag{5}$$

where $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the squared distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. The smaller the distance, the larger k_{ij} is, which means $\boldsymbol{x}_j$ is more similar to $\boldsymbol{x}_i$.

2.3 Sparse Distance Metric Learning

Given the transformation matrix $A \in R^{d \times d}$, any input vector $\boldsymbol{x}_i$ can get its corresponding transformation vector $\hat{\boldsymbol{x}}_i$, which is obtained by calculating $\hat{\boldsymbol{x}}_i = A\boldsymbol{x}_i$. In the input space, the Mahalanobis distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is defined as:

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i - \boldsymbol{x}_j)^T M (\boldsymbol{x}_i - \boldsymbol{x}_j), \tag{6}$$

where metric matrix M can be any symmetric positive semi-definite real matrix [9]. Setting M to be the identity matrix can recover the standard

Euclidean metric. Mathematically, M can be decomposed as $M = A^T A$. Therefore, the Mahalanobis distance in the input space is equivalent to the Euclidean distance in the transformed space. That is

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i - \boldsymbol{x}_j)^T A^T A(\boldsymbol{x}_i - \boldsymbol{x}_j) = \|A(\boldsymbol{x}_i - \boldsymbol{x}_j)\|^2. \tag{7}$$

As the observed data are probably contaminated by noise, ideally the transformation vector $\hat{\boldsymbol{x}}_i$ should have fewer dimensions than its corresponding input vector $\boldsymbol{x}_i$ as a consequence. Let A_i denote the i-th row vector of A, if $\|A_i\|$=0, then the i-th entry of $\hat{\boldsymbol{x}}_i$ becomes 0. That is, $\|A_i\|$=0 has the effect of feature selection. Therefore, to obtain a sparse transformation vector $\hat{\boldsymbol{x}}_i$, we can enforce a L_1-norm regularization across the vector $(\|A_1\|, \|A_2\|, \ldots, \|A_d\|)$, i.e., $\sum_{i=1}^{d} \|A_i\|$. Therefore, the sparse representation can be realized by a mixed $(2,1)$-norm regularization over transformation matrix A.

We aim at learning a Mahalinobis metirc as well as a low-rank metric matrix M. Let $M = A^T A = (M_1, M_2, \ldots, M_d)$, where M_i is the i-th row vector of M. It is obvious that $\|M_i\| = 0$ is equivalent to $\|A_i\| = 0$. Therefore, instead of enforcing L_1-regularization over vector $(\|A_1\|, \|A_2\|, \ldots, \|A_d\|)$, we can enforce L_1-norm regularization across the vector $(\|M_1\|, \|M_2\|, \ldots, \|M_d\|)$ to get the sparse solution [11]. The $(2,1)$-norm regularization over M is denoted by $\|M\|_{(2,1)} = \sum_{i=1}^{d} \|M_i\|$. A similar mixed $(2,1)$-norm regularization was used for multi-task learning and multi-class classification to learn the sparse representation shared across different tasks or classes [12].

2.4 Sparse Metric Learning for Kernel Regression

The objective of kernel regression is to make the accumulated quadratic leave-one-out regression error $L = \sum_i (y_i - \hat{y}_i)^2$ as small as possible. To learn a sparse distance metric for kernel regression, we enforce the mixed $(2,1)$-norm regularization over the metric matrix M. Therefore, the objective function of our proposed SMLKR algorithm can be formulated as follows

$$L(M) = \sum_i (y_i - \hat{y}_i)^2 + \mu \|M\|_{(2,1)}, \tag{8}$$

where μ is a positive step-size constant and will be fixed by cross-validation. With reference to [11],

$$\min \|M\|_{(2,1)} = Tr(M). \tag{9}$$

Therefore, the objective function of SMLKR can be reformulated as

$$L(M) = \sum_i (y_i - \hat{y}_i)^2 + \mu Tr(M). \tag{10}$$

$L(M)$ is differentiable over M. Consulting [7] and making use of the fact that $\frac{\partial Tr(M)}{\partial M} = I$, the gradient of (10) with respect to M can be stated as

$$\frac{\partial L(M)}{\partial M} = 2\sum_{i}(\hat{y}_i - y_i)\frac{\sum_{j=i-m}^{i-1}(\hat{y}_i - y_j)k_{ij}\boldsymbol{x}_{ij}\boldsymbol{x}_{ij}^T}{\sum_{j=i-m}^{i-1}k_{ij}} + \mu I. \tag{11}$$

After setting initial value of M, we adjust its subsequent values by a gradient descent procedure. Let G^t denote the gradient of the objective function at the t-th iteration, then the metric matrix M can be updated by

$$M_{(t)} = M_{(t-1)} - \alpha G^t, \tag{12}$$

where α is a small positive step-size constant. To keep M positive semi-definite, it is projected to the cone of positive semi-definite matrices by the eigen-decomposition of matrix $M_{(t)}$. That is, $M_{(t)} = P\Lambda_+ P^T$, where P is the eigen-vector matrix, and $\Lambda_+ = diag(max\{0, \lambda_1\}, max\{0, \lambda_2\}, \ldots, max\{0, \lambda_d\})$ with λ_i being the eigen-value of $M_{(t)}$.

Gradient descent steps for sparse Mahalanobis metric learning in our model can be illustrated as follows.

- **Begin**
- **Input** Matrix M, step-size α for adapting M, step-size μ for adapting $L(M)$, stop criterion θ, $t \leftarrow 0$.
  - **Do** $t \leftarrow t + 1$
  - Compute the gradient of objective function G^t at the t-th iteration.
  - $M_{(t)} \leftarrow M_{(t-1)} - \alpha G^t$
  - $M_{(t)} \leftarrow P\Lambda_+ P^T$
  - Compute the value of objective function $L(M_t)$ at the t-th iteration.
  - **Until** $|L(M_t) - L(M_{t-1})| \leq \theta$.
- **Output** M
- **End**

3 Experiment

3.1 Data Description and Configuration

The problem addressed in this paper is to forecast the future traffic flow rates at given roadway locations from the historical data on a transportation network. The data used in our experiments are from Beijing's Traffic Management Bureau. From the real urban traffic map, we select a representative patch to verify the proposed approach, which is given in Fig. 1 [10]. Each circle in the sketch map denotes a road junction. An arrow shows the direction of traffic flow, which reaches the corresponding road link from its upstream link. Paths without arrows are of no traffic flow records. Vehicular flow rates of discrete time series are recorded every 15 minutes. The recording period is 25 days (totally 2400 recorded

entries) from March, 2002. In our experiment, the raw data are divided into two sets, 2112 recorded entries of the first 22 days as the training set and the rest recorded entries as the test set. For evaluation, experiments are performed with multiple randomly selected roads from Fig. 1.

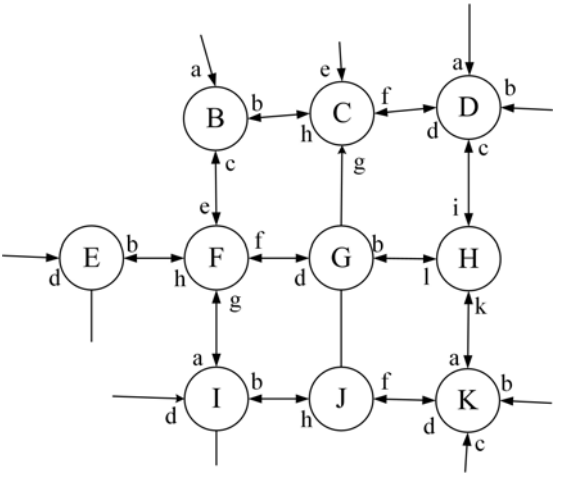


Fig. 1. A patch of traffic map taken from the East Section of the Third Circle of Beijing City Map where the UTC/SCOOT system is mounted

Let x_1, x_2, x_3, ..., x_{2400} denote the original 2400 ordered recorded entries. First we need to format the raw data into examples of vector form, which is represented as $(\boldsymbol{x}, y)$, where $\boldsymbol{x} \in R^d$ and $y \in R$. That is, for an example $(\boldsymbol{x}_i, y_i)$, y_i is the current traffic flow x_i and $\boldsymbol{x}_i$ is constructed by x_i's d past traffic flows x_{i-d}, x_{i-d+1}, ..., x_{i-1}. Then y_i, y_{i+1}, ..., y_{i+m-1} are used to predict y_{i+m}. In our experiment, d and m are empirically set as 15 and 8, respectively.

3.2 Experimental Results

The proposed SMLKR is applied to short-term traffic flow forecasting to evaluate its effectiveness. For comparison purpose, related kernel regression algorithms KR and MLKR are also conducted to serve as base lines. The objective of kernel regression is to make the accumulated quadratic leave-one-out regression error on test examples as small as possible. Therefore, $L = \sum_i (y_i - \hat{y}_i)^2$ is adopted as the first comparison criterion of the three kernel regression algorithms.

Besides accumulated quadratic leave-one-out regression error L, another two widely-used criterions are also adopted to evaluate the three algorithms. They are mean absolute relative error (MARE) and root mean squared error (RMSE), respectively, which are formulated as follows:

$$MARE = \frac{1}{n} \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{y_i}, \tag{13}$$

and

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \tag{14}$$

where n is the number of test examples.

The accumulated quadratic leave-one-out regression error on each training set and test set of three kernel regression algorithms are listed in Table 1. The bold number in the table represents that the corresponding algorithm performs best. The final dimension of the metric matrix M learnt by SMLKR are also reported in this table. In addition, performance comparison of the three algorithms based on MARE and RMSE on training sets and test sets are reported in Table 2 and Table 3 respectively. In order to give an intuitive illustration of the forecasting performance, we draw the forecasting results of Roadway Gb on the test set using KR, MLKR and SMLKR, which is shown in Fig. 2, where blue lines represent real recorded data and red stars represent forecasted results.

Real traffic flow forecasting results reported in Table 1, Table 2 and Table 3 reveal that MLKR and SMLKR are all superior to the traditional kernel regression algorithm KR, which means metric learning can effectively improve the performance of kernel regression algorithms. Different from MLKR, the proposed SMLKR is the first to combine kernel regression with sparse metric learning. As shown in Table 1, only SMLKR has the capability of learning a low-rank metric matrix.

Table 1. Comparison of L and final dimension of metric matrix M

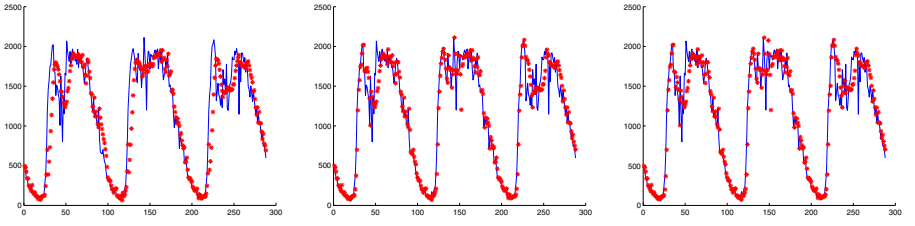
		L			
	Road	KR	MLKR	SMLKR	Dimension
Training	Ba	1.275e+008	4.852e+007	**4.646e+007**	8
	Cf	2.620e+007	**2.331e+007**	2.363e+007	8
	Fe	1.982e+008	5.408e+007	**5.384e+007**	7
	Gb	1.739e+007	1.532e+007	**1.461e+007**	10
	Hi	2.101e+007	1.683e+007	**1.615e+007**	9
Test	Ba	2.612e+007	1.103e+007	**1.090e+007**	
	Cf	3.690e+006	3.381e+006	**3.354e+006**	
	Fe	3.307e+007	8.330e+006	**8.325e+006**	
	Gb	3.107e+006	2.905e+006	**2.740e+006**	
	Hi	3.627e+006	3.370e+006	**3.356e+006**	

Table 2. Training error comparison

	MARE			RMSE		
ML	KR	MLKR	SMLKR	KR	MLKR	SMLKR
Ba	0.165	0.144	**0.143**	247.055	152.408	**149.138**
Cf	0.116	**0.113**	0.115	111.993	**105.622**	106.349
Fe	0.148	**0.112**	**0.112**	308.012	160.900	**160.759**
Gb	0.158	0.153	**0.150**	91.246	85.629	**83.642**
Hi	0.167	0.161	**0.158**	100.288	89.763	**87.714**

Table 3. Test error comparison

	MARE			RMSE		
ML	KR	MLKR	SMLKR	KR	MLKR	SMLKR
Ba	0.188	0.152	**0.150**	301.149	195.671	**194.565**
Cf	0.105	**0.102**	0.107	113.191	108.343	**107.918**
Fe	0.152	**0.108**	**0.108**	338.836	170.070	**170.017**
Gb	0.159	0.154	**0.151**	103.866	100.424	**97.533**
Hi	0.159	0.156	**0.155**	112.221	108.123	**107.949**



(a) KR (b) MLKR (c) SMLKR

Fig. 2. Forecasting results of KR, MLKR and SMLKR for Gb

Furthermore, SMLKR performs better than MLKR on almost all the datasets except Cf. Therefore, we can conclude that the proposed algorithm is better than KR and MLKR, it can learn a good metric and effectively remove noise leading to dimension reduction as well.

4 Conclusion

In this paper, a sparse kernel regression algorithm is proposed by introducing a mixed $(2,1)$-norm regularization over the metric matrix M into the objective function of kernel regression. The proposed algorithm is the first to combine kernel regression and sparse metric learning. When applied to short-term traffic flow forecasting, SMLKR gets the best forecasting results with comparison to two related kernel regression algorithms. The promising results demonstrate that SMLKR is an effective and better kernel regression algorithm for short-term traffic flow forecasting.

As an effective approach for short-term traffic flow forecasting, kernel regression with sparse metric learning seems to be another improvement in kernel regression. In the future, developing the potential of SMLKR in other domains is our pursuit.

Acknowledgments. This work is supported in part by the National Natural Science Foundation of China under Project 61075005, 2011 Shanghai Rising-Star Program, and the Fundamental Research Funds for the Central Universities.

References

1. Yu, G., Hu, J., Zhang, C., Zhuang, L., Song, J.: Short-Term Traffic Flow Forecasting based on Markov Chain Model. In: Proc. IEEE Intelligent Vehicles Symp., Columbus, OH, pp. 208–212 (2003)
2. Lee, S., Fambro, D.: Application of Subsets Autoregressive Integrated Moving Average Model for Short-Term Freeway Traffic Volume Forecasting. Transp. Res. Rec. 1678, 179–188 (1999)
3. Okutani, I., Stephanedes, Y.: Dynamic Prediction of Traffic Volume through Kalman Filter Theory. Transp. Res., Part B: Methodol. 18, 1–11 (1984)
4. Sun, S., Zhang, C.: A Bayesian Network Approach to Traffic Flow Forecasting. IEEE Trans. Intell. Transp. Syst. 7, 124–132 (2006)
5. Müller, K., Smola, A., Rätsch, G., Schölkopf, B., Kohlmorgen, J., Vapnik, V.: Predicting Time Series with Support Vector Machines. In: Gerstner, W., Hasler, M., Germond, A., Nicoud, J.-D. (eds.) ICANN 1997. LNCS, vol. 1327, pp. 999–1004. Springer, Heidelberg (1997)
6. Smith, B., Williams, B., Oswald, R.: Comparison of Parametric and Nonparametric Models for Traffic Flow Forecasting. Transp. Res., Part C: Emerg. Technol. 10, 303–321 (2002)
7. Sun, S., Chen, Q.: Kernel Regression with a Mahalanobis Metric for Short-Term Traffic Flow Forecasting. In: Fyfe, C., Kim, D., Lee, S.-Y., Yin, H. (eds.) IDEAL 2008. LNCS, vol. 5326, pp. 9–16. Springer, Heidelberg (2008)
8. Bach, F.: Consistency of Trace Norm Minimization. J. Mach. Lear. Res. 9, 1019–1048 (2008)
9. Weinberger, K., Tesauro, G.: Metric Learning for Kernel Regression. In: Proc. 11th Int. Conf. Artificial Intelligence and Statistics, pp. 608–615. Omnipress, Puerto Rico (2007)
10. Sun, S., Zhang, C.: The Selective Random Subspace Predictor for Traffic Flow Forecasting. IEEE Trans. Int. Transportation Systems, 367–373 (2007)
11. Ying, Y., Huang, K., Campbell, C.: Sparse Metric Learning via Smooth Optimization. In: NIPS (2009)
12. Argyriou, A., Evgeniou, T., Pontil, M.: Convex Multi-Task Feature Learning. Mach. Lear. 73, 243–272 (2008)
13. Duda, R., Hart, P., Stork, D.: Pattern Classification. John Wiley & Sons, New York (2000)

Rate-Dependent Hysteresis Modeling and Compensation Using Least Squares Support Vector Machines

Qingsong Xu*, Pak-Kin Wong, and Yangmin Li

Department of Electromechanical Engineering, Faculty of Science and Technology, University of Macau, Av. Padre Tomás Pereira, Taipa, Macao SAR, China
{qsxu,fstpkw,ymli}@umac.mo
http://www.fst.umac.mo/en/staff/fstqsx.html

Abstract. This paper is concentrated on the rate-dependent hysteresis modeling and compensation for a piezoelectric actuator. A least squares support vector machines (LS-SVM) model is proposed and trained by introducing the current input value and input variation rate as the input data set to formulate a one-to-one mapping. After demonstrating the effectiveness of the presented model, a LS-SVM inverse model based feedforward control combined with a PID feedback control is designed to compensate the hysteresis nonlinearity. Simulation results show that the hybrid scheme is superior to either of the stand-alone controllers, and the rate-dependent hysteresis is suppressed to a negligible level, which validate the effectiveness of the constructed controller. Owing to the simple procedure, the proposed modeling and control approaches are expected to be extended to other types of hysteretic systems as well.

Keywords: Piezoelectric actuator, hysteresis, least squares support vector machines (LS-SVM), motion control.

1 Introduction

Piezoelectric actuators (PEA) are capable of positioning with subnanometer resolution, rapid response, and large blocking force. Hence, they are popularly applied in various micro/nano positioning systems such as scanning probe microscopes and optical fibre alignment devices. However, the PEA introduces nonlinearity into the system mainly attributed to the piezoelectric hysteresis and drift effects. The hysteresis is a nonlinear relationship between the applied voltage and output displacement of the PEA and induces a severe open-loop positioning error of the system. Thus, the hysteresis has to be suppressed in high precision applications.

Extensive works have been carried out for the compensation of the hysteretic behaviors. Normally, the hysteresis is modeled with Preisach model [1], Prandtl-Ishlinskii model [2], Duhem model [3], or Dahl model [4], etc.. Then, an inverse

* Corresponding author, qsxu@umac.mo

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 85–93, 2011.

hysteresis model is constructed and utilized as an input shaper to cancel the hysteresis effect. However, the hysteresis effect is dependant not only on the amplitude but also on the frequency of input signals. It is very difficult to precisely capture the complicated rate-dependent hysteretic behavior. Most of the existing models employ a great number of parameters to describe the rate-dependent hysteresis [5,6], which blocks their use in real-time control as an adverse effect. Recently, it has been shown that artificial neural networks (ANN) provide an efficient way to model the nonlinear hysteresis [7,8]. Nevertheless, ANN have the problems of overfitting and sinking into local optima, which are their major drawbacks for practical applications. Alternatively, support vector machines (SVM) are a promising way to estimate nonlinear system models accurately. Based on statistical learning theory and structural risk minimization principle, the SVM approach is capable of modeling nonlinear systems by transforming the regression problem into a convex quadratic programming (QP) problem and then solving it with a QP solver. Compared to conventional ANN, SVM have the major advantages of global optimization and higher generalization capability. Furthermore, the least squares support vector machines (LS-SVM) utilize equality constraints instead of the inequality constraints as in the ordinary SVM. Hence, it simplifies the regression to a problem that can be easily solved from a set of linear equations [9].

Although SVM have been widely applied to solve classification and regression problems [10], their application in the treatment of hysteresis is still limited. To the best knowledge of the authors, only a few of previous works employ the SVM techniques for the modeling and compensation of the hysteresis. Specifically, a SVM-based feedforward controller combined with a self-tuning PID controller is presented in [11] for the control of a PEA. However, details about how to model the hysteresis are not given in the paper. Reference [12] proposes two SVM-based hysteresis modeling methods by using the hysteresis curve direction as one of input variables and adopting an improved version based on the autoregressive algorithm, respectively. The superiority of SVM over ANN in term of modeling accuracy is illustrated in the paper. Additionally, a LS-SVM model is proposed in [13] for the hysteresis modeling by introducing an input sequences matrix to transform the multi-valued mapping to one-to-one mapping. Besides, the hysteresis modeling and compensation of a humidity sensor is performed in [14] by using two SVM and treating the increasing and decreasing curves separately. The aforementioned three works formulate the one-to-one mapping by defining the hysteresis loop directions firstly, which are not feasible especially when the input signals are arbitrary and unknown beforehand. The rate-dependent hysteresis for a giant magnetostrictive actuator (GMA) is modeled in [15] based on LS-SVM by using an NARX model to form a one-to-one mapping of the hysteresis nonlinearity. However, hysteresis modeling is only carried out for the major hysteresis loop of the GMA actuator, whereas no compensation issues are mentioned.

In the current research, we propose a LS-SVM-based rate-dependent hysteresis model for a PEA by introducing the current input value and input variation

rate as one data set to construct a one-to-one mapping, which is much simpler and more intuitive than the previous approaches. By adopting the RBF kernel function, the LS-SVM model only has two parameters to be tuned. Moreover, the hysteresis nonlinearity is suppressed by a hybrid control employing a LS-SVM inverse model based feedforward controller combined with an incremental-type PID feedback controller. The effectiveness of the presented modeling and control approaches will be validated by a series of simulation studies.

2 LS-SVM-Based Hysteresis Modeling

The hysteresis modeling is treated as a nonlinear regression problem, and the LS-SVM is employed to model the piezoelectric hysteresis for a PEA in this section. Owing to the hysteresis effects, an input corresponds to multiple outputs. Thus, one of the challenges lies in how to convert the one-to-many mapping into a one-to-one mapping. In this research, both the current input and input variation rate are introduced to form the input data set, which determines a unique output value. By taking into account the input variation rate, the rate dependency of the hysteretic behavior will be captured as well.

The hysteresis model can be identified by using the input voltage (U) and voltage variation rate ($\dot{U}$) as the inputs and the displacement (Y) as the output to train the LS-SVM as outlined below.

2.1 LS-SVM Modeling

LS-SVM maps the input data into a high dimensional feature space and constructs a linear regression function therein. The unknown hysteresis function is approximated by the equation:

$$y(x) = w^T \varphi(x) + b \tag{1}$$

with the given training data $\{x_i, y_i\}_{i=1}^N$ where N represents the number of training data, $x_i = \{U_i, \dot{U}_i\}$ are the input data and $y_i = \{Y_i\}$ are the output data. Additionally, the weight vector $w \in R^{n_h}$, the nonlinear mapping $\varphi(\cdot) : R^2 \to R^{n_h}$ denotes a map from the input space to a feature space, and b is the bias term.

The LS-SVM approach formulates the regression as an optimization problem:

$$\min_{w,b,e} J(w, e) = \frac{1}{2} w^T w + \frac{1}{2} \gamma \sum_{i=1}^{N} e_i^2 \tag{2}$$

subject to the equality constraints:

$$y_i = w^T \varphi(x_i) + b + e_i, \quad i = 1, 2, \cdots, N. \tag{3}$$

In order to solve the optimization problem, a Lagrangian function is defined:

$$L(w, b, e; \alpha) = J(w, e) - \sum_{i=1}^{N} \alpha_i [w^T \varphi(x_i) + b + e_i - y_i] \tag{4}$$

where α_i are the Lagrange multipliers which can be either positive or negative values. The conditions for optimality can be obtained by solving the following partial derivatives:

$$\frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{i=1}^{N} \alpha_i \varphi(x_i) \tag{5a}$$

$$\frac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^{N} \alpha_i = 0 \tag{5b}$$

$$\frac{\partial L}{\partial e_i} = 0 \rightarrow \alpha_i = \gamma e_i, \quad i = 1, 2, \cdots, N \tag{5c}$$

$$\frac{\partial L}{\partial \alpha_i} = 0 \rightarrow w^T \varphi(x_i) + b + e_i - y_i = 0, \quad i = 1, 2, \cdots, N \tag{5d}$$

which can be assembled in the matrix form by eliminating w and e_i, i.e.,

$$\underbrace{\begin{bmatrix} 0 & 1_s^T \\ 1_s & \Omega + \gamma^{-1} I \end{bmatrix}}_{\Phi} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \tag{6}$$

where $1_s = [1, 1, \cdots, 1]^T$, $\alpha = [\alpha_1, \alpha_2, \cdots, \alpha_N]^T$, $y = [y_1, y_2, \cdots, y_N]^T$, I is an identity matrix, and $\Omega_{ij} = \varphi(x_i)^T \varphi(x_j) = K(x_i, x_j)$ with $i, j = 1, 2, \cdots, N$.

It is observed that the LS-SVM approach utilizes the equality constraints instead of the inequality constraints as in the ordinary SVM. Thus, it simplifies the regression to a problem that can be easily solved from a set of linear equations. Assume that Φ is invertible, then b and α can be calculated from (6):

$$\begin{bmatrix} b \\ \alpha \end{bmatrix} = \Phi^{-1} \begin{bmatrix} 0 \\ y \end{bmatrix} \tag{7}$$

Thus, in view of (5), one can derive the solution for the regression problem:

$$y(x) = \sum_{i=1}^{N} \alpha_i K(x, x_i) + b \tag{8}$$

where $K(x, x_i)$ is the kernel function satisfying Mercer's condition, x_i is the training data, and x denotes the new input hereafter.

By adopting the RBF kernel function:

$$K(x, x_i) = \exp\left(-\frac{\|x - x_i\|^2}{\sigma^2}\right) \tag{9}$$

with the width parameter $\sigma > 0$, the LS-SVM model for the hysteresis model estimation becomes

$$Y(x) = \sum_{i=1}^{N} \alpha_i \exp\left(-\frac{\|x - x_i\|^2}{\sigma^2}\right) + b. \tag{10}$$

With the assigned regularization parameter γ and kernel parameter σ, the purpose of the training process is to determine the values of α_i and b.

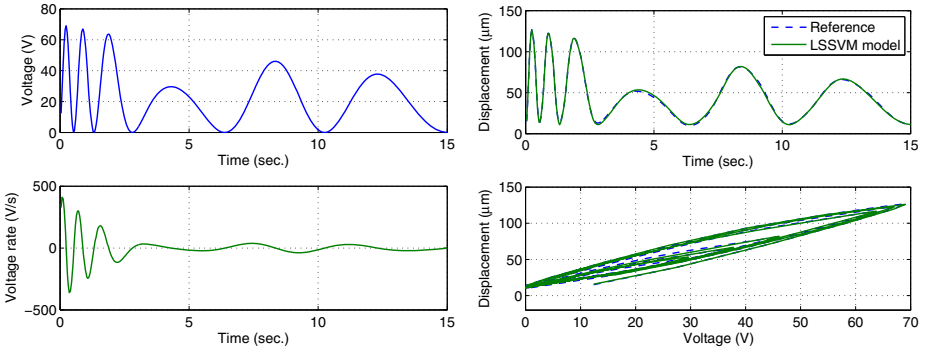


Fig. 1. (a) Input data sets for LS-SVM training; (b) output of the trained LS-SVM hysteresis model

2.2 Simulation Studies

For simulation studies in the current research, a PEA hysteretic system is represented by the Bouc-Wen hysteresis model as follows [16]:

$$M\ddot{Y} + B\dot{Y} + KY = K(DU - H) \tag{11}$$

$$\dot{H} = \kappa D\dot{U} - \beta|\dot{U}|H - \nu\dot{U}|H| \tag{12}$$

where M, B, K, and Y represent the mass, damping coefficient, stiffness, and output displacement of the system, respectively; d is the piezoelectric coefficient, U denotes the input voltage, and H indicates the hysteretic loop in terms of displacement whose magnitude and shape are determined by parameters κ, β, and ν. The following model parameters are selected [17]: $M = 0.1040$ kg, $B = 1.8625\times10^4$ N s/m, $K = 1.9920\times10^5$ N/m, $D = 1.8347\times10^{-5}$ m/V, $\kappa = 0.0786$, $\beta = 0.0995$, and $\nu = 0.0008$.

For the training process, the input voltage signal [see Fig. 1(a)] is chosen as:

$$U(t) = 35e^{-0.05t}[\sin(4.0\pi e^{-0.23t}t - 1.2) + 1]. \tag{13}$$

The output data are depicted in Fig. 1(b) which are generated by the Bouc-Wen simulation model. The input and output data sets are then adopted to train the LS-SVM. The LS-SVM model parameters (γ and σ) are tuned in two steps [18]. Specifically, the parameters are first determined using the coupled simulated annealing optimization technique, and then finely tuned by resorting to the simplex optimization procedure, which give $\gamma = 3112.67$ and $\sigma = 3.23$. Once the training is completed, the LS-SVM model produces the output as shown in Fig. 1(b), which exhibits that the LS-SVM model approximates the actual output accurately.

To verify the generalization ability of the obtained model, the input signal [see Fig. 2(a)] is selected as:

$$U_t(t) = 35e^{-0.07t}[\sin(3.8\pi e^{-0.22t}t - 1.2) + 1]. \tag{14}$$

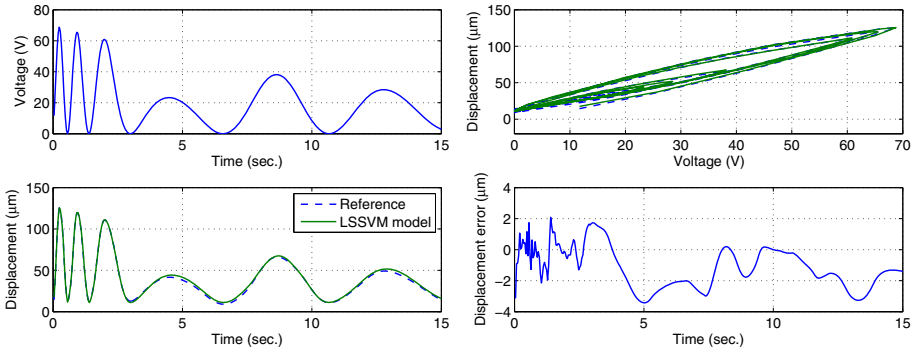


Fig. 2. (a) Input and output of the LS-SVM model; (b) output and output error of the LS-SVM hysteresis model

The LS-SVM model outputs are depicted in Figs. 2(a) and (b). The model output error with respect to the "actual" output (Y_d) given by the Bouc-Wen model is illustrated in Fig. 2(b). Based on the displacement error $E = Y_d - Y$, the mean absolute error (MAE) and root mean square error (RMSE) are defined as follows:

$$\text{MAE} = \frac{1}{N_t}\sum_{i=1}^{N_t}|E_i| \tag{15}$$

$$\text{RMSE} = \sqrt{\frac{1}{N_t}\sum_{i=1}^{N_t}E_i^2} \tag{16}$$

where N_t is the number of test data sets. It is observed that the MAE and RMSE are 1.46 μm and 1.77 μm, which accounts for 1.2% and 1.4% of the overall motion range, respectively. The results reveal the efficiency of the established LS-SVM hysteresis model.

3 Controller Design and Verification

3.1 Controller Design

In order to construct a feedforward control to suppress the hysteresis, an inverse hysteresis model can be identified by the same principle as shown in Section 2.1 with the selection of the displacement (Y) and displacement rate ($\dot{Y}$) as the inputs and the corresponding voltage (U) as the output for the LS-SVM training. Once trained offline, the LS-SVM inverse model provides online the feedforward (FF) control effort U_{FF}.

Due to the existence of the modeling error, the hysteresis cannot be completed eliminated by the stand-alone inverse model-based FF compensator. Therefore, a PID feedback (FB) control is employed to create a hybrid control as shown in Fig. 3.

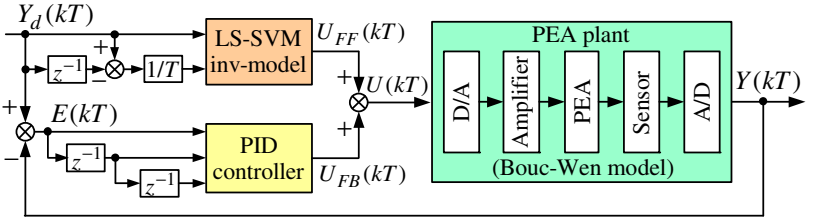


Fig. 3. Block diagram of the feedforward (FF) plus feedback (FB) hybrid controller for a PEA system

By adopting an incremental PID algorithm, the overall control input can be derived in the discretized form:

$$\begin{aligned} U(kT) &= U_{FF}(kT) + U_{FB}(kT) \\ &= U_{FF}(kT) + U_{FB}(kT - T) + K_p\,[E(kT) - E(kT - T)] + K_i\,E(kT) \\ &\quad + K_d\,[E(kT) - 2E(kT - T) + E(kT - 2T)] \end{aligned} \tag{17}$$

where E represents the displacement error, T is the sampling time, k denotes the index of time series, $U_{FB}(kT - T)$ is the FB control command in the previous time step, and the FF term $U_{FF}(kT)$ is given by the LS-SVM inverse hysteresis model. Additionally, K_p, K_i, and K_d are the positive control gains.

3.2 Controller Verification

The LS-SVM inverse hysteresis model is trained using the tuned parameters $\gamma = 294.23$ and $\sigma = 2.58$. In addition, the sampling time interval is assigned as 0.01 s. The PID gains are chosen as $K_p = 1.980$, $K_i = 2.585$, and $K_d = 0.015$ by the trial and error approach. With the reference input as shown in Fig. 2(a), the control results and tracking errors of the FF, FB, and FF+FB methods are shown in Fig. 4(a). The effectiveness of the FF+FB hybrid control is evident from the control results. Specifically, the FF+FB produces the MAE and RMSE of 1.4% and 1.6%, respectively, which have been significantly improved by 73.0% and 72.3% in comparison with the FF control results, and substantially enhanced by 49.4% and 55.6% compared to the stand-alone FB outputs, respectively. As a result, the hysteresis effects have been suppressed by the FF+FB approach to a negligible level as indicated in Fig. 4(b). The control efforts of the three methods are compared in Fig. 4(b). The RMS control inputs are 24.3 V, 22.3 V, and 23.3 V, respectively, which means that the hybrid controller produces the best results with the moderate magnitude of the control effort.

The results show that introducing the variation rate as an auxiliary input is enough to establish a one-to-one mapping between the input and output data. In the future, more training data sets will be employed to generate a more accurate model. Experimental studies will be conducted to test the effectiveness of the proposed control, and comparison studies will be performed with respect to other hysteresis models.

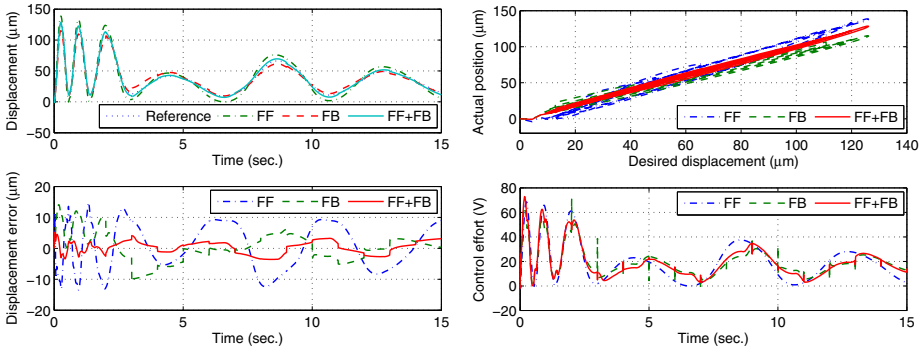


Fig. 4. (a) Control results and control errors of feedforward (FF), feedback (FB), and FF+FB approaches; (b) hysteresis loops and control efforts produced by the three controllers

4 Conclusions

The presented investigation shows that the rate-dependent hysteresis of a piezoelectric actuator can be accurately modeled by the LS-SVM regression model. By selecting the input variation rate as an auxiliary input variable, the multi-valued mapping due to the hysteresis nonlinearity is converted into a one-to-one mapping, and the LS-SVM is trained to capture the rate-dependent hysteretic behavior. Simulation results demonstrate that the hybrid control using the LS-SVM inverse model based feedforward control combined with a simple PID control is capable of suppressing the hysteresis nonlinearity effectively. Due to the simple structure of the presented modeling and control framework, it can be easily extended to hysteretic systems driven by shape memory alloy or other types of smart actuators as well.

References

1. Tan, X., Baras, J.S.: Adaptive identification and control of hysteresis in smart materials. IEEE Trans. Automat. Contr. 50(6), 827–839 (2005)
2. Kuhnen, K.: Modeling, identification and compensation of complex hysteretic nonlinearities: A modified Prandtl-Ishlinskii approach. European J. Control 9(4), 407–421 (2003)
3. Xie, W.F., Fu, J., Su, C.Y.: Observer based control of piezoelectric actuators with classical Duhem modeled hysteresis. In: Proc. of American Control Conf., pp. 4221–4226 (2009)
4. Xu, Q., Li, Y.: Dahl model-based hysteresis compensation and precise positioning control of an XY parallel micromanipulator with piezoelectric actuation. J. Dyn. Syst. Meas. Control-Trans. ASME 132(4), 041011 (2010)
5. Yu, Y., Xiao, Z., Naganathan, N.G., Dukkipati, R.V.: Dynamic Preisach modelling of hysteresis for the piezoceramic actuator system. Mech. Mach. Theory 37(1), 75–89 (2002)

6. Ang, W.T., Khosla, P.K., Riviere, C.N.: Feedforward controller with inverse rate-dependent model for piezoelectric actuators in trajectory-tracking applications. IEEE/ASME Trans. Mechatron. 12(2), 134–142 (2007)
7. Yu, S., Alici, G., Shirinzadeh, B., Smith, J.: Sliding mode control of a piezoelectric actuator with neural network compensating rate-dependent hysteresis. In: Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 3641–3645 (2005)
8. Dong, R., Tan, Y., Chen, H., Xie, Y.: A neural networks based model for rate-dependent hysteresis for piezoelectric actuators. Sens. Actuator A-Phys. 143(2), 370–376 (2008)
9. Wong, P.K., Vong, C.M., Tam, L.M., Li, K.: Data preprocessing and modelling of electronically-controlled automotive engine power performance using kernel principal components analysis and least-square support vector machines. Int. J. Vehicle Systems Modelling and Testing 3(4), 312–330 (2008)
10. Suykens, J.A.K.: Support vector machines: a nonlinear modeling and control perspective. European J. Control 7(2-3), 311–327 (2001)
11. Ji, H.W., Yang, S.X., Wu, Z.T., Yan, G.B.: Precision control of piezoelectric actuator using support vector regression nonlinear model and neural networks. In: Proc. of Int. Conf. on Machine Learning and Cybernetics, pp. 1186–1191 (2005)
12. Wang, B., Zhong, W., Pi, D., Sun, Y., Xu, C., Chu, S.: Support vector machine based modeling of nonlinear systems with hysteresis. In: Proc. of 6th World Congress on Intelligent Control and Automation, pp. 1722–1725 (2006)
13. Yang, X.F., Li, W., Wang, Y.Q., Su, X.P.: A multi-loop hysteresis model of piezo actuator based on LS-SVM. In: Proc. of 7th Int. Conf. on Asia Simulation Conf., pp. 1451–1454 (2008)
14. Wang, X., Ye, M.: Hysteresis and nonlinearity compensation of relative humidity sensor using support vector machines. Sens. Actuator A-Phys. 129(1), 274–284 (2008)
15. Lei, W., Mao, J., Ma, Y.: A new modeling method for nonlinear rate-dependent hysteresis system based on LS-SVM. In: Proc. of 10th Int. Conf. on Control, Automation, Robotics and Vision, pp. 1442–1446 (2008)
16. Li, Y., Xu, Q.: Adaptive sliding mode control with perturbation estimation and PID sliding surface for motion tracking of a piezo-driven micromanipulator. IEEE Trans. Contr. Syst. Technol. 18(4), 798–810 (2010)
17. Xu, Q., Li, Y.: Fuzzy sliding mode control with perturbation estimation for a piezoactuated micromanipulator. In: Zeng, Z., Wang, J. (eds.) Advances in Neural Network Research and Applications. Lecture Notes in Electrical Engineering, vol. 67, pp. 153–160. Springer, Heidelberg (2010)
18. De Brabanter, K., Karsmakers, P., Ojeda, F., Alzate, C., De Brabanter, J., Pelckmans, K., De Moor, B., Vandewalle, J., Suykens, J.A.K.: LS-SVMlab toolbox user's guide version 1.7. Internal Report 10-146, ESAT-SISTA, K.U. Leuven, Leuven, Belgium (2010)

Nomogram Visualization for Ranking Support Vector Machine

Nguyen Thi Thanh Thuy[1], Nguyen Thi Ngoc Vinh[1], and Ngo Anh Vien[2]

[1] Faculty of Information Technology,
Posts and Telecommunications Institute of Technology, Hanoi, Vietnam
thuyr205@gmail.com, ntngocvinh@yahoo.com
[2] Department of Computer Science, School of Computing,
National University of Singapore, Singapore
ngoav@comp.nus.edu.sg

Abstract. In this paper, we propose a visualization model for a trained ranking support vector machine. In addition, we also introduce a feature selection method for the ranking support vector machine, and show visually each feature's effect. Nomogram is a well-known visualization model that graphically describes completely the model on a single graph. The complexity of the visualization does not depend on the number of the features but on the properties of the kernel. In order to represent the effect of each feature on the log odds ratio on the nomograms, we use probabilistic ranking support vector machines which map the support vector machine outputs into a probabilistic sigmoid function whose parameters are trained by using cross-validation. The experiments show the effectiveness of our proposal which helps the analysts study the effects of predictive features.

Keywords: Nomogram, visualization, SVM, ranking SVM, probabilistic ranking SVM.

1 Introduction

Ranking support vector machine (SVM) [1] is the most favorite ranking method that was applied to various different applications [2], [3], [4]. Besides its various advantages, ranking SVM still has difficulty in intuitively presenting the classifier which is also the disadvantage of original SVM. Inspired by the nomogram based visualization for SVMs of Jakulin [5], we also proposed a method which intuitively presents the ranking SVM. In order to present a ranking SVM on a nomogram, we must use the posterior probabilities of the output of ranking SVM proposed in [6].

Feature selection recently has gained increasing attention in the data mining field with many applications such as text mining, bioinformatics, sensor networks, etc. Feature selection selects a subset of relevant features, and also removes irrelevant and redundant features from the data to build robust learning models. Many applications deal with a very large number of features (for example, tens or hundreds

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 94–102, 2011.

of thousands of features) and often comparably few training samples. There are many potential benefits of feature selection: facilitating data visualization and data understanding, reducing the measurement and storage requirements, reducing training and utilization times, and defying the curse of dimensionality to improve prediction performance [7].

Inspired by the nomogram-based recursive feature elimination (RFE) methods in classification problem [8], in this paper we propose a nomogram based feature selection for a ranking problem. Nomogram-based RFE feature selection is a method in which a feature is more important when the length of its line in the nomogram representation is longer. Consequently, features having small effect are removed by computing their length in the nomogram representation. Because, features with small effect means noisy or redundant features which reduce the accuracy of the classifier. So our contribution are two-folds: firstly, we propose a nomogram based visualization method for ranking SVMs. Secondly, based on the nomogram presentation, we propose a nomogram-based RFE feature selection method for ranking problems.

The remaining of this paper is organized as follows: First, we briefly summarize an approach for visualization of Support Vector Machines in section 2. Following is the nomogram-based RFE algorithm to eliminate irrelevant and redundant features which having the shortest length in the nomogram. In section 3, we propose a new approach that uses nomogram to visualize ranking Support Vector Machine. And finally, experimental results and conclusions are described in section 4 and section 5, respectively.

2 Nomogram Visualization in Classification Problem

2.1 Nomogram Visualization for SVM

In this section, we briefly discuss how to visualize a Support Vector Machines (SVM) model with a method proposed by Jakulin in [5]. This approach employs logistic regression to convert the distance from the separating hyperplane into a probability, and then represents the effect of each predictive feature on the log odds ratio scale as required for the nomogram. The main advantage of this approach is that it captures a complete classification model in a single, easy-to-interpret graph and for all common types of attributes and even for non-linear SVM kernels.

Suppose that we have a training dataset $\mathcal{D} = \{\mathbf{x}_i, y_i\}_1^l$ in which $\mathbf{x}_i$[1] is a feature vector in n dimensional feature space $\Re^n$ and $y_i \in \{+1, -1\}$ is the class label of $\mathbf{x}_i$. The distance from a sample $(\mathbf{x}_i, y_i)$ to the separating hyperplane of the SVM can be replaced by the decision function in the SVM as follows [9], [10]:

$$f(\mathbf{x}) = \sum_{1}^{M} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \tag{1}$$

[1] We denote the bold variables as vectors or matrices.

where $M(< l)$ is the number of support vectors, $\alpha_i > 0$ are the Lagrange multipliers for support vectors, b is bias, and $K(\mathbf{x}_i, \mathbf{x})$ is called kernel function, that returns a similarity between $\mathbf{x}_i$ and $\mathbf{x}$. Depending on positive or negative sign of $f(\mathbf{x})$, SVM classifier predicts the label of an unknown instance of the testing dataset.

In the case of linearly decomposable kernel with respect to each feature, the distance becomes:

$$f(\mathbf{x}) = \sum_{k=1}^{n} [\mathbf{w}]_k + b \tag{2}$$

and the weight vector is defined as:

$$[\mathbf{w}]_k = \sum_{i=1}^{M} \alpha_i y_i K(\mathbf{x}_{i,k}, \mathbf{x}_k) \tag{3}$$

where n is the number of features, $\mathbf{x}_k$ is kth feature of sample $\mathbf{x}$, and $\mathbf{x}_{i,k}$ is kth feature of ith support vector [5], [8].

According to the method presented in [11], the posterior probability that the sample $\mathbf{x}$ belong to the positive class (in binary classification problem) is calculated as:

$$P(y = +1|\mathbf{x}) = \frac{1}{1 + \exp(Af(\mathbf{x}) + B)} \tag{4}$$

The two parameters A and B are fitted using maximum likelihood estimation from a training set and found by minimizing the negative log likelihood function of the training data. To avoid overfitting, a cross-validation method is used.

After finding two parameters A and B, these symbols A, B, $\mathbf{w}$ and b can be rewritten as the intercept β_0 and the effect function β. The intercept β_0 is a constant delineating the prior probability in the absence of any features, and the effect function β maps the value of a feature for the instance $\mathbf{x}$ into a point score, and finally using the inverse link function maps these functions into the outcome probability for an instance. The nomogram is based upon one effect function for each feature. Each line in the nomogram corresponds to a single feature, and a single effect function. The mapping is as follows:

$$\beta_0 = Ab + B \tag{5}$$

$$[\beta]_k = A[\mathbf{w}]_k \tag{6}$$

Then, the posterior probability (4) can be rewritten as:

$$P(y = +1|\mathbf{x}) = \frac{1}{1 + \exp(\beta_0 + \sum_{1}^{n} [\beta]_k)} \tag{7}$$

2.2 Nomogram-Based Recursive Feature Elimination (RFE)

Nomogram-based RFE algorithm [8] is implemented via 3-fold cross validations (Please refer to [8] for more detail). Initially, the selected feature list is set to null, the training subset of features (or surviving features) is the full set of features. At each iteration, we run 3-fold cross validations to get the accuracy with the current subset of the surviving features. This accuracy is compared to the stored best accuracy (initially, best accuracy = 0). If the accuracy is greater, the selected feature list is set to the current subset of the surviving features and update the best accuracy to the current accuracy. At the end of each iteration, we will eliminate one feature from the current subset of the surviving features. The eliminated feature is the one having the shortest length in the nomogram. To compute the length of each feature in the nomogram, we train an SVM model with the restricted samples (the current subset of the surviving features) and compute the nomogram representation from the SVM model. The next iteration is implemented with the new subset of the surviving features. The loop ends when the subset of the surviving features is empty.

3 Proposed Nomogram Visualization for Ranking SVM

Similarly to the drawing of a nomogram for a SVM that is summarized in section 2, we propose a nomogram visualization for a ranking SVM. Assume that there is an input space $X \in \Re^n$, where n is the dimension. And assume that we are given a ranking dataset (detailed in [6], [1], [4])

$$\mathcal{D}' = \{\boldsymbol{x}_i^{(1)} - \boldsymbol{x}_i^{(2)}, z_i\}_1^k, \quad x_i^{(1)}, x_i^{(2)} \in X \text{ for } i = 1, \ldots, k \tag{8}$$

And the score of the ranking SVM function is expressed as:

$$f(\mathbf{x}) = \sum_1^M \alpha_i z_i K(\mathbf{x}_i^{(1)} - \mathbf{x}_i^{(2)}, \mathbf{x}) + b \tag{9}$$

where $M(< k)$ is the number of support vectors in the ranking problem. When the kernel is linearly decomposable with respect to each feature, the distance becomes:

$$f(\mathbf{x}) = \sum_{k=1}^n [\mathbf{w}]_k + b \tag{10}$$

and the weight vector is defined as:

$$[\mathbf{w}]_k = \sum_{i=1}^M \alpha_i z_i K(\mathbf{x}_{i,k}^{(1)} - \mathbf{x}_{i,k}^{(2)}, \mathbf{x}_k) \tag{11}$$

where n is the number of features, $\mathbf{x}_k$ is kth feature of sample $\mathbf{x}$, and $(\mathbf{x}_{i,k}^{(1)} - \mathbf{x}_{i,k}^{(2)})$ is kth feature of ith support vector.

The posterior probability that the sample $(\mathbf{x}^{(1)} - \mathbf{x}^{(2)})$ belong to the positive class, it means that $z = +1$ or $(\mathbf{x}^{(1)} > \mathbf{x}^{(2)})$, is calculated as:

$$P(\mathbf{x}^{(1)} > \mathbf{x}^{(2)} | \mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = \frac{1}{1 + \exp\{Af(\mathbf{x}^{(1)} - \mathbf{x}^{(2)}) + B\}} \tag{12}$$

The above posterior probability for an output of ranking SVM was proposed in [6] which also discussed how to find the two parameters A and B.

Similarity with the nomogram visualization with SVM, we convert A, B, $\mathbf{w}$ and b to the intercept β_0 and the effect vector β, and use these parameters to represent the line of the Log OR for the feature in a nomogram.

$$\beta_0 = Ab + B \tag{13}$$

$$[\beta]_k = A[\mathbf{w}]_k \tag{14}$$

Thus, the posterior probability (12) can be rewritten as:

$$P(\mathbf{x}^{(1)} > \mathbf{x}^{(2)} | \mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = \frac{1}{1 + \exp(\beta_0 + \sum_{k=1}^{n} [\beta]_k)} \tag{15}$$

Here, we also use the nomogram based-RFE algorithm as the same in section 2 to eliminate irrelevant and redundant features which having the shortest length in the nomogram. Instead of considering the input with the beginning classification training set $\mathcal{D}$, we run the algorithm with the ranking training set $\mathcal{D}'$. It means that, the training samples consist of all pairs $(\mathbf{x}_i^{(1)} - \mathbf{x}_i^{(2)})$ with their class labels z_i. Other steps in the algorithm are invariant. The output is the best feature list.

4 Experimental Results

We evaluate the performance of ranking support vector machine visualization on the OHSUMED datasets using LIBSVM [12] and VRIFA [13]. We test our nomogram based method with two kernel: linear and the localized radial basic function (LRBF) kernel, that is a nonlinear kernel which was proposed by B.H.Cho in [8]. Both of them are proved to be linearly decomposable kernels.

OHSUMED dataset is available in the LETOR package [14]. OHSUMED dataset consists of 348,566 references and 106 queries, which is a subset of MEDLINE, a database on medical publications. It extracted 25 features (10 from title, 10 from abstract, and 5 from title + abstract). There are totally 16,140 query-document pairs with relevance judgments. The relevance degrees of documents with respect to each query are judged on three levels: definitely relevant, possibly relevant, or irrelevant.

Figure 1 shows the nomogram of a linear ranking SVM. We observe that the feature 20 has the widest range (that means the most important), whereas the features $5, 6, 7, 15, 16, 17$ have effect ranges equal to zero that means they contribute none to the accuracy of the classifier. This is due to an observation that the values of those

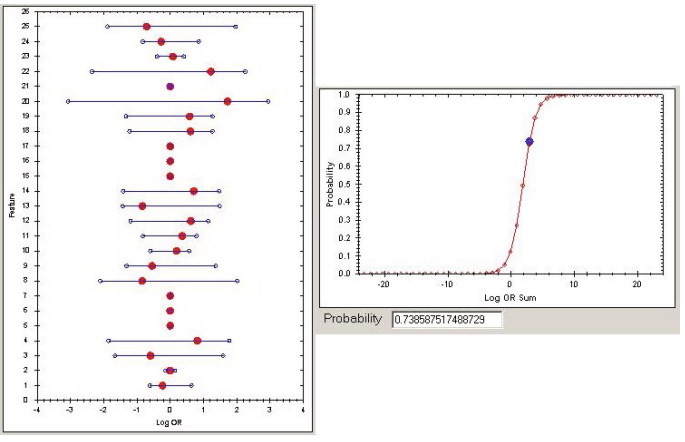


Fig. 1. Nomogram visualization with a linear ranking SVM without using feature selection. The left panel shows the effect ranges of all features, with an input instance indicated by a red circle. The right panel shows the probability map and the final probability output with that respective instance.

features in the dataset are all equal to each other (because we only read a certain query for ranking), so it makes the ranking dataset with all data pair $x_i^{(1)} - x_i^{(2)}$ at feature ith is equal to zero. Thus $z_i = 0$, then $[\mathbf{w}]_k$ in 3 ie equal to zero, so the effect function in 6 is equal to zero. These features are called noisy features. The result in Fig. 2 show only a subset of the selected features on the nomogram which makes the largest accuracy of the cross-validation. Figure 2 shows the accuracies of various feature selection. We observe that the best subset of feature has 17 features (with highest accuracy is 0.76302) which are drawn on the left panel. The eliminated features are: 1, 2, 5, 6, 7, 10, 11, 15, 16 ,17, 22, and 23.

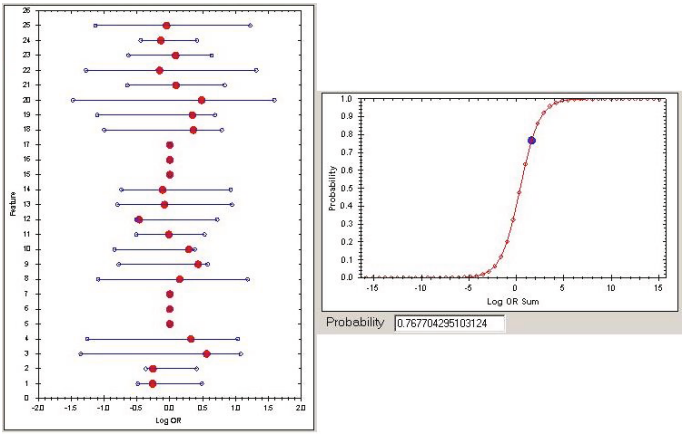


Fig. 2. Nomogram visualization with a LRBF ranking SVM without using feature selection

Figures 3 and 4 draw the respective nomogram of a LRBF kernel ranking SVM. Fig. 3 draws the nomogram without feature selection. Fig. 4 draw the best subset of selected features on the nomogram, and the the accuracies depending on the various subsets of selected features. The eliminated features are: 2, 5, 6, 7, 15, 16 ,17, and 24.

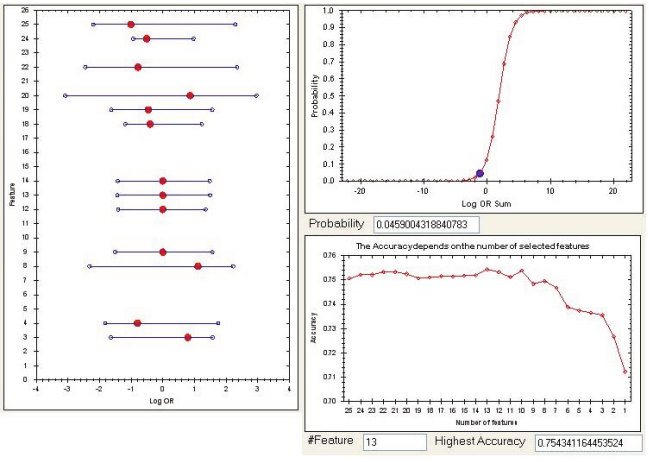


Fig. 3. Nomogram visualization with a linear ranking SVM using feature selection. The left panel shows the effect ranges of the best subset of selected features, with an input instance indicated by a red circle. The top-right panel shows the probability map and the final probability output with that respective instance. The bottom-right panel shows the accuracy depending on the number of selected features. The best subset of selected features is 13 which gives the highest accuracy = 0.7543.

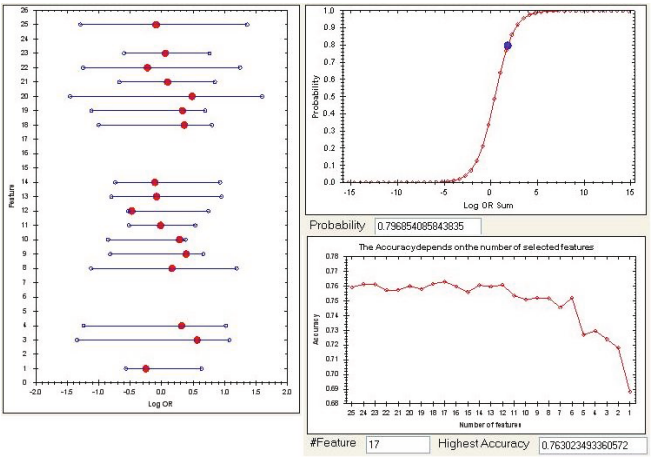


Fig. 4. Nomogram visualization with a LBRF ranking SVM using feature selection. The best subset of selected features is 17 which gives the highest accuracy = 0.7630.

5 Conclusion

In this paper, we proposed a nomogram based method to effectively visualize ranking support vector machines. Nomogram showed us its effectiveness in presenting ranking SVM with many dimensions. More specifically, individual features are drawn vertically in a nomogram. Each line on the nomogram shows the effect of one feature. In order to draw this nomogram, calibrated ranking SVM outputs [6] were used to calculate the effect function of features, and the ranking SVM was re-written in the form of a generalized additive model. Through nomogram presentation, analysts can have an insight and study the effects of predictive factors. Moreover, using nomogram presentation, we proposed a nomogram based-RFE algorithm for a ranking SVM. The proposed feature selection showed its robustness in eliminating noisy and redundant features, then improved the overall accuracy. In the experiment, we drew nomograms with both linear and nonlinear (LRBF [8]) kernels which are both linearly decomposable.

References

1. Herbrich, R., Graepel, T., Obermayer, K.: Large margin rank boundaries for ordinal regression. In: Advances in Large Margin Classifiers, pp. 115–132. MIT Press, Cambridge (2000)
2. Yu, H.: Svm selective sampling for ranking with application to data retrieval. In: The Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, pp. 354–363. ACM, New York (2005)
3. Yu, H., Kim, Y., Hwang, S.W.: Rvm: An efficient method for learning ranking svm. Technical report, Department of Computer Science and Engineering, Pohang University of Science and Technology (POSTECH), Korea (2008), http://iis.hwanjoyu.org/rvm
4. Cao, Y., Xu, J., Liu, T.Y., Li, H., Huang, Y., Hon, H.W.: Adapting ranking svm to document retrieval. In: ACM SIGIR 2006, pp. 186–193 (2006)
5. Jakulin, A., Mozina, M., Demsar, J., Bratko, I., Zupan, B.: Nomograms for visualizing support vector machines. In: The Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, pp. 108–117 (2005)
6. Thuy, N.T.T., Vien, N.A., Viet, N.H., Chung, T.: Probabilistic ranking support vector machine. In: Yu, W., He, H., Zhang, N. (eds.) ISNN 2009. LNCS, vol. 5552, pp. 345–353. Springer, Heidelberg (2009)
7. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Machine Learning Research 3, 1157–1182 (2003)
8. Cho, B., Yu, H., Lee, J., Chee, Y., Kim, I., Kim, S.: Nonlinear support vector machine visualization for risk factor analysis using nomograms and localized radial basis function kernels. IEEE Transactions on Information Technology in Biomedicine 12, 247–256 (2008)
9. Vapnik, V.N.: Statistical Learning Theory. John Wiley and Sons, New York (1998)
10. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2, 121–167 (1998)

11. Platt, J.C.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Advances in Large Margin Classifiers, pp. 61–74. MIT Press, Cambridge (1999)
12. Chang, C.C., Lin, C.J.: Libsvm: a library for support vector machines (2001), software http://www.csie.ntu.edu.tw/~cjlin/libsvm
13. Vien, N.A., Viet, N.H., Chung, T., Yu, H., Kim, S., Cho, B.H.: Vrifa: a nonlinear svm visualization tool using nomogram and localized radial basis function (lrbf) kernels. In: CIKM, pp. 2081–2082 (2009)
14. Liu, T.Y., Xu, J., Qin, T., Xiong, W., Li, H.: Letor: Benchmark dataset for research on learning to rank for information retrieval. In: The Learning to Rank workshop in the 30th annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR (2007)

New Multi-class Classification Method Based on the SVDD Model

Lei Yang[1*], Wei-Min Ma[2], and Bo Tian[3]

[1] Shanghai Key Laboratory of Power Station Automation Technology,
School of Mechatronics Engineering and Automation,
Shanghai University, Shanghai 200072, China
[2] School of Economics and Management, Tongji University,
Si-Ping Road 1239, Shanghai, 200092, China
[3] School of Information Management and Engineering,
Shanghai University of Finance Economic, Guo-ding Road 777,
Shanghai, 200433, China

Abstract. New decision-making function for multi-class support vector domain description (SVDD) classifier using the conception of attraction force was proposed in this paper. As for multi-class classification problems, multiple optimized hyperspheres which described each class of dataset were constructed separately similar with in the preliminary SVDD. Then new decision-making function was proposed based on the parameters of the multi-class SVDD model with the conception of attraction force. Experimental results showed that the proposed decision-making function for multi-class SVDD classifier is more accurate than the decision-making function using local density degree.

Keywords: Multi-class classification, support vector domain description model, decision-making function.

1 Introduction

Classical methods of pattern classification mainly include statistical testing methods (Kay, 1998; Nanda, 2001). In statistical methods, numbers of samples are usually assumed to be sufficiently large, and samples are assumed to be some known distribution. But samples are usually finite even deficient in practice, and distributions of samples are unknown. So new pattern classification methods such as neural networks, clustering method, support vector machine (SVM) are proposed in recent years (Bernhard, 1997; Gomez-Skarmeta, 1999; Martens, 2008; Pendharkar,2002). Data-driven classification methods are based on statistical learning theories, and the disadvantages of statistical asymptotic assumption in classical statistical methods can be tided over. Minimization of experimental risk is used in neural networks predication method (Pendharkar, 2002). But several disadvantages of neural networks such as over-fitting phenomenon in learning process, lack of generalization ability, and local extremum values limit their practical applications (Cortes, 1995; Vapnik, 1998).

* Corresponding author, yangyoungya@gmail.com

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 103–112, 2011.

The complexity of models and experimental risk can be balanced effectively in SVM, and generalization ability of model is improved. Different improved SVM models have been investigated by researchers. Suykens proposed least squares SVM (Suykens, 1999). Zhang proposed wavelet SVM (Zhang, 2004). Doumpos proposed additive SVM (Doumpos, 2007). Jayadeva proposed Twin SVM (Jayadeva, 2007). Tax proposed support vector domain description (SVDD) model (Tax, 1999; Tax, 2004). The SVDD model is mainly used to deal with the problem of one-class classification. And SVDD models can be used to describe dataset and detect outliers (Cho, 2009; Guo, 2009; Lee, 2005; Tax, 1999; Tax, 2004). Some extended SVDD models were proposed by other researchers. Zhang proposed fuzzy SVDD (Zhang, 2009). Lee used domain described support vector classifier to deal with multi-classification problem (Lee, 2007). Hao proposed multisphere method in SVM (Hao, 2007). And as for multi-classification problem, each class of samples can be described by an optimized hypersphere. Using this ideal, multi-class SVDD model can describe multiple classes of dataset. After multi-class SVDD model is established by training the known dataset, the radius and center of each hypersphere are achieved. As for a new sample, Mu used the nearest distance between it and the centers of hyperspheres to predict which class the new sample belongs to (Mu, 2009). This decision criterion only considered the center position of each class of dataset in feature space. Lee used density estimation method to construct a new decision-making function based on the squared radial distance decision criterion (Lee, 2007). His decision-making function considered the numbers of each class of samples. Inspired by the law of universal gravitation, new decision-making function was proposed in this paper. If each sample is assumed to be unit mass, the mass of each class of samples is proportion to the numbers of the class of samples and the volume of the hypersphere. And attraction force between the new sample and each class of samples is proportion to the mass of that class of samples and reciprocal proportion to the distance between the sample and the class of samples. When maximum attraction force was used as decision-making criterion, we proposed new decision-making function. The paper is organized as follows. The spirit of this paper is discussed in section one. The preliminary SVDD model to describe one class of dataset is analyzed in section two. And multi-class SVDD model is established in section three. New decision-making function based on parameters of multi-class SVDD model with the conception of attraction force is proposed in section four. Then experimental results are reported in section five. And conclusion is drawn in the last section.

2 The Preliminary SVDD Classifier

The objective of preliminary SVDD model is to describe target dataset by using hypersphere with minimized radius R in feature space (Tax, 2004).. And target samples are located in the optimized hypersphere. Let dataset $\{x_i, i = 1, 2, ..., N\}$ be training samples. The mathematic form of the model is minimizing the function $F(R, \mathbf{a}) = R^2$ with the constraint condition $\|x_i - \mathbf{a}\|^2 \leq R^2$, and radius R and center $\mathbf{a}$ of the hypersphere is computed by optimal conditions. The SVDD model to describe one class of samples can be sued to classify of samples of two classes.

Consider dataset $\{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$ come from two different classes of samples, where N is the number of samples, and x_i is the i th sample, $y_i = 1$ or -1, $i = 1, 2, ..., N$. Not losing generality, for samples x_i, $i = 1, 2, ..., l$, let $y_i = 1$, and for samples $x_i, i = l+1, l+2, ..., N$, let $y_i = -1$. In other words, $\{x_i, i = 1, 2, ..., l\}$ are positive samples, and $\{x_i, i = l+1, l+2, ..., N\}$ are negative samples or outliers. The positive samples $\{x_1, x_2 ..., x_l\}$ are assumed in hypersphere, and negative samples $\{x_{l+1}, x_{l+2} ..., x_N\}$ are outside of hypersphere. If errors are allowed for in both classes of samples, lack variables $\xi_i^+ \geq 0$, $(i = 1, 2, ..., l)$; $\xi_i^- \geq 0$, $(i = l+1, l+2, ..., N)$ are introduced in the objective function. The problem of minimizing the radius of the hypersphere can be formulated by the following quadratic programming with inequality constraints

$$\begin{cases} \min R^2 + C_1 \sum_{i=1}^{l} \xi_i^+ + C_2 \sum_{i=l+1}^{N} \xi_i^- \\ sub: \left\| x_i - \mathbf{a} \right\|^2 \leq R^2 + \xi_i^+, \xi_i^+ \geq 0, i = 1, 2, ...l; \\ \quad \left\| x_i - \mathbf{a} \right\|^2 \geq R^2 - \xi_i^-, \xi_i^- \geq 0, i = l+1, l+2, ...N. \end{cases} \tag{1}$$

where the positive constant parameters C_1 and C_2 are penalty factors. They control the trade-off between the radius of hypersphere and the error. Using Lagrange multipliers algorithm for Eq.(1), we can draw the corresponding Lagrange function as

$$\begin{aligned} L(R, \mathbf{a}, \alpha, \beta, \xi_i^+, \xi_i^-) &= R^2 + C_1 \sum_{i=1}^{l} \xi_i^+ + C_2 \sum_{i=l+1}^{N} \xi_i^- \\ &- \sum_{i=1}^{l} \beta_i \xi_i^+ - \sum_{i=l+1}^{N} \beta_i \xi_i^- - \sum_{i=1}^{l} \alpha_i (R^2 + \xi_i^+ - \left\| x_i - \mathbf{a} \right\|^2) \\ &- \sum_{i=l+1}^{N} \alpha_i (\left\| x_i - \mathbf{a} \right\|^2 + \xi_i^- - R^2) \end{aligned} \tag{2}$$

where $\alpha_i \geq 0, \beta_i \geq 0$ are Lagrange multipliers, $i = 1, 2, ..., N$. Lagrange function L should be minimized with respect to $R, \mathbf{a}, \xi_i^+, \xi_i^-$, and maximized with respect to α_i and β_i. The extremum conditions of Lagrange function L are

$$\frac{\partial L}{\partial R} = 0, \frac{\partial L}{\partial \mathbf{a}} = 0, \frac{\partial L}{\partial \xi_i^+} = 0, \frac{\partial L}{\partial \xi_i^-} = 0 \tag{3}$$

Resubstituting the solutions of Eq.(3) into Eq.(2) results in the dual form of the Lagrange optimization problem shown as following quadratic programming problem with inequality constraints

$$\begin{cases} \max \sum_{i=1}^{l} \alpha_i (x_i \cdot x_i) - \sum_{i=l+1}^{N} \alpha_i (x_i \cdot x_i) \\ \quad - \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j (x_i \cdot x_j) + 2 \sum_{i=1}^{l} \sum_{j=l+1}^{N} \alpha_i \alpha_j (x_i \cdot x_j) - \sum_{i=l+1}^{N} \sum_{j=l+1}^{N} \alpha_i \alpha_j (x_i \cdot x_j) \\ sub: \sum_{i=1}^{l} \alpha_i - \sum_{i=l+1}^{N} \alpha_i = 1, \\ \quad 0 \le \alpha_i \le C_1, i = 1, 2, ..., l; \quad 0 \le \alpha_i \le C_2, \quad i = l+1, l+2, ..., N. \end{cases} \tag{4}$$

Let $\alpha_i' = y_i \alpha_i$, then we have

$$\sum_{i=1}^{N} \alpha_i' = 1 \tag{5}$$

and

$$\mathbf{a} = \sum_{i=1}^{N} \alpha_i' x_i \tag{6}$$

Then Eq.(4) can be simplified as

$$\begin{cases} \max \sum_{i=1}^{N} \alpha_i' (x_i \cdot x_i) - \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i' \alpha_j' (x_i \cdot x_j) \\ sub: \sum_{i=1}^{N} \alpha_i' = 1, \ 0 \le \alpha_i \le C_1, i = 1, 2, ..., l; \\ \quad 0 \le \alpha_i \le C_2, \ i = l+1, l+2, ..., N. \end{cases} \tag{7}$$

The quadratic programming problem containing inequality constraints denoted as Eq.(7) can be solved using simple iterative multiplicative updating algorithm (Sha, 2007). Then radius R and center $\mathbf{a}$ of the hypersphere are solved. And decision-making function of SVDD to classify a new sample can be constructed. If a new sample x is in the hypersphere, it belongs to the positive class. Otherwise it belongs to the negative one. So the decision function can be shown as

$$y(x) = \mathrm{sgn}(R^2 - ((x \cdot x) - 2 \sum_{i=1}^{N} \alpha_i' (x_i \cdot x) + \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i' \alpha_j' (x_i \cdot x_j))) \tag{8}$$

In order to determine the decision function, R and center $\mathbf{a}$ of the hypersphere should be computed. In practice, only part of parameters α_i are non-zero. The samples on the boundary of the hypersphere are support vectors. They determine radius R and center $\mathbf{a}$ of the hypersphere. We assume that α_k is corresponding to some support vector x_k. Then radius R of the hypersphere can be calculated as

$$R = ((x_k \cdot x_k) - 2\sum_{i=1}^{N} \alpha_i' (x_k \cdot x_k) + \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i' \alpha_j' (x_i \cdot x_j)))^{\frac{1}{2}} . \tag{9}$$

And center $\mathbf{a}$ of the hypersphere can also be solved. If the inner product $x_i \cdot x_j$ of x_i and x_j is substituted by kernel function $K(x_i, x_j)$, the decision function is shown as

$$y(x) = \operatorname{sgn}(R^2 - (K(x,x) - 2\sum_{i=1}^{N} \alpha_i' K(x_i, x) + \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i' \alpha_j' K(x_i, x_j))) . \tag{10}$$

Kernel functions are usually constructed by mapping function which satisfied positive kernel conditions (Vapnik, 1998).

3 Multi-class SVDD Model

Similar with binary SVDD model, sample dataset $D_k = \{x_1^{(k)}, x_2^{(k)}, .., x_{N_k}^{(k)}\}$ come from k th class of samples, where $k = 1, 2, ..., K,$ and K denotes the number of classes, N_k is the number of the k th classes of samples. In multi-class SVDD model, each optimized hypersphere $\{R_k, \mathbf{a}_k\}$ is constructed to describe each class of sample dataset $D_k, k = 1, 2, ..., K$. And other classes of samples are out of this hypersphere. Slack variables $\xi_i^{(k)} \geq 0,\ (i = 1, 2, ..., N_k)$ are introduced in objective function for each sample similar with in the preliminary SVDD model because errors are allowed for. So the problem of minimizing radius of K hyperspheres of multi-class SVDD model can be formulated by the following quadratic programming with inequality constraints

$$\begin{cases} \min \sum_{k=1}^{K} R_k^2 + \sum_{k=1}^{K} (C_k \sum_{i=1}^{N_k} \xi_i^{(k)}) \\ sub: \left\| x_i^{(k)} - \mathbf{a}_k \right\|^2 \leq R_k^2 + \xi_i^{(k)}, \\ \quad \xi_i^{(k)} \geq 0, i = 1, 2, ..., N_k, k = 1, 2, ..., K. \end{cases} \tag{11}$$

where positive constant parameters C_k are penalty factors. Using the Lagrange multipliers algorithm, we can draw the corresponding Lagrange function as

$$
\begin{aligned}
&L(R_1, R_2, ..., R_K; \mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_K; \xi_1^{(k)}, \xi_2^{(k)}, ..., \xi_{N_k}^{(k)}; \alpha_1^{(k)}, \alpha_2^{(k)}, ..., \alpha_{N_k}^{(k)}; \\
&\beta_1^{(k)}, \beta_2^{(k)}, ..., \beta_{N_k}^{(k)}) \\
&= \sum_{k=1}^{K} R_k^2 + \sum_{k=1}^{K} (C_k \sum_{i=1}^{N_k} \xi_i^{(k)}) \\
&\quad - \sum_{k=1}^{K} \sum_{i=1}^{N_k} \alpha_i^{(k)} (R_k^2 + \xi_i^{(k)} - \left\| x_i - \mathbf{a}_k \right\|^2) - \sum_{k=1}^{K} \sum_{i=1}^{N_k} \beta_i^{(k)} \xi_i^{(k)}
\end{aligned}
\tag{12}
$$

where $\alpha_1^{(k)}, \alpha_2^{(k)}, ..., \alpha_{N_k}^{(k)}$ and $\beta_1^{(k)}, \beta_2^{(k)}, ..., \beta_{N_k}^{(k)}$ are Lagrange multipliers. Lagrange function L should be minimized with respect to $R_1, R_2, ..., R_K$; $\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_K$; $\xi_1^{(k)}, \xi_2^{(k)}, ..., \xi_{N_k}^{(k)}$, and maximized with respect to $\alpha_1^{(k)}, \alpha_2^{(k)}, ..., \alpha_{N_k}^{(k)}$ and $\beta_1^{(k)}, \beta_2^{(k)}, ..., \beta_{N_k}^{(k)}$. The extremum conditions of Lagrange function L give the following formula

$$
\frac{\partial L}{\partial R_k} = 0; \frac{\partial L}{\partial \mathbf{a}_k} = 0; \frac{\partial L}{\partial \xi_i^{(k)}} = 0, k = 1, 2, ..., K.
\tag{13}
$$

Resubstituting the solutions of Eq.(13) into Eq.(12), the dual form of the Lagrange optimization problem is shown as following quadratic programming problem containing inequality constraints

$$
\begin{cases}
\max \sum_{k=1}^{K} (\sum_{i=1}^{N_k} \alpha_i^{(k)} (x_i^{(k)} \cdot x_i^{(k)}) - \sum_{i=1}^{N_k} \sum_{j=1}^{N_k} \alpha_i^{(k)} \alpha_j^{(k)} (x_i^{(k)} \cdot x_j^{(k)})) \\
sub: \sum_{i=1}^{N_k} \alpha_i^{(k)} = 1;\ 0 \le \alpha_i^{(k)} \le C_k,\ i = 1, 2, ..., N_k, k = 1, 2, ..., K.
\end{cases}
\tag{14}
$$

We noticed that parameters set $\{\alpha_1^{(k)}, \alpha_2^{(k)}, ..., \alpha_{N_k}^{(k)}\}$ can be solved separately in Eq.(14). The optimization problem Eq.(14) can be separated into K SVDDs with one-class samples such as Eq.(7). After computing parameters $\{\alpha_1^{(k)}, \alpha_2^{(k)}, ..., \alpha_{N_k}^{(k)}\}$, radius R_k and centers $\mathbf{a}_k$ of each hypersphere are achieved. For example, let non-zero $\alpha^{(k)}$ is corresponding to some support vector $x^{(k)}$, R_k and centers $\mathbf{a}_k$ can be calculated.

4 New Decision-Making Function for Multi-class SVDD Model

Dataset of one class was described by an optimized hypersphere in SVDD model. And as for multi-classification problem, we describe multiple classes of dataset by using multiple optimized hyperspheres. Several researchers have proposed different decision-making function for multi-class model based on different theories.

After multi-class SVDD model is established by training the known dataset, the radiuses R_k and centers $\mathbf{a}_k$ of each hypersphere are achieved, $k = 1, 2, ..., K$. Then as for a new sample x, the nearest distance between it and the centers of hyperspheres is used to predict which class the new sample belongs to (Mu, 2009). The decision-making function is shown as

$$y(x) = \arg \min_{k=1,2,...,K} \left\| x - \mathbf{a}_k \right\| . \tag{15}$$

This decision function only considers the center position of each class of samples. Lee used density estimation method to construct an improved decision-making function based on the squared radial distance decision criterion (Lee, 2007). The decision-making function with the conception of local density degree is shown as

$$y(x) = \arg \max_{k=1,2,...,K} N_k * (R_k^2 - \left\| x - \mathbf{a}_k \right\|^2) . \tag{16}$$

This decision-making function considers the numbers of each class of dataset. But information about volume of the hypersphere is not included. The law of universal gravitation sees that attraction force between two objects is proportion to their masses and reciprocal proportion to their squared distance. Let each sample be unit mass such as $m = 1$, and we assume that the mass M_k of each class of sample datasets D_k is proportion to the numbers of the class of samples and the volume of the hypersphere such as $\mathrm{M}_k = N_k V_k$, where V_k is volume of the hypersphere in d dimension feature space. We know that volume of the hypersphere in d dimension space is $V = cR^d$,where c is constant, and R is radius of the hypersphere. So attraction force between the new sample x $(x \neq \mathbf{a}_k)$ and each class of samples D_k can be simulated as

$$f_k = \frac{m\mathrm{M}_k}{\left\| x - \mathbf{a}_k \right\|^2} = \frac{cN_k R_k^d}{\left\| x - \mathbf{a}_k \right\|^2} . \tag{17}$$

When maximum attraction force is used as decision-making criterion, new decision-making function based on parameters of the multi-class VDD model with the conception of attraction force can be shown as

$$y(x) = \arg \max_{k=1,2,...,K} \frac{N_k R_k^d}{\left\| x - \mathbf{a}_k \right\|^2} . \tag{18}$$

From Eq.(18) we see that if the effects $R^d / \|x-\mathbf{a}\|^2$ of two classes of samples are equal, the new sample x belongs to the class with larger number of the samples.

This is consistent with the density estimation thehod(Lee, 2007). If the effects $N_k / \|x-\mathbf{a}\|^2$ of two classes of samples are equal, the new sample x belongs to the class with larger volume of hypersphere. We also notice if the effects of volume of the hypersphere V_k and the number of the samples N_k are not considered, Eq.(18) will be equivalent with Eq.(15).

5 Experimental Results

Several groups of experiments on different datasets have been performed. Follows report experimental results on some classical datasets. Table 1 lists these datasets. The first dataset is artificial dataset produced by four Gauss distribution functions in two dimension space (Heijden, 2004). The second dataset is artificial dataset produced by using three-spiral functions. The third datasets named Heart and Sonar datasets are from the UCI Machine Learning Repository (Asuncion, 2007). Each dataset is divided into training subset and predicting subset.

Table 1. Description of datasets used in experiments

No.	Name	Classes	Dimension	Total samples	No. of training	No. of predicting
1	Four-Gauss	4	2	600	300	200
2	Three-spiral	3	2	300	150	150
3	Iris	3	4	150	90	60

We compare the improvement on the accuracy of predication of the proposed decision-making function Eq.(18) over the decision-making functions Eq.(15) and Eq.(16). Because same numerical algorithm named iterative multiplicative updating algorithm is used to compute the parameters of models in training stages, and dimensions of matrixes used in the algorithm are the numbers of samples, running times of algorithms are equal approximately. So running-times are not listed. Gauss function $K(x, y) = \exp(-\|x-y\|^2 / \delta)$ is used as kernel function. Experiments with different penalty factor parameter and different parameter δ of kernel function have been performed.

Table two shows a group of average correct predicting rate (ACPR) using the preliminary-SVDD (P-SVDD) and multi-class SVDD (M-SVDD) models together with different decision-making function when parameters (C, δ) are $(0.25,1)$ after ten times experiments. In table two, ACPR is the average ratio of numbers of correct classified samples to numbers of the total predicated samples after ten times

experimens.Table two illustrates that the proposed decision-making function Eq.(18) improves ACPR compared with decision-making function Eq.(15) and Eq(16).

Table 2. ACPR of multiple classification problem(%)

Dataset	M-SVDD+Eq.(25)	M-SVDD+Eq.(26)	M-SVDD+Eq.(30)
Four-Gauss	93.85	94.30	**96.05**
Three-spiral	88.67	91.20	**94.06**
Iris	92.0	91.83	**94.50**

6 Conclusion

As for multi-class classification problems, multiple optimized hyperspheres which describe each class of dataset are constructed separately, and multi-class SVDD model is established. Then new decision-making function is proposed based on parameters of the multi-class SVDD model in which effects of three aspects named center positions of hyperspheres, volumes of hyperspheres and numbers of the samples are all considered. Experimental results show that the proposed decision-making function for multi-class SVDD classifier is more accurate than the decision-making function using density estimation method under same experiment conditions.

Acknowledgments. The work was supported by the National Natural Science Foundation of China (61005015), the third National Post-Doctoral special Foundation of China (201003280), National Post-Doctoral Foundation of China (20100470108), and Shanghai University 11th Five-Year Plan 211 Construction Projection.

References

1. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA (2007), http://www.ics.uci.edu/~mlearn/MLRepository.html
2. Bernhard, S., Sung, K.K.: Comparing support vector machines with Gaussian kernels to radical basis function classifiers. IEEE Transaction on Signal Processing 45(11), 2758–2765 (1997)
3. Cho, H.W.: Data description and noise filtering based detection with its application and performance comparison. Expert Systems with Applications 36(1), 434–441 (2009)
4. Cortes, C., Vapnik, V.: Support vector networks. Machine Learning 20(3), 273–297 (1995)
5. Doumpos, M., Zopounidis, C., Golfinopoulou, V.: Additive support vector machines for pattern classification. IEEE Trans on Systems, Man, and Cybernetics-Part B: Cybernetics 37(3), 540–550 (2007)
6. Gomez-Skarmeta, A.F., Delgado, M., Vila, M.A.: About the use of fuzzy clustering techniques for fuzzy model identification. Fuzzy Sets and Systems 106(2), 179–188 (1999)
7. Guo, S.M., Chen, L.C., Tsai, J.S.H.: A boundary method for outlier detection based on support vector domain description. Pattern Recognition 42(1), 77–83 (2009)

8. Hao, P.Y., Lin, Y.H.: A new multi-class support vector machine with multi-sphere in the feature space. In: Okuno, H.G., Ali, M. (eds.) IEA/AIE 2007. LNCS (LNAI), vol. 4570, pp. 756–765. Springer, Heidelberg (2007)
9. Heijden, F.R., Duin, R., de Ridder, D., Tax, D.M.J.: Classification, Parameter Estimation and State Estimation: An Engineering Approach Using MATLAB. John Wiley and Sons, Chichester (2004)
10. Jayadeva, K.R., Chandra, S.: Twin support vector machines for pattern classification. IEEE Trans. on Pattern Anal. Machine Intell. 29(5), 905–910 (2007)
11. Kay, S.M.: Fundamentals of statistical signal processing. Volume I: Estimation theory/Volume II: Detection theory. Prentice-Hall, New Jersey (1998)
12. Lee, D., Lee, J.: Domain described support vector classifier for multi-classification problems. Pattern Recognition 40(1), 41–51 (2007)
13. Lee, K.Y., Kim, D.W., Lee, D.: Improving support vector data description using local density degree. Pattern Recognition 38, 1768–1771 (2005)
14. Martens, D., Huysmans, J., Setiono, R.: Rule extraction from support vector machines: an overview of issues and application in credit scoring. Studies in Computational Intelligence 80(1), 33–63 (2008)
15. Mu, T., Nandi, A.K.: Multiclass classification based on extended support vector data description. IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetics 39(5), 1206–1216 (2009)
16. Nanda, S., Pendharkar, P.C.: Development and comparison of analytical techniques for predicting insolvency risk. International Journal of Intelligent Systems in Accounting, Finance and Management 10(3), 155–168 (2001)
17. Pendharkar, P.C.: A computational study on the performance of ANNs under changing structural design and data distributions. European Journal of Operational Research 138(1), 155–177 (2002)
18. Sha, F., Lin, Y., Saul, L.K., Lee, D.D.: Multiplicative updates for nonnegative quadratic programming. Neural Computation 19(8), 2004–2031 (2007)
19. Suykens, J.A.K., Vandewalle, J.: Least squares support vector machine classifiers. Neural Proc. Lett. 9(3), 293–300 (1999)
20. Tax, D.M.J., Duin, R.P.W.: Support vector data description. Machine Learning 54(1), 45–66 (2004)
21. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1998)
22. Zhang, L., Zhou, W., Jiao, L.: Wavelet support vector machine. IEEE Trans. on Systems, Man, and Cybernetics-part B: Cybernetics 34(1), 34–39 (2004)
23. Zhang, Y., Chi, Z.X., Li, K.Q.: Fuzzy multi-class classifier based on support vector data description and improved PCM. Expert Systems with Applications 36(5), 8714–8718 (2009)

Intelligence Statistical Process Control in Cellular Manufacturing Based on SVM

Shaoxiong Wu

Dept. of Economics and Management, Fujian University of Technology
Fuzhou, Fujian 350108, P.R. China
wsx@fjut.edu.cn

Abstract. According to peculiarity of cellular manufacturing, the method of drawing control chart was proposal. In the modeling of structure for patterns recognition of control chart in cellular manufacturing, the mixture kernel function was proposed, and one-against-one algorithm multi-class classification support vector machine was applied, and genetic algorithm was used to optimize the parameters of SVM. The simulation results show that the performance of mixture kernel is superior to a single common kernel, and it can recognize each pattern of the control chart accurately, and it is superior to probabilistic neural network and wavelet probabilistic neural network in the aggregate classification rate, type I error, type II error, and also has such advantages as simple structure, quick convergence, which can be used in control chart patterns recognition in cellular manufacturing.

Keywords: Cellular Manufacturing; Support Vector Machines (SVM); Control Chart; Patterns Recognition.

1 Introduction

The control chart as the important tools of quality management and quality control, how is application on the cellular manufacturing, there are many problem which now confront us. The traditional statistical control chart was designed and applied in the production manner of large batch. But the production manner of middle or small batch in cellular manufacturing, it is difficult to apply traditional statistical control chart as it is short of quality character data. In the study of quality control in cellular manufacturing, Shahrukh A(1999) [1] suggested the general method of quality control. On the middle and small batch quality control, Salti M(1994)[2] reviewed the use of SPC in batch production, Jianwen W(2002)[3] Rui M(2005)[4] suggested the general method of quality control in middle and small batch. Fajun W(2002)[5] Chong X(2000)[6] discussed the applications of group technique in the control chart.

As to intelligence statistical process control, the key problem is how to recognize the patterns of control chart. In recent years, much attention has been paid to the development of control chart recognition which is based on neural networks. The key factors for their wide use in the field of control chart recognition are the properties they have. These properties are the ability of learning and generalizing, nonlinear

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 113–120, 2011.

mapping, parallelism of computation and vitality (Pharm, D. T. and E. Oztemel(1992)[7], Hwarng, H. B. and N. F. Hubele (1993) [8], Smith, A. E.(1994) [9], Pham, D. T. and E. Oztemel(1994) [10], Cheng C. S.(1995) [11], Ruey-shiang Guh. (2003) [12]).Because BP neural network has the characteristics of simple structure, stabilized working attitude, strong and easy to realize generalizing ability, early research has been concentrated on the structure of BP neural network mainly. Nevertheless, BP neural network also has such characteristics as slow training, more time consuming and worse adaptability. So some researchers had to recognize the control chart patters by improving the BP neural network or other neural networks (Cheng S. I. and C. A. Aw. (1996) [13], Cheng C. S. (1997) [14], AI-Ghanim, A. (1997) [15], A. S. Anagun (1998) [16], Li Mengqing and Chen Zhixiang(2000) [17], Le Qinghong, Gao Xinghai(2004) [18]).

Support vector machines (SVM) which introduced by Vapnik [19,20] is widely used to classification and nonlinear function estimation, such as pattern recognize and regression analysis and feature abstraction. In this paper, a method of SVM was presented, and it was used to pattern recognition of control chart in cellular manufacturing.

2 Statistical Process Control in Cellular Manufacturing

2.1 $\overline{\overline{X}}$ Control Chart in Cellular Manufacturing

Supposing the process of cellular manufacturing produced K kind of different specification parts for some time, it should be taken $n \times m_k (k = 1,2,\cdots,K)$ parts from k^{th} product batch. On the other hand, the number of time node is m_k in each product batch, and it was taken number n parts in series. The k^{th} product batch sample data shows as following

$$\begin{pmatrix} x_{11}^{(k)} & x_{12}^{(k)} & \cdots & x_{1n}^{(k)} \\ x_{21}^{(k)} & x_{22}^{(k)} & \cdots & x_{2n}^{(k)} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m_k 1}^{(k)} & x_{m_k 2}^{(k)} & \cdots & x_{m_k n}^{(k)} \end{pmatrix}$$

Supposing each product batch has same distribution, that is normal distribution, there is $x_{ij}^{(k)} \sim N(\mu_k, \sigma_k^2)$.

In order to draw control chart in one chart with the different distribution quality data which obtained from process, normalization method should be used to standard each group data in their own way. When it is down, the every group has same scale space, and yield $N(0,1)$ distribution.

$$y_{ij}^{(k)} = \frac{x_{ij}^{(k)} - \mu_k}{\sigma_k} \tag{1}$$

Then $y_{ij}^{(k)} \sim N(0,1)$

Where μ_k, σ_k^2 are unknown parameter, its estimator can be computed by (2) and (3)

$$\hat{\mu}_k = \bar{x}^{(k)} = \frac{1}{nm_k} \sum_{i=1}^{n} \sum_{j=1}^{m_k} x_{ij}^{(k)} \tag{2}$$

$$\hat{\sigma}_k^2 = S_k^2 = \frac{1}{n(m_k - 1)} \sum_{i=1}^{n} \sum_{j=1}^{m_k} x_{ij}^{(k)} \tag{3}$$

2.2 Control Chart Patterns and Its Description

The control chart patterns for this research can be divided into six types: one is normal pattern, other are the abnormal pattern, which are included in cyclic pattern, increasing trend, decreasing trend, upward shift and downward shift. This is illustrated in Fig. 1.

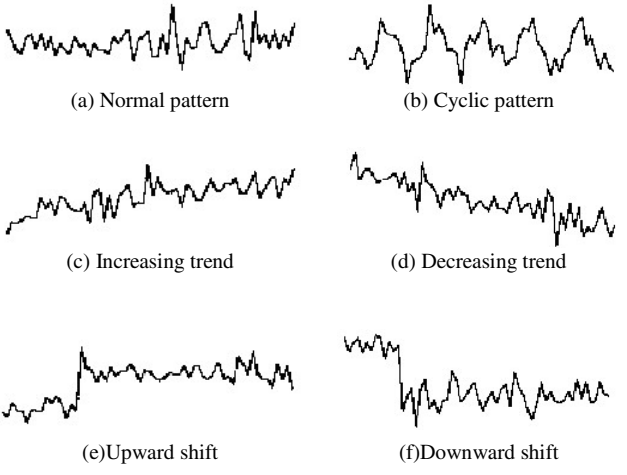


Fig. 1. Control chart patterns

The mixed patterns are mixed by two or more different abnormal pattern. It was divided into eight mixed patterns for this research: increasing trend & cyclic pattern, decreasing trend & cyclic pattern, upward shift & cyclic pattern, downward shift & cyclic pattern, upward shift & increasing trend, upward shift & decreasing trend, downward shift & increasing trend, downward shift & decreasing trend.

The following expressions is used to generate the training and testing data sets, which is expressed in a general form and includes the process and two noise components:

$$Y(t)=\mu+r(t)+S(t) \tag{4}$$

Where *Y(t)* is the sample value at time *t*, μ is the mean value of the process variable being monitored, *r(t)* is a random normal noise or variation, and *S(t)* is a special disturbance due to some assignable causes.

1) Normal patterns: S(t)=0
2) Cyclic patterns: S(t)= $a\sin(2\pi t/T)$

Where *a* is amplitude of cyclic variations, *T* is period of a cycle.

3) Increasing or decreasing trends: S(t)= ±gt

Where *g* is magnitude of the gradient of the trend, if S(t)>0, it expresses increasing trends, otherwise it is decreasing trends.

4) Upward or downward shifts: S(t)= ±ks

Where *k* is parameter determining the shift position, *s* is magnitude of the shift, if *S(t)>0*, it expresses upward trend, otherwise it is downward trend.

5) Increasing or decreasing trends & cyclic pattern: S(t)= ±gt +asin(2πt/T).
6) Upward or downward shift & cyclic pattern: S(t)= ±ks +asin(2πt/T).
7) Increasing or decreasing trends & shift: S(t)= ±gt±ks.

3 Basic Method of Control Chart Patterns Recognition

3.1 Basic Flow of Control Chart Patterns Recognition

The general framework is illustrated in Fig. 2. The steps are as follows:

1) Generated training data by formula (4).
2) Treated the training data using normalization method.
3) Inputted training data to train the M-SVMs
4) Obtained the process data
5) Pretreated the training data using normalization method.
6) Fed the pretreated process data to M-SVMs for control chart patterns recognition.

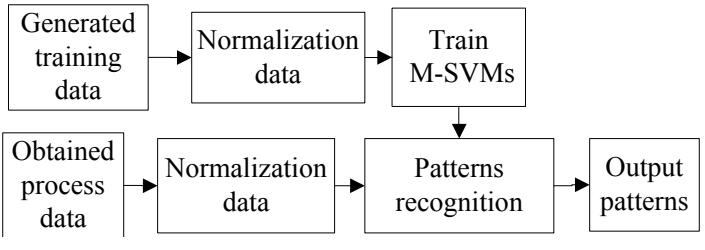


Fig. 2. Basic flow of control chart patterns recognition

3.2 Patterns Recognition Algorithm Choice

Support vector machines (SVMs), which were originally designed for binary classifications, are an excellent tool for machine learning. For the multiclass classifications, they

are usually converted into binary ones before they can be used to classify the examples. Among the existing multi-class SVM methods, such as one-against-all, one-against-one, directed acyclic graph, est., the one-against-one method is one of the most suitable methods for practical use. In this paper, the method of one-against-one algorithm was applied for multiclass classifications of control chart patterns recognition.

3.3 Kernel Function Choice

SVM is a kernel based approach, which allows the use of polynomial and RBF kernels and others that satisfy Mercer's condition [19,20].

(1) Polynomial kernel

$$K(x_i, x_j) = [(x_i \bullet x_j) + 1]^q \tag{5}$$

(2) RBF kernel

$$K(x_i, x_j) = \exp\{-\frac{\left|x_i - x_j\right|^2}{\delta^2}\} \tag{6}$$

(3) Perceptron kernel

$$K(x_i, x_j) = \tanh(v(x_i \bullet x_j) + c) \tag{7}$$

Polynomial kernel (a global kernel) shows better extrapolation abilities at lower orders of the degrees, but requires higher orders of degrees for good interpolation. On the other hand, the RBF kernel (a local kernel) has good interpolation abilities, but fails to provide longer range extrapolation. Preferably one wants to combine the 'good' characteristics of two kernels. Therefore, we will investigate whether the advantages of polynomial and RBF kernels can be combined by using mixtures [21]. There are several ways of mixing kernels. What is important though, is that the resulting kernel must be an admissible kernel. One way to guarantee that the mixed kernel is admissible, is to use a convex combination of the two kernels: $K_{mix} = \alpha K_{poly} + (1-\alpha) K_{gaussan}$, where the optimal mixing coefficient α has to be determined. The value of α is a constant scalar $\alpha \in (0,1)$. The mixed kernel, which is satisfied Mercer's condition, can be use for SVM, and it is shown that using mixtures of kernels can result in having both good interpolation and extrapolation abilities.

4 Simulation and Results

4.1 Training Sample and Testing Sample

In this work, the number of input data contained 32 point used to input data from 32 consecutive sample data point in a control chart. The training sample and test sample was generated by formula (4), we takes u=0 and σ=1. Each pattern of training sample was generated 100, and total numbers were 1400. Each pattern of testing sample was generated 50, and total numbers were 700. The generated training and testing samples was normalized by formula (1), and it was the input vectors of M-SVMs.

The output layer consists of 14 patterns, each one used for one of control chart patterns. The out value shows in Tab. 1.

Table 1. Goal value of SVM output layer

Pattern		Output value
Pattern 1	Normal	1
Pattern 2	Cyclic pattern	2
Pattern 3	Increasing trend	3
Pattern 4	Decreasing trend	4
Pattern 5	Upward shift	5
Pattern 6	Downward shift	6
Pattern 7	Increasing+cyclic	7
Pattern 8	Decreasing+cyclic	8
Pattern 9	Upward+cyclic	9
Pattern 10	Downward+cyclic	10
Pattern 11	Upward+increasing	11
Pattern 12	Upward+decreasing	12
Pattern 13	Downward+increasing	13
Pattern 14	Downward+decreasing	14

4.2 SVM Parameters Optimal

In order to optimal parameter (C, α, q, δ^2) of mixed-kernel function SVM, the method of the real-code genetic algorithm was used. The optimal results is (2000, 0.08, 1,0.6).

4.3 Simulation Results

4.3.1 Compare Recognition Results with Single Kernel Function

The several models will be built to test their patterns recognition performance. One model was mixed-kernel function with the parameter(C=2000, α=0.08, q=1, δ^2=0.6). The next was RBF kernel function with the parameter(C=2000, δ^2=0.6). The third was polynomial kernel function with the parameter(C=2000, q=1). Then we can compare their performance in aggregate recognition rate, type I error rate and type II error rate.

The test results shows in Tab.2.

Table 2. Compare recognition results with single kernel function

Kernel function	type I error rate (%)	type II error rate (%)	aggregate recognition rate (%)	Recongition rates(%) Pattern													
				1	2	3	4	5	6	7	8	9	10	11	12	13	14
polynomial	8	0	97.85	92	100	96	98	100	98	100	100	100	100	100	96	98	100
RBF	6	0	98.57	94	100	98	98	100	98	100	100	100	100	100	94	98	100
mixed	0	0	99.71	100	100	100	100	100	100	100	100	100	100	100	98	100	100

From the patterns recognition results, the aggregate recognition rate and each patterns recognition rate in mixed-kernel function are obviously highe than those in

RBF kernel function and polynomial kernel function, and the type I error rate in mixed-kernel function are obviously lower than those in RBF kernel function and polynomial kernel function. It shows the method of mixed-kernel function have many advantages, such as high accuracy and reliability in control chart patterns recognition.

4.3.2 Compare Recongition Results with Neural Network

In order to compare with different models, three type models of control chart patterns recognition were designed.

Model 1 (PNN): applied probabilistic neural network to patterns recognition, that the input data was put immediately into probabilistic neural network for patterns recognition..

Model 2 (WPNN) : input data was decomposed to the 3rd lever by wavelet transform. The wavelet transform function was coif5. The features vector combined the approximations decomposed at the 3rd lever with energy of each lever's detail coefficient, and it was fed to PNN for patterns recognition.

Model 3(M-SVMs): applied M-SVMs to patterns recognition with the mixed-kernel function.

The test results shows in Tab.3.

Table 3. Each model test recognition results

Kernel function	type I error rate (%)	type II error rate (%)	aggregate recognition rate (%)	Recongition rates(%) Pattern 1	2	3	4	5	6	7	8	9	10	11	12	13	14
PNN	78	0	85.28	22	100	78	80	64	60	100	100	100	100	100	90	100	100
WPNN	4	0.15	98.86	96	100	100	94	100	100	100	100	100	100	100	94	100	100
M-SVMs	0	0	99.71	100	100	100	100	100	100	100	100	100	100	100	98	100	100

From the patterns recognition results, the aggregate recognition rate in WPNN ans M-SVMs are obviously highe than those in PNN. The aggregate recognition rate, type I error rate and type II error rate in M-SVMs are superior to those in WPNN.

5 Conclusions

In this work, the SVM with mixed-kernel function was applied in the pattern recognition of control chart in cellular manufacturing. The simulation results show it have many advantages, such as quicker training and better recognition performance than single kernel function SVM, ANN and PNN. From the simulation results, we can also come to the conclusion as follows:

(1) It is feasible that genetic algorithm is applied parameter optimal in the SVM.

(2) It is feasible that SVM with mixed-kernel function is applied in the patterns recognition of control chart in cellular manufacturing. The results shows it have high aggregate classification rate, low type I error and type II error.

Acknowledgement. This work is supported by Science Foundation of Fujian province, China (Project number: 2009J01309).

References

1. Shahrukh, A.: Handbook of cellular manufacturing system. John Wiley & Sons, New York (1999)
2. Salti, M.M., Statham, A.: A Review of the Literature on the Use of SPC in Batch Production. Quality and Reliability Engineering International 10, 49–61 (1994)
3. Jianwen, W.: Quality control map for producing variety of production in small batch. Application of Statistics and Management 21(4), 34–37 (2002)
4. Rui, M., Xiaoming, S., Shugang, L., Dong, Y.: Research on statistical process quality control based on low volume manufacturing. Computer Integrated Manufacturing Systems 11(11), 1633–1635 (2005)
5. Fajun, W., Linna, Z., Fengxia, Z.: Discussion on the applications of group technique in the control chart's modeling. Journal of Zhengzhou University (Engineering Science) 23(1), 59–61 (2002)
6. Chong, X., Yulin, M.: Flexible automation oriented group statistical quality control. High Technology Letters (8), 64–66 (2000)
7. Pharm, D.T., Oztemel, E.: Control chart pattern recognition using neural networks. J. Syst. Eng. 2, 256–262 (1992)
8. Hwarng, H.B., Hubele, N.F.: Back-propagation pattern recognizers for X control charts. Methodology and performance. Computers Ind. Engng. 24, 219–235 (1993)
9. Smith, A.E.: X-bar and R control chart interpretation using neural computing. Int. J. Prod. Res. 32, 309–320 (1994)
10. Pham, D.T., Oztemel, E.: Control chart pattern recognition using learning vector quantization networks. Int. J. Prod Res. 32, 721–729 (1994)
11. Cheng, C.S.: A multi-layer neural network model for detecting changes in the process mean. Computers Ind. Engng. 28, 51–61 (1995)
12. Guh, R.-s.: Intergrating Artificial Intelligence into On-line statistical Process Control. Quality and Reliability Engineering International 19, 1–20 (2003)
13. Cheng, S.I., Aw, C.A.: A neural fuzzy control chart for detecting and classifying process mean shift. Int. J. Prod. Res. 34, 2265–2278 (1996)
14. Cheng, C.S.: Aneural network approach for the analysis of control chart patterns. Int. J. Prod. Res. 35, 667–697 (1997)
15. AI-Ghanim, A.: An unsupervised learning neural algorithm for identifying process behavior on control charts and a comparison with supervised learning approaches. Computers Ind. Engng. 32, 627–639 (1997)
16. Anagun, A.S.: A Neural network applied to pattern recognition in statistical process control. Computers Ind. Engng. 35(1), 185–188 (1998)
17. Li, M., Chen, Z.: Asynthetical approach of fuzzy logic and neural network fortrend pattern recognition in control charts. J. Huazhong Uniiv. of Sci. & Tech. (5), 24–26 (2000)
18. Le, Q., Gao, X.: A new neural network adaptable to pattern recognition. Computer Engineering 30(17), 17–18 (2004)
19. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
20. Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)
21. Smits, G., Jordaan, E.: Improved SVM regression using mixtures of kernels. In: IJCNN (2002)

Morlet-RBF SVM Model for Medical Images Classification

Huiyan Jiang[1], Xiangying Liu[1], Lingbo Zhou[1],
Hiroshi Fujita[2], and Xiangrong Zhou[2]

[1] Software College, Northeastern University, Shenyang, 110819, China
[2]Graduate School of Medicine, Gifu University,Yanagido, Gifu 501-1194, Japan
hyjiang@mail.neu.edu.cn

Abstract. Mapping way plays a significant role in Support Vector Machine (SVM). An appropriate mapping can make data distribution in higher dimensional space easily separable. In this paper Morlet-RBF kernel model is proposed. That is, Morlet wavelet kernel is firstly used to transform data, then Radial Basis Function (RBF)is used to map the already transformed data into another higher space. And particle swarm optimization (PSO) is applied to find best parameters in the new kernel. Morlet-RBF kernel is compared with Mexican-Hat wavelet kernel and RBF kernel. Experimental results show the feasibility and validity of this new mapping way in classification of medical images.

Keywords: Wavelet kernel, Morlet-Rbf kernel, PSO, Medical Images.

1 Introduction

Recently Support Vector Machine are widely used in pattern recognition[1][2][3]. Since wavelet technique is promise for classification[4], many researches are focused on combination of wavelet theory with SVM. H.Y.Liu applied Mexican-Hat wavelet to the kernel of SVM to classify analogue module of signals. M.H.Banki etc.[5] also used Mexican-Hat wavelet kernel SVM to classify hyperspectral images, and the result showed higher overall accuracy. L.Zhang, etc.[6] put forward Morlet wavelet kernel SVM and applied this model to the recognition of 1-D images of radar target. Their results verify that the training speed of the new SVM model is slightly faster than the Gaussian kernel SVM, at the same time, it has higher classification accuracy.

This paper is organized as follows. First, wavelet kernel is described. Second, Morlet-RBF SVM model is introduced in detail. Third, some experimental results are given. At last, conclusions are drawed.

2 Wavelet Kernel

Kernels admissible in SVM must satisfy Mercer's condition[5], which is showed in theorem 1,2 :

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 121–129, 2011.

Theorem 1: Any symmetric function $K(x,x^{'})$ in the input space can represent an inner product in feature space if

$$\iint K(x,x^{'})g(x)g(x^{'})dxdx^{'} \geq 0,\ \forall g \neq 0 \text{ for which } \int g^{2}(\xi)d\xi \prec \infty. \quad (1)$$

Where x and $x^{'}$ are feature vectors; g(x) is square-integrable function. Then $K(x,x^{'})$ can be written as

$$K(x,x^{'}) = \left\langle \phi(x) \cdot \phi(x^{1}) \right\rangle. \quad (2)$$

Theorem 2: function $K(x,x^{'})$ can be written as the form of $K(x-x^{'})$ if and only if its Fourier transform

$$F_{K}(w) = (2\pi)^{-N/2} \int \mathrm{e}^{(-j(w\cdot x))} K(x)dx \geq 0. \quad (3)$$

Two kinds of wavelet kernels must satisfy the condition in theorem 3.

Theorem 3: Let $\Psi(x)$ is a mother wavelet, and $x, x^{'} \in R^{N}$, then dot-product wavelet kernels are

$$K(x,x^{'}) = \prod_{i=1}^{N} \Psi(\frac{x_i - c_i}{a_i}) \Psi(\frac{x_i^{'} - c_i^{'}}{a_i}). \quad (4)$$

And translation-invariant wavelet kernels are

$$K(x,x^{'}) = \prod_{i=1}^{N} \Psi(\frac{x_i - x_i^{'}}{a_i}). \quad (5)$$

There are several kinds of wavelet SVM have been proposed. Li Zhang etc.[6] made use of Morlet wavelet function in the construction of kernel,

$$K(x_i,x_j) = \prod_{k=1}^{N} (\cos(1.75 \times \frac{(x_k^i - x_k^j)}{a_k}) \exp(-\frac{\left\| x_k^i - x_k^j \right\|^2}{2a_k^{\ 2}})). \quad (6)$$

Where x_k^i, x_k^j are elements of feature vectors x_i and x_j.

3 Morlet-Rbf Kernel Support Vector Machine

In recent years, wavelet transform is one of the most popular transformation techniques in signals processing and it has been successfully applied in signal approximation and classification[4][8].

A function $\Psi(x)$ is mother wavelet function if it is a square-integrable function and its Fourier transform $\Psi(w)$ satisfies,

$$\int_R \frac{\left|\Psi(w)^2\right|}{w} dw < \infty. \tag{7}$$

Wavelet base function $\Psi_{a,\tau}(x)$ is generated by dilation and translation of a mother wavelet function $\Psi(x)$,

$$\Psi_{a,\tau}(x) = \left|a\right|^{-\frac{1}{2}} \Psi(\frac{x-\tau}{a}), a \neq 0, a, \tau \in R. \tag{8}$$

Where a is dilation factor, τ is a factor of time.

As for a N dimensional vector, multidimensional mother wavelet function is shown as (9),

$$\Psi_n(x) = \prod_{i=1}^{N} \Psi(x_i). \tag{9}$$

We know that data in input space which is non-linear separable are mapped into a higher dimensional space by a mapping function and this function is hidden. Sometimes one time of mapping may not make data linearly separable or easily separable. So twice mapping can be used. But the two kernels should be similar so that the overall transform process is consistent. In the series of wavelet functions, Morlet and Gaussian wavelets are similar in distribution where they both have peaks and troughs as shown in Fig.1and Fig.2.

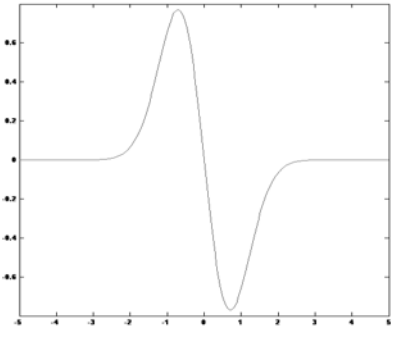


Fig.1. Morlet wavelet

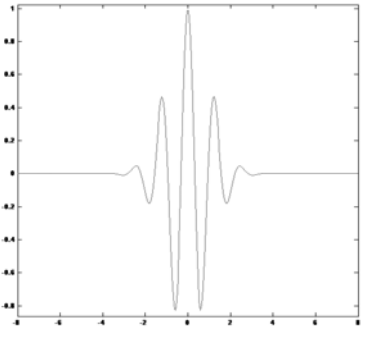


Fig.2. Gaussian wavelet

In this paper, Morlet wavelet kernel and Gaussian kernel are used to map data into higher feature space. K_1 and K_2 represent Morlet kernel and Gaussian kernel respectively. Since mapping function can't be explicitly expressed, but only in the form of

product, therefore K_1 is used firstly and K_2 is used secondly. Then quadratic programming problem is changed as follow:

$$\begin{aligned} &\min_{\alpha} \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j \tilde{K}(x_i, x_j)\alpha_i\alpha_j - \sum_{j=1}^{l}\alpha_j \\ &s.t.\ \sum_{i=1}^{l} y_i\alpha_i = 0,\ 0 \le \alpha_i \le C, i = 1,...,l. \end{aligned} \tag{10}$$

Where

$$\begin{aligned} \tilde{K}(x_i, x_j) &= \psi(\varphi(x_i))\cdot\psi(\varphi(x_j)) = K_2(\varphi(x_i), \varphi(x_j)) \\ &= \exp(-\gamma\left\|\varphi(x_i) - \varphi(x_j)\right\|^2) \\ &= \exp\{-\gamma[\varphi(x_i)\cdot\varphi(x_i) - 2\varphi(x_i)\cdot\varphi(x_j) \\ &\quad + \varphi(x_j)\cdot\varphi(x_j)]\} \\ &= \exp\{-\gamma[2\text{-}2\mathrm{K}_1(x_i, x_j)]\}. \end{aligned} \tag{11}$$

The discriminant function is

$$f(x) = \mathrm{sgn}(\sum_{i=1}^{nSV} y_i\alpha_i^* \tilde{K}(x_i, x) + b^*). \tag{12}$$

4 Evaluation Indexes for Classifier

There are several evaluation methods of classifier. Consider crossing matrix as shown in Table 1, No. of actual positive samples is P=TP+FN, No. of actual negative samples is N=FP+TN.

Table 1. Crossing matrix

Actual value	Predicted positive examples(+1)	Predicted negative examples(-1)
Positive examples(+1)	Correct positive examples(TP)	False negative examples(FN)
Negative examples(-1)	False positive examples(FP)	Correct negative examples(TN)

(1) Accuracy, it is defined by the proportion of correct predicted samples in total testing samples, which is computed as follow:

$$accuracy = \frac{TP + TN}{P + N}. \tag{13}$$

(2) Precision, it is the ratio of correct predicted positive samples to all the samples which are classified as positive as the follow formula:

$$precision = \frac{TP}{TP + FP}. \tag{14}$$

(3) Recall, it is the proportion of correct predicted positive sample in total positive samples:

$$recall = \frac{TP}{P}. \tag{15}$$

(4) F_1 value, it is the harmonic mean of precision and recall:

$$F_1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} \tag{16}$$

5 Experimental Results and Analysis

Our dataset come from library and [10], there are eight groups of data for binary classification and four groups of data for regression. Data for classification are liver cyst, diabetes, heart disease, liver cancer, fatty liver, breast cancer. And dimension of data feature vectors is 28, 8, 13, 6, 28, 10 respectively. The dataset for classification are randomly divided into two parts for generating training and testing data which are shown in Table 2. In this paper, parameters for kernel in SVM are selected by PSO. RBF-SVM, Mexican-Hat SVM and Morlet-RBF SVM are used to classify those data. The classification codes are programmed by C++ language, and the codes of drawing are programmed by Matlab. The whole program is run at the platform of Windows.

Table 2. No. of training and testing samples for classification

Group of samples	No. of training samples	No.of testing samples
Liver cyst-normal	53	24
Diabetes-normal	574	194
Heart disease-normal	170	100
Liver cancer-normal	92	41
Fatty liver-non fatty liver	79	21
Breast cancer-normal	263	34

The parameters C and σ for classification using RBF are shown in Table 3.

Table 3. Parameters for classification using RBF kernel

Group of samples	C	σ
Liver cyst-normal	100.0	0.70
Diabetes-normal	485.0	0.01
Heart disease-normal	112.5	0.01
Liver cancer-normal	200.0	0.80
Fatty liver-non fatty liver	200.0	0.002
Breast cancer-normal	714.162	2.167

Parameters C , σ and a for classification using Mexican-Hat SVM are listed in Table 4.

Table 4. Parameters for classification using Mexican-Hat kernel

Group of samples	C	σ	a
Liver cyst-normal	150.0	1.0	1.0
Diabetes-normal	1000.0	2.868	71.46
Heart disease-normal	1000.0	100.0	97.89
Liver cancer-normal	200.0	0.01	2.5
Fatty liver-non fatty liver	740.77	61.209	62.042
Breast cancer-normal	137.725	78.780	44.718

Parameters C , σ and a for classification using Morlet-RBF SVM are shown in Table 5.

Table 5. Parameters for classification using Morlet-RBF kernel

Group of samples	C	σ	a
Liver cyst-normal	840.0	2.384	37.59
Diabetes-normal	16.57	100.0	43.95
Heart disease-normal	0.1	100.0	8.80
Liver cancer-normal	913.397	22.939	31.038
Fatty liver-non fatty liver	423.17	1.658	86.415
Breast cancer-normal	13.893	15.783	90.220

Finally, the classification results are shown in Fig. 3-8.

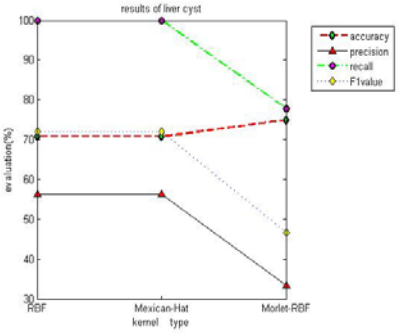


Fig. 3. Result of liver cyst

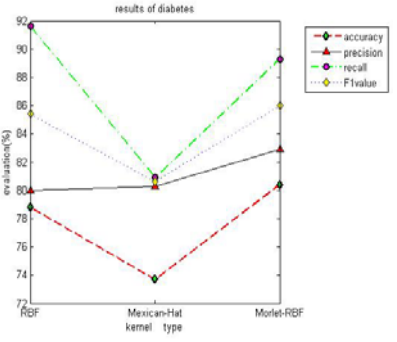


Fig. 4. Result of diabetes

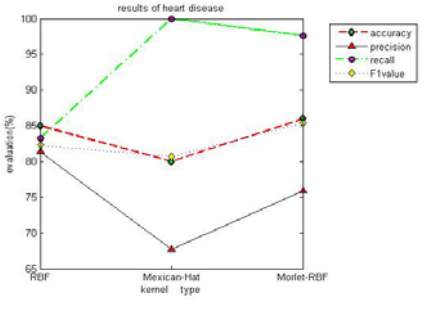


Fig. 5. Result of heart disease

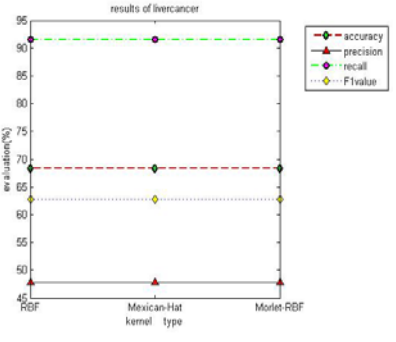


Fig. 6. Result of liver cancer

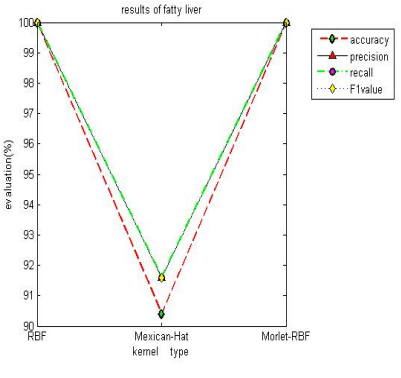


Fig. 7. Result of fatty liver

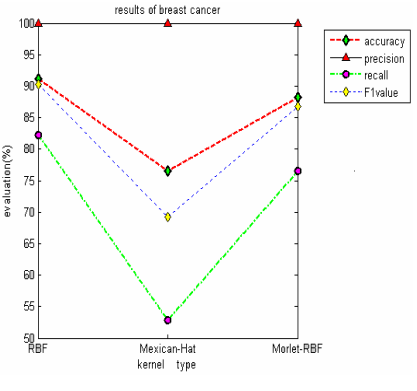


Fig. 8. Result of breast cancer

From the results, we can see that in Fig.3, Morlet-RBF has higher accuracy than the other two kernels, but lower in other evaluation indexes; In Fig.4, Mexican-Hat SVM does worst in accuracy, and Morlet-RBF has better result from all of assessment indexes. In Fig.5, Morlet-RBF has higher results than that of Mexican-Hat, but has lower precision than RBF SVM; In Fig.6, the three kernels have the same results; In Fig.7, the Mexican-Hat has the worst result, RBF and Morlet-RBF kernels have the same results. In Fig.8, Morlet-RBF is lower than RBF in each evaluation, but higher than Mexican-Hat model. By and large, Morlet-RBF gets better performance than the other two kernels. Except in the liver cyst data, where Morlet-RBF only achieves higher accuracy but lower precision, recall and F1 value, because many positive samples are false predicted. When one time of mapping can't get good result, the proposed two mapping model can be used to make data in higher dimension separated easily.

6 Conclusion

As discussed above, data in input space are mapped two times using Morel-RBF. This model of SVM has a overall better performance although there is lower result in an individual example. Moreover , PSO algorithm optimizes kernel parameters, avoiding triviality of random selection method. However, how to make PSO algorithm faster and how to apply Morlet-RBF to multiclass SVM are our research contents in the future.

Acknowledgment. This research is supported by the National Science Foundation of China (No: 60973071) and the Liaoning province Natural Science Foundation (No: 20092004).

References

1. Zhang, R., Ma, J.: An improved SVM method P-SVM for classification of remotely sensed data. Remote Sensing 29(20), 6029–6036 (2008)
2. Gletsos, M., Mougiakakou, S.G., Matsopoulos, G.K., Nikita, K.S., Nikita, A.S., Kelekis, D.: A computer-aided diagnostic system to characterize CT focal liver lesions: design and optimization of a neural network classifier. IEEE Transactions on Information Technology in Biomedicine 7(3), 153–162 (2003)
3. Xian, G.M.: An identification method of malignant and benign liver tumors from ultrasonography based on GLCM texture features and fuzzy SVM. Expert Systems With Applications 37(10), 6737–6741 (2010)

4. Szu, H.H., Telfer, B., Kadambe, S.: Neural network adaptive wavelets for signal representational and classification. Optical Engineering 31(9), 1907–1916 (1992)
5. Banki, M.H., Asghar Beheshti Sharazi, A.: NewKernel Function for Hyperspectral Image Classification. In: The 2nd International Conference on Computer and Automation Engineering, vol. 1, pp. 780–783 (2010)
6. Li, Z., Zhou, W.D., Jiao, L.C.: Wavelet Support Vector Machine. IEEE Transactions on Systems, Man and Cybernetics 34(1), 4–39 (2004)
7. Zhang, X.Y., Guo, Y.L.: Optimization of SVM Parameters Based on PSO Algorithm. In: Fifth International Conference on ICNC 2009, vol. 1, pp. 536–539 (2009)
8. Shioyama, T., Wu, H.Y., Nojima, T.: Recognition algorithm based on wavelet transform for handprinted Chinese characters. In: IEEE Fourteenth International Conference on Pattern Recognition, vol. 1, pp. 229–232 (1998)
9. Liu, H.Y., Sun, J.C.: A Modulation Type Recognition Method Using Wavelet Support Vector Machines. In: CISP Conference on Image and Signal Processing, pp. 1–4 (2009)
10. Information on http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary

COD Prediction for SBR Batch Processes Based on MKPCA and LSSVM Method

XiaoPing Guo and LiPing Fan

Information Engineering School, Shenyang University of Chemical Technology, Shenyang 110142, China
guoxiaoping@syuct.edu.cn

Abstract. Sequencing batch reactor (SBR) processes, a typical batch process, due to nonlinear and unavailability of direct on-line quality measurements, it is difficult for on-line quality control. A MKPCA-LSSVM quality prediction method is proposed for dedicating to reveal the nonlinearly relationship between process variables and final COD of effluent for SBR batch process. Three-way batch data of the SBR process are unfolded batch-wisely, and then nonlinear PCA is used to capture the nonlinear characteristics within the batch processes and obtain irrelevant variables of un-fold data as input of LS-SVM. Compared with the models of LS-SVM, the result obtained by the proposed quality prediction approach shows better estimation accuracy and is more extendable. The COD prediction of sewage disposing effluent quality can be helpful to optimal control of the wastewater treatment process, and it has some practical worthiness.

Keywords: quality prediction, batch process, MKPCA, LSSVM, SBR.

1 Introduction

Sequencing batch reactor (SBR) processes have demonstrated their efficiency and flexibility in the treatment of wastewaters with high concentrations of nutrients (nitrogen, phosphorous) and toxic compounds from domestic and industrial sources. However, due to the process high dimensionality, complexity, batch-to-batch variation, the final quality are usually available at the end of the batch, which is analysed (mostly offline) after the batch completion. It is difficult for on-line quality control[1~3].

Several statistical modeling methods namely, principal component analysis (PCA) and partial least squares (PLS), which perform dimensionality reduction and regression, respectively, are commonly used in batch process modeling and monitoring[4]. In industrial processes where severe nonlinear correlations exist among process variables, linear statistical techniques are not very effective in reducing the process data dimensions. If a linear PCA is used in these processes, a large number of PCs are required to explain sufficient data variance. For nonlinearly correlated data, the results from linear PCA may be inadequate because minor components can contain important information on nonlinearity. By discarding the

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 130–135, 2011.

minor components, this important information nonlinearity is lost. However, if these minor components are kept, the linear methods may require too much information to be useful. For the process quality prediction with nonlinearity, nonlinear statistical techniques are more appropriate.

The kernel principal component analysis (KPCA)is an emerging technique to address the nonlinear problems on the basis of PCA. The basic idea of KPCA is to map the input space into a feature space first via a nonlinear map and then to extract the principal component in that feature space. KPCA extends standard PCA to nonlinear data distributions. Using KPCA can capture the high-ordered nonlinear principal components in input data space[5~7].

In addition, a significant drawback of the PLS is that it is a linear regression formalism and thus makes poor predictions when relationships between process inputs and outputs are nonlinear. However, an artificial neural network (ANN) had the capability to handle the modeling problems associated with nonlinear static or dynamic behaviors. The NNPLS method [8]differs from the direct ANN approach in that, the input–output datas are not directly used to train the NN, but are preprocessed by the PLS outer transform. Jia bo Zhu et al (1998) [3] proposed a time-delay neural network (TDNN) modeling method for predicting the treatment result. Suykens and Vandewalle (1999) presented LS-SVM method, in which the objective function includes an additional sum squared error term. LS-SVM is one of the methods by which the statistical learning theory can be introduced to practical application. It has its own advantages in solving the pattern recognition problem with small samples, nonlinearity, and higher dimension. And it can be easily introduced into learning problem such as function estimation[9].

Thus, in this paper, for SBR batch process, a MKPCA-LSSVM quality prediction method is proposed for dedicating to reveal the nonlinear relationship between process variables and COD of effluent qualities, and to build a quality prediction model. Firstly, three-way batch data of the SBR processes are unfolded batch-wisely, and KPCA is used to capture the nonlinear characteristics of batch-to-batch and variables, and can obtain irrelevant variables of un-fold data as inputs of model, COD measurements are taken as output of model. Then using LS-SVM to establish a correlated regression model between the featured principal components and COD variable. Finally, a nonlinear model is developed for COD prediction. Compared with the LS-SVM models, the result obtained by the proposed approach shows better estimation accuracy and is more extendable.

2 Process Description

The Sequencing Batch Reactor (SBR) is an activated sludge process in which one or more tanks are filled with wastewater and then operate in a batch mode. The influent and effluent from each tank are discontinuously allowing for the full treatment to take place in the same tank. If only one reaction tank is used an equalization tank is required in front of the reaction tank in order to store the wastewater during the period in which the reaction tank does not receive any inlet. After the treatment in the SBR system the water is discharged[1~2].

The SBR under study has a unique cyclic batch operation, usually with five well-defined phases:fill(anoxic and aerated), react, settle, decant and idle as shown figure 1.

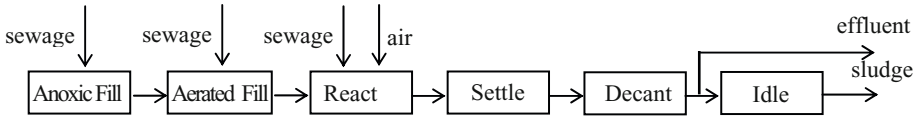


Fig. 1. Flow chart of Sequencing Batch Reactor Operation

In fill phase wastewater is admitted to the reaction tank. The phase can last until the tank is full or it can be time controlled. During the fill phase the reaction tank can allow for nitrifcation or denitrifcation. During react phase nitrifcation takes place.The phase is time controlled, however it can be omitted during high hydraulic loading. In Settle phase sedimentation of the sludge takes place. The phase is time controlled. If needed hauling of excess sludge can be initiated in the middle of the phase. In decant phase the treated water is decanted through a decanter. The phase is controlled by the capacity of the decanter. Hauling of excess sludge takes occurs throughout the entire phase. In idle phase there is no inlet and outlet and no aeration. The phase is time controlled, however, it can be omitted during high hydraulic loading. Hauling of excess sludge is performed throughout the entire phase.

The SBR goal is mainly nitrogen removal. Nitrogen removal has been in two steps: Nitrification : the ammonia is converted to nitrate by aerobic microorganisms and Denitrification : nitrate is converted to nitrogen gas under anoxic conditions by anoxic microorganisms.

3 MKPCA-LSSVM Modeling

3.1 Kernel Principal Component Analysis (KPCA)

Kernel principal components analysis (KPCA) is a nonlinear PCA method introduced by Sholkopf et al. [6],an it is a method of non-linear feature extraction. The non-linearity is introduced via a mapping of the data from the input space to a feature space. Linear PCA is then performed in the feature space, this can be expressed solely in terms of dot products in the feature space. Hence, the non-linear mapping need not be explicitly constructed, but can be specified by defining the form of the dot products in terms of a Mercer Kernel function. We concentrate on the case of a Gaussian kernel function. For the reasons of brevity, the detailed mathematical formulation of the KPCA can refer to documents [6].

3.2 LS-SVM (Least Square-Support Vector Machine)

LS-SVM follow the approach of a primal-dual optimization formulation, where this technique makes use of a so-called feature space where the inputs have been transformed by means of a (possibly infinite dimensional) nonlinear mapping. This is converted to the dual space by means of Mercer's theorem and the use of a positive definite kernel, without computing explicitly the mapping. It has its own advantages

in solving the pattern recognition problem with small samples, nonlinearity, and higher dimension. For the reasons of brevity, the detailed mathematical formulation of the KPCA can refer to documents [9].

3.3 MKPCA LS-SVM Modeling

The LS-SVM—likewise most ANNs—performs poorly if the network's input space contains redundant inputs, which unnecessarily increase the dimensionality of the input space and thus the size of the training set. So, KPCA is performed on the input variable data for overcoming this difficulty, thereby achieving the dimensionality reduction of the input space and extracting the nonlinear structure of input.

In this paper, a nonlinear batch irrelevant input variables of SBR is extracted on the basis of the MKPCA. Input data and output data are gathered from SBR Batch processes, input data forms a three-dimensional data matrix, $\mathbf{X}(I \times J \times K)$, where for batch process applications, I denotes cycle number, J denotes input variable number, and K denotes the number of samples within a cycle, output data forms a two-dimensional data matrix, $Y(I \times J_1)$, where for batch process applications, I denotes cycle number, J_1 denotes output variable number. MKPCA needs to unfold this matrix in order to obtain a two-way matrix, and then perform KPCA to extract the nonlinear structure of the unfolded matrix. That is the $\mathbf{X}(I \times J \times K)$ is unfolded, with each of the K time slabs concatenated to produce a two-way array, $\mathbf{X}_{new}(I \times JK)$. Multiway KPCA is equivalent to performing an KPCA on $\mathbf{X}_{new}(I \times JK)$ and un-fold data is proposed to extract the nonlinear local covariance information of process variable and get uncorrelated variables as input variables of model. LS-SVM model is established using uncorrelated input variables data and output variables data.

The data used in this research were collected from a pilot-scale SBR system. The operation cycles of the process are fixed. Each batch spend 8 hours of the time, it has 392 samples. Each cycle of the pilot plant SBR was based on alternating anoxic and aerobic reaction, where the filling only occurred during anoxic stages. The anoxic period was longer than aerobic period for increasing denitrification. Total filling volume was 200 liters, divided in six feeding parts during the cycle of 8 hours. The settling and draw spend of 1 hour and 0.46 hours respectively. Ten measurement variables can be measured online during the SBR run, including Influent Flow(Q), Turbidity of sewage, Suspended Substance of sewage(SS), Dissolved Oxygen(DO), Time of aerobic filter(T), Pondus Hydrogenii(PH),Oxidation Redution Potential (ORP), Mixed Liquor Suspended Solids(MLSS), temperature of aerobic filter(Kelvin K), SS of effluent, output quality variable is Chemical Oxygen Demand(COD).. Data for building the model is 60 batches, which is arranged in a three-way array $\underline{X}(60 \times 10 \times 392)$, Data for testing the model is 30 batches, which is arranged in a three-way array $\underline{X}(30 \times 10 \times 392)$.

4 Experimental Results

In this research, the Gaussian kernel is selected for MKPCA and for the mapping to a high-dimensional feature space since it is found to be appropriate to capture the

nonlinearity of the considered system by testing the prediction performance of a range of kernel functions.Twenty PCs were retained by the broken stick rule explaining 83.8% of the variation in the feature space. These PCs are input variables of LSSVM.

For illustration, the results of the LS-SVM and MKPCA-LSSVM output prediction for 30 batches are shown in Fig. 2. In Fig. 2, the solid line with circle symbols indicates the COD measurements, and the solid line with triangle symbols plots the corresponding COD prediction using LSSVM model and the solid line with square symbols plots the corresponding COD prediction using MKPCA LSSVM model.

It is clear that the predictions of two methods are much closer to the COD measurements. But the COD predicted by MKPCA-LSSVM model can be more exactly predicted. An analysis suggests that the method proposed can extract the nonlinear local covariance information of process variable and get uncorrelated variables as input variables of model.

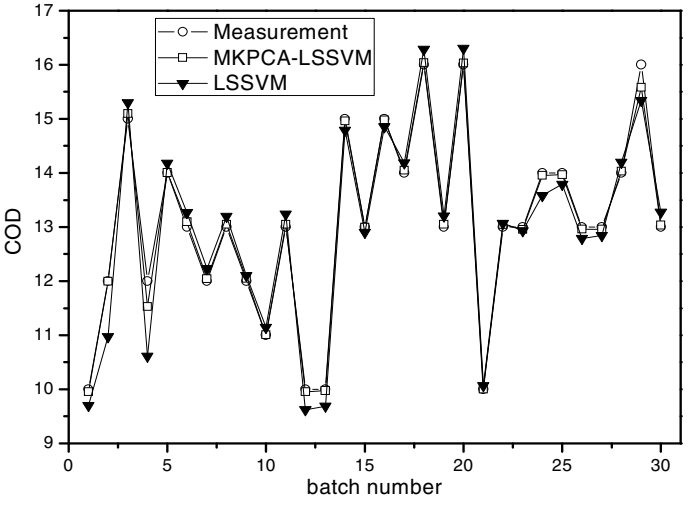


Fig. 2. Comparison of LSSVM and MKPCA LSSVM based prediction of COD using data

5 Conclusion

In this paper, a hybrid strategy integrating MKPCA and LSSVM has been presented for COD prediction of SBR batch processes. The proposed MKPCA-LSSVM strategy uses MKPCA for reducing the dimensionality of the process input space and can effectively be capture the nonlinear relations among the batch operations of the SBR and the first few principal component scores that explain a large amount of variance in the input data are used to develop a LSSVM model correlating inputs and outputs. Principal advantages of LSSVM-based models are (i) ability to approximate nonlinear input–output relationships efficiently; (ii) a model can be constructed exclusively from the historic process input–output data. The results obtained demonstrate that the proposed methodology is an attractive formalism for COD prediction of SBR

batch processes and the predicted precision of the proposed model is superior to the LSSVM model. The results have demonstrated the effectiveness of the proposed method. The COD prediction of sewage disposing effluent quality can be helpful to optimal control of the wastewater treatment process, and it has a certain practical worthiness.

Acknowledgments. This work was supported by the cooperation project between governments of China and Bulgaria under Grant 12-11.

References

1. Rubio, M., Colomer, J., Ruiz, M., Colprim, J., Melndez, J.: Qualitative trends for situation assessment in SBR wastewater treatment process, Technical report, Workshop Besai 2004, Valencia, Spain (August 2004)
2. Lee, D.S., Vanrolleghem, P.A.: Monitoring of a Sequencing Batch Reactor Using Adaptive Multiblock Principal Component Analysis,Biotechnol. Bioeng. 82, 489–497 (2003)
3. Jiabao Z., Zurcher, J., Rao, M., Meng, M.Q.-H.: An Online Wastewater Quality Predication System Based on a Time-delay Neural Network. J. Engineering Applications of Artificial Intelligence 11,747–758 (1998)
4. Nomikos, P., Macgregor, J.F.: Multiway Partial Least Squares in Monitoring Batch Processes. J.Chemometrics Intell. Lab. Syst. 30, 108–197 (1995)
5. Lee, J.-M., Yoo, C.K., Choi, S.W., Vanrolleghem, P.A., Lee, I.-B.: Nonlinear Process Monitoring Using Kernel Principal Component Analysis. Chem. Eng. Sci. 59, 223–234 (2004a)
6. Sholkopf, B., Smola, A., Müller, K.-R.: Nonlinear Component Analysis as a Kernel Eigenvalue Problem. Neural Comput.10, 1299–1399 (1998)
7. Lee, J.-M., Yoo, C.K., Lee, I.-B.: Fault Detection of Batch Processes Using Multiway Kernel Principal Component Analysis. Comput. Chem. Eng. 28, 1837–1847 (2004b)
8. Qin, S.J., McAvoy, T.J.: Nonlinear PLS Modeling using Neural Networks. J. Comput. Chem. Eng. 16, 379–391 (1992)
9. Suykens, J.A.K., Vandewalle, J.: Least Squares Support Vector Machine classifiers. J. Neural Processing Letters 9, 293–300 (1999)
10. Smola, A.J.: Regression Estimation with Support Vector Learning Machines. Master's Thesis, Technische Universität München (1996)

A Fixed-Point EM Algorithm for Straight Line Detection

Chonglun Fang and Jinwen Ma *

Department of Information Science, School of Mathematical Sciences and LMAM, Peking University, Beijing, 100871, China

Abstract. Straight line detection is a basic technique in image processing and pattern recognition. It has been investigated from different aspects, but is still very challenging in practical applications. In this paper, based on the finite mixture model and under the EM framework, we maximize the Q-function by differentiation and construct a fixed-point EM algorithm for straight line detection. It is demonstrated by the experiments that this proposed algorithm can effectively detect the straight lines from a digital image or dataset.

Keywords: Straight line detection, Expectation Maximization (EM), Fixed-Point iteration.

1 Introduction

Straight line detection is a basic technique in image processing and pattern recognition. In fact, it is a process of locating the straight lines from a digital image or 2-dimensional dataset and there are a variety of learning algorithms for straight line detection. In the literature, the Hough transform (HT) [1] and its extensions [2] are important tools for straight line detection. Generally, the Hough transform suffers from heavy computational cost. In order to alleviate this weakness, Random Hough Transform (RHT) [3] and the constrained Hough Transform [4] were further established. From the other aspects, there have also established many learning algorithms for straight line or curve detection (e.g., [5]-[8]).

The local principal component analysis (PCA) algorithm [9]-[12], as an extension of PCA [13], is often used for straight line detection. It implements the least mean square error reconstruction (LMSER) principle [14] and detects the straight lines via minimizing the following cost function [11]:

$$E = \sum_{k=1}^{K} E_k = \sum_{k=1}^{K} \sum_{x_t \in \mathcal{L}_k} d^2(x_t, \mathcal{L}_k) \tag{1}$$

where x_t is the t-th sample point belonging to the line $\mathcal{L}_k$ and $d(x_t, \mathcal{L}_k)$ denotes the Euclidean distance from the data point x_t to the line $\mathcal{L}_k$. Actually, the line $\mathcal{L}_k$ can be considered as a special subset of points. On the other hand, the line $\mathcal{L}_k$

* Corresponding author, jwma@math.pku.edu.cn

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 136–143, 2011.

can be also regarded as a cluster. Actually, E reaches its minimum when the k-th straight line is the first principal component vector of the cluster $\mathcal{L}_k$. Therefore, the solution of $\mathcal{L}_k$ is just the same as that of the local principal component analysis method.

Each x_t is assigned to a straight line by the classification membership function given by

$$I(x_t, k) = \begin{cases} 1 & \text{if } k = \arg\min d(x_t, \mathcal{L}_j),\ j = 1, \cdots, K \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

In fact, x_t belongs to $\mathcal{L}_k$ if and only if $I(x_t, k) = 1$. It means that the distance from x_t to $\mathcal{L}_k$ is the smallest one. Thus, we can update $\mathcal{L}_k$ by the following rule: $\mathcal{L}_k$ is the first principal component of the subset $\mathcal{L}_k$. Recently, the RPCL algorithm [15] was combined with the local PCA algorithm for straight line detection [6].

In this paper, we utilize the finite mixture model for straight line detection. It is not so easy to solve the maximum solution of the log-likelihood function directly. So, we resort to the Expectation Maximization (EM) [16] and analyze the Q-function. By the differentiation of the Q-function, we construct a fixed-point EM algorithm for straight line detection. It is demonstrated by the experiments that the proposed fixed-point EM algorithm can detect the straight lines from a datatet effectively.

The rest of the paper is organized as follows. We begin to introduce the finite mixture model for straight line detection in Section 2. We then derive and present our fixed-point EM algorithm in Section 3. The experimental results are further demonstrated in Section 4. Finally, a brief conclusion is made in Section 5.

2 The Finite Mixture Model for Straight Line Detection

Here, it is assumed that the number of straight lines in our learning model is equal to the number of actual straight lines in the image or dataset. For simplicity, we only focus on the 2-dimensional situation, but the derivation and analysis can be easily generalized to the situations with higher dimensions. Let the dataset be denoted by $\mathcal{S} = \{x_t\}_{t=1}^N$ and the point $x_t = (x_{t1}, x_{t2})^T$. Then, a point $x = (x_1, x_2)^T$ on a straight line $\mathcal{L}_k$ satisfies:

$$\frac{x_1 - m_{k1}}{\ell_{k1}} = \frac{x_2 - m_{k2}}{\ell_{k2}}, \tag{3}$$

where $m = (m_{k1}, m_{k2})^T$ is a specific point on the line $\mathcal{L}_k$, and

$$\ell_{k1}{}^2 + \ell_{k2}{}^2 = 1. \tag{4}$$

Thus, the distance from the sample point x_t to the line $\mathcal{L}_k$ can be computed by

$$\begin{aligned} d^2(x_t, \mathcal{L}_k) &= d^2(x_t, \ell_k, m_k) && (5) \\ &= |x_t - m_k|^2 - |(x_t - m_k, \ell_k)|^2 && (6) \\ &= (x_{t1} - m_{k1})^2 + (x_{t2} - m_{k2})^2 - [(x_{t1} - m_{k1})\ell_{k1} + (x_{t2} - m_{k2})\ell_{k2}]^2 \end{aligned}$$

where $(x_t - m_k, \ell_k)$ denotes the inner product of $x_t - m_k$ and ℓ_k.

In this situation, we can establish the following finite mixture model:

$$q(x|\Theta_K) = \sum_{j=1}^{K} \pi_j q(x|\ell_j, m_j, \sigma_j), \tag{7}$$

where

$$q(x|\ell_j, m_j, \sigma_j) = \frac{1}{\sqrt{2\pi}\sigma_j} \exp\{-\frac{d^2(x_t, \ell_j, m_j)}{2\sigma_j^2}\}, \tag{8}$$

$$\|\ell_j\|^2 = {\ell_{j1}}^2 + {\ell_{j2}}^2 = 1, \quad j = 1, \ldots, K, \tag{9}$$

$$\sum_{j=1}^{K} \pi_j = 1. \tag{10}$$

In this mixture model, π_i represents the mixing proportion. $d^2(x_t, \ell_k, m_k)$ denotes the distance between the sample point x_t and the line with parameters ℓ_k and m_k. σ_i can be considered as the noise level of the dataset. That is, when σ_i is large, the noise level is high.

As it is a finite mixture model, we are difficult to find the maximum of its log-likelihood function directly. So, we resort to the EM algorithm [16]. Under the EM framework, we introduce a missing variable j and construct the Q-function:

$$Q(\Theta_K^h, \Theta_K^{h+1}) = \frac{1}{N} \sum_{t=1}^{N} \sum_{j=1}^{K} p(j|x_t, \Theta_K^h) \ln q(x_t|j, \Theta_K^{h+1}) \tag{11}$$

$$= \frac{1}{N} \sum_{t=1}^{N} \sum_{j=1}^{K} \frac{\pi_j^h q(x_t|\ell_j^h, m_j^h, \sigma_j^h)}{\sum_{i=1}^{K} \pi_i^h q(x_t|\ell_j^h, m_j^h, \sigma_j^h)} \ln[\pi_j^{h+1} q(x_t|\ell_j^{h+1}, m_j^{h+1}, \sigma_j^{h+1})]$$

$$= \frac{1}{N} \sum_{t=1}^{N} \sum_{j=1}^{K} p_j(t) \ln[\pi_j^{h+1} q(x_t|\ell_j^{h+1}, m_j^{h+1}, \sigma_j^{h+1})] \tag{12}$$

where $p_j(t) = \frac{\pi_j^h q(x_t|\ell_j^h, m_j^h, \sigma_j^h)}{\sum_{i=1}^{K} \pi_i^h q(x_t|\ell_j^h, m_j^h, \sigma_j^h)}$. For simplicity, the Q-function is denoted by

$$Q = \frac{1}{N} \sum_{t=1}^{N} \sum_{j=1}^{K} p_j(t) \ln[\pi_j q(x_t|\ell_j, m_j, \sigma_j)]. \tag{13}$$

3 Proposed Fixed-Point EM Algorithm

In the EM algorithm, it is key to solve the maximum of the Q-function. We now analyze the Q-function and try to establish a fixed-point learning algorithm to solve the maximum of Q-function.

Since $\sum_{j=1}^{K} \pi_j = 1$ and ${\ell_{j1}}^2 + {\ell_{j2}}^2 = 1$ for any j, we introduce the Lagrange multiplier $\beta, \lambda_j (j = 1, \ldots, K)$ and the Lagrange function

$$L(\Theta_K, \beta, \lambda_1, \ldots, \lambda_K) = Q + \beta(1 - \sum_{j=1}^{K} \pi_j) + \sum_{j=1}^{K} \lambda_j (1 - \ell_{j1}^2 - \ell_{j2}^2). \tag{14}$$

By differentiation, we have the following derivatives:

$$\frac{\partial L}{\partial \pi_j} = \frac{1}{N}\sum_{j=1}^{K}\frac{1}{\pi_j}p_j(t) - \beta, \tag{15}$$

$$\frac{\partial L}{\partial \beta} = 1 - \sum_{j=1}^{K}\pi_j, \tag{16}$$

$$\frac{\partial L}{\partial \lambda_j} = 1 - l_{j1}^2 - l_{j2}^2, \tag{17}$$

$$\frac{\partial L}{\partial \ell_{j1}} = \frac{1}{N}\sum_{j=1}^{K} p_j(t)\frac{1}{\sigma_j^2}\{-(x_{t1}-m_{j1})[(x_{t1}-m_{j1})\ell_{j1} + (x_{t2}-m_{j2})\ell_{j2}]\}, \tag{18}$$

$$\frac{\partial L}{\partial m_{j1}} = \frac{1}{N}\sum_{j=1}^{K} p_j(t)\frac{1}{\sigma_j^2}\{-(x_{t1}-m_{j1}) + \ell_{j1}(x_t - m_j, \ell_j)\}, \tag{19}$$

$$\frac{\partial L}{\partial \sigma_j} = \frac{1}{N}\sum_{j=1}^{K} p_j(t)\{\frac{1}{\sigma_j} + \frac{1}{\sigma_j^3}d^2(x_t, \ell_j, m_j)\}. \tag{20}$$

By letting these derivatives given by Eqs. (15)-(20) be 0, we have

$$\beta = \frac{1}{N}\sum_{j=1}^{K}\sum_{t=1}^{N} p_j(t), \tag{21}$$

and further obtain the following fixed-point learning algorithm:

$$m_j^{h+1} = \frac{\sum_t p_j(t)x_t}{\sum_t p_j(t)}, \tag{22}$$

$$\pi_j^{h+1} = \frac{1}{N}\sum_t p_j(t), \tag{23}$$

$$(\sigma_j^{h+1})^2 = \frac{\sum_t p_j(t)d^2(x_t, \ell_k, m_k)}{\sum_t p_j(t)}, \tag{24}$$

and ℓ_j is the eigenvector of $\Sigma_j = \sum_{j=1}^{N} p_j(t)(x_t - m_j)(x_t - m_j)^T$ corresponding to the largest eigenvalue.

Based on the above fixed-point learning algorithm, we can establish the fixed-point EM algorithm which consists of the following three steps:

(i) Initialization of the parameters.
(ii) Update m_j, π_j, σ_j^2 by Eqs. (22)-(24). Update ℓ_j by the eigenvector of $\Sigma_j = \sum_{j=1}^{N} p_j(t)(x_t - m_j)(x_t - m_j)^T$ corresponding to the largest eigenvalue.
(iii) Repeat Step (ii) until the values of parameters are unchanged.

4 Experiments Results

In this section, several simulation experiments are carried out to demonstrate the performance of the fixed-point EM algorithm for straight line detection. We consider the binary images or datasets of four straight lines with three kinds of noises. The true parameters of the finite mixture models for the three datasets are listed in Table 1. Obviously, the noise levels in $\mathcal{S}_2$ and $\mathcal{S}_3$ are much higher than that of $\mathcal{S}_1$.

In our experiments, we set the number of straight lines to be the true number of straight lines, i.e., $K = 4$. We implement the fixed-point EM algorithm on each dataset, with the parameters being initialized by the random Hough transform [3]. The algorithm stops if $|Q(\Theta_K^{new}) - Q(\Theta_K^{old})| < 10^{-6}$. The results of the straight line detection as well as the obtained Q-function are shown in Fig. 1-3, respectively. The learned parameters of the finite mixture model on each experiment are listed in Table. 2.

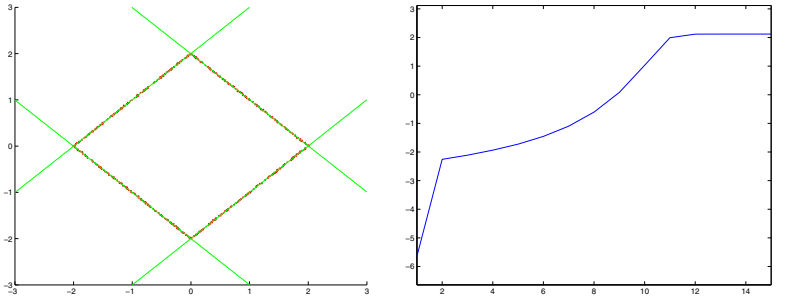
Table 1. The true parameters of the finite mixture models for the datasets $\mathcal{S}_1, \mathcal{S}_2$ and $\mathcal{S}_3$, respectively

Sample set	π_i	ℓ_i	m_i	σ_i
$\mathcal{S}_1$	0.25	(-0.7071,0.7071)	(1,1)	0.01
	0.25	(-0.7071,0.7071)	(-1,1)	0.01
	0.25	(-0.7071,0.7071)	(-1,-1)	0.01
	0.25	(0.7071,0.7071)	(1,-1)	0.01
$\mathcal{S}_2$	0.25	(-0.7071,0.7071)	(1,1)	0.2
	0.25	(-0.7071,0.7071)	(-1,1)	0.2
	0.25	(-0.7071,0.7071)	(-1,-1	0.2
	0.25	(0.7071,0.7071)	(1,-1)	0.2
$\mathcal{S}_3$	0.25	(-0.7071,0.7071	(1,1)	0.3
	0.25	(-0.7071,0.7071	(-1,1)	0.3
	0.25	(-0.7071,0.7071	(-1,-1	0.3
	0.25	(0.7071,0.7071)	(1,-1)	0.3

It can be observed from the figures in Fig. 1 that the Q-function increases during the iterations and finally reaches its maximum. Meanwhile, the straight lines are accurately located in each case. We can also observe that the Q-function increases sharply at the beginning of the iterations. The reason may be that the parameters are initialized by the random Hough transform, being only some rough estimates of the parameters. As the initialization of the parameters becomes better, the curve of the Q-function will be more smooth.

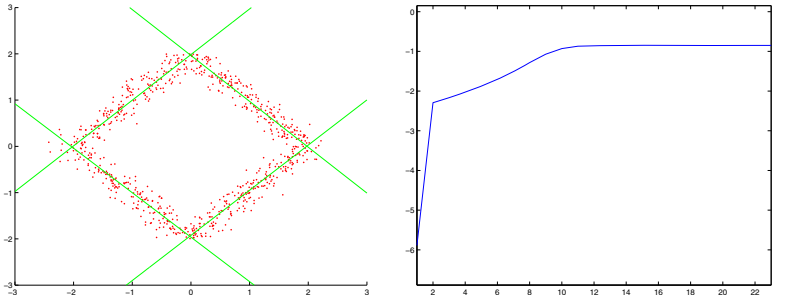
From these experimental results, we are sure that our proposed fixed-point EM algorithm can effectively detect the straight lines in all the three datasets with different noise levels. Moreover, it is shown in Fig. 3-(c) that the fixed-point EM algorithm also performs well on the strongly noisy situation.

It is pity that the number of straight lines in the learning mixture model should be known in advance. But this information may be not available in



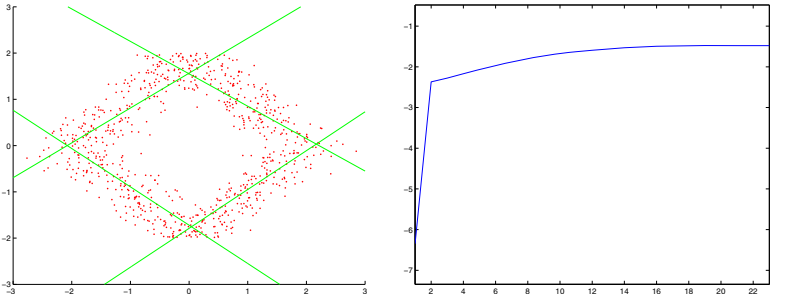
(a) (b)

Fig. 1. (a). The experimental result of straight line detection on the dataset $\mathcal{S}_1$, (b). The sketch of the Q-function on the iterations



(a) (b)

Fig. 2. (a). The experimental result of straight line detection on the dataset $\mathcal{S}_2$, (b). The sketch of the Q-function on the iterations



(a) (b)

Fig. 3. (a). The experimental result of straight line detection on the dataset $\mathcal{S}_3$, (b). The sketch of the Q-function on the iterations

Table 2. The learned parameters of the finite mixture models on the three datasets $\mathcal{S}_1, \mathcal{S}_2$ and $\mathcal{S}_3$, respectively

Sample set	π_i	ℓ_i	m_i	σ_i
$\mathcal{S}_1$	0.2498	(-0.7084,0.7058)	(0.9715,1.0290)	0.0103
	0.2489	(-0.7084,0.7058)	(-0.9695,1.0302)	0.0083
	0.2502	(-0.7077,0.7065)	(-0.9516,-1.0491)	0.0089
	0.2511	(0.7067,0.7075)	(1.0120,-0.9871)	0.0097
$\mathcal{S}_2$	0.2617	(0.7101,-0.7041)	(1.0192,0.9542)	0.1726
	0.2421	(0.7119, 0.7023)	(-1.0138,0.9835)	0.1812
	0.2380	(0.7210,-0.6930)	(-0.9720,-1.0285)	0.1760
	0.2582	(-0.7157,-0.6984)	(0.9778,-0.9707)	0.2138
$\mathcal{S}_3$	0.2386	(-0.7908,0.6120)	(0.8080,1.0306)	0.2695
	0.2557	(-0.7459,-0.6661)	(-0.8380,1.0016)	0.2826
	0.2335	(0.7296,-0.6839)	(-0.8854,-0.9882)	0.2867
	0.2722	(-0.7323,-0.6810)	(0.8837,-0.9461)	0.3386

practical applications. In order to overcome this weakness, we can introduce the Bayesian Ying-Yang (BYY) harmony learning system [17]-[18] and the entropy penalized automated model selection mechanism [19] into the fixed-point learning algorithm.

5 Conclusions

We have investigated the straight line detection problem from the finite mixture modeling. Since it is difficulty to solve the maximum of the log-likelihood function, we resort to the EM algorithm and analyze the Q-function. By differentiation, we derive a fixed-point learning procedure for maximizing the Q-function and thus construct a fixed-point EM algorithm for straight line detection. It is demonstrated by the experiments that the proposed fixed-point EM algorithm can effectively locate the straight lines in a dataset.

Acknowledgements

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

References

1. Ballard, D.: Generalizing the Hough Transform to Detect Arbitrary Shapes. Pattern Recognition 13(2), 111–122 (1981)
2. Wolfson, H.: Generalizing the Generalized Hough Transform. Pattern Recognition Letters 12(9), 565–573 (1991)
3. Xu, L., Oja, E., Kultanen, P.: A New Curve Detection Method - Randomized Hough Transform (RHT). Pattern Recognition Letters 11(5), 331–338 (1990)

4. Olson, C.F.: Constrained Hough Transforms for Curve Detection. Computer Vision and Image Understanding 73, 329–345 (1998)
5. Liu, Z., Qiao, H., Xu, L.: Multisets Mixture Learning-Based Ellipse Detection. Pattern Recognition 39(4), 731–735 (2006)
6. Liu, Z., Chiu, K., Xu, L.: Strip Line Detection and Thinning by RPCL-Based Local PCA. Pattern Recognition Letters 24(14), 2335–2344 (2003)
7. Lu, Z.W., Cheng, Q.S., Ma, J.W.: A Gradient BYY Harmony Learning Algorithm on Mixture of Experts for Curve Detection. In: Gallagher, M., Hogan, J.P., Maire, F. (eds.) IDEAL 2005. LNCS, vol. 3578, pp. 250–257. Springer, Heidelberg (2005)
8. Chen, G., Li, L., Ma, J.: A Gradient BYY Harmony Learning Algorithm for Straight Line Detection. In: Sun, F., Zhang, J., Tan, Y., Cao, J., Yu, W. (eds.) ISNN 2008, Part I. LNCS, vol. 5263, pp. 618–626. Springer, Heidelberg (2008)
9. Xu, L.: Multisets Modeling Learning: an Unified Theory for Supervised and Unsupervised Learning. In: IEEE International Conference on Neural Networks, 1994. IEEE World Congress on Computational Intelligence, vol. 1, pp. 315–320 (1994)
10. Kambhatla, N., Leen, T.: Dimension Reduction by Local Principal Component Analysis. Neural Computation 9(7), 1493–1516 (1997)
11. Xu, L.: Vector Quantization by Local and Hierarchical LMSER. In: Proc. of 1995 Intl. Conf. on Artificial Neural Networks, vol. 2, pp. 575–579 (1995)
12. Xu, L.: An Overview on Unsupervised Learning from Data Mining Perspective. In: Allinson, N., Yin, H., Allinson, L., Slack, J. (eds.) Advances in Self-Organising Maps, pp. 181–209 (2001)
13. Jolliffe, I.T.: Principal Component Analysis, 2nd edn. Springer, Heidelberg (2002)
14. Xu, L.: Least Mean-Square Error Reconstruction Principle for Self-Organizing Neural-Nets. Neural Networks 6(5), 627–648 (1993)
15. Xu, L., Krzyzak, A., Oja, E.: Rival Penalized Competitive Learning for Clustering Analysis, RBF Net, and Curve Detection. IEEE Transactions on Neural Networks 4(4), 636–649 (1993)
16. Dempster, A., Laird, N., Rubin, D.: Maximum Likelihood from Incomplete Data via EM Algorithm. Journal of the Royal Statistical Society Series B-Methodological 39(1), 1–38 (1977)
17. Ma, J.W., He, X.F.: A Fast Fixed-Point BYY Harmony Learning Algorithm on Gaussian Mixture with Automated Model Selection. Pattern Recognition Letters 29(6), 701–711 (2008)
18. Ma, J.W., Liu, J.F.: The BYY Annealing Learning Algorithm for Gaussian Mixture with Automated Model Selection. Pattern Recognition 40(7), 2029–2037 (2007)
19. Ma, J.W., Wang, T.J.: Entropy Penalized Automated Model Selection on Gaussian Mixture. International Journal of Pattern Recognition and Artificial Intelligence 18(8), 1501–1512 (2004)

A Novel Classifier Ensemble Method Based on Class Weightening in Huge Dataset

Hamid Parvin, Behrouz Minaei, Hosein Alizadeh, and Akram Beigi

School of Computer Engineering,
Iran University of Science and Technology (IUST), Tehran, Iran
{parvin,b_minaei,halizadeh,beigi}@iust.ac.ir

Abstract. While there are many methods in classifier ensemble, there is not any method which uses weighting in class level. Random Forest which uses decision trees for problem solving is the base of our proposed ensemble. In this work, we propose a weightening based classifier ensemble method in class level. The proposed method is like Random Forest method in employing decision tree and neural networks as classifiers, and differs from Random Forest in employing a weight vector per classifier. For evaluating the proposed weighting method, both ensemble of decision tree and neural networks classifiers are applied in experimental results. Main presumption of this method is that the reliability of the predictions of each classifier differs among classes. The proposed ensemble methods were tested on a huge Persian data set of handwritten digits and have improvements in comparison with competitors.

Keywords: Classifier Ensembles, Random Forest, Bagging, Class Weightening.

1 Introduction

Ensemble algorithms train multiple base classifiers and then combine their predictions. Generalization ability of an ensemble could be significantly better than a single classifier for difficult problems [4].

In [11] and [12], the relationship between the ensemble and its component artificial neural networks (ANN) has been analyzed from the context of both regression and classification, which has revealed that it may be better to ensemble many instead of all of the ANNs at hand. They trained a number of ANNs at first. Then random weights were assigned to those networks and genetic algorithm (GA) was employed to evolve the weights so that they can characterize to some extent the fitness of the ANNs in constituting an ensemble. Finally some ANNs were selected based on the evolved weights to make up the ensemble.

In contrary, assuming that the reliability of the classifiers differs among classes, an approach based on dynamic selection of the classifiers by taking into account their individual votes, was proposed in [5]. In particular, a subset of the predictions of each classifier was taken into account during weighted majority voting. Others were considered as unreliable and were not used during combination.

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 144–150, 2011.

In general, an ensemble is built in two steps: (a) generating multiple base classifiers and then (b) combining their predictions. AdaBoost [8] and Bagging [1] are two famous methods in this field.

AdaBoost sequentially generates a series of base classifiers where the training instances wrongly predicted by a base classifier will play more important role in the training of its subsequent classifier. Bagging generates many samples from the original training set via bootstrap sampling [7] and then trains a base classifier from each of these samples, whose predictions are combined via majority voting. A kind of bagging method is Random Forest, where many decision trees (DT) are trained over distinguished perspectives of training dataset [2].

An ANN has to be configured to be able to produce the desired set of outputs, given an arbitrary set of inputs. Various methods of setting the strength of connections exist. One way is to set the weights explicitly, using a prior knowledge. Another way is to 'train' the ANN, feeding it by teaching patterns and then letting it change its weights according to some learning rule [10]. In this paper an MLP neural network is used as classifier.

GA which is one of the optimization paradigms, bases on natural process [3]. A GA can be considered as a composition of three essential elements: first, a set of potential solutions called individuals or chromosomes that will evolve during a number of iterations (generations). This set of solutions is also called population. Second, an evaluation mechanism (fitness function) that allows assessing the quality or fitness of each individual of the population. And third, an evolution procedure that is based on some "genetic" operators such as selection, crossover and mutation. The crossover takes two individuals to produce two new individuals.

The quality of the individuals is assessed with a fitness function. The result is a real value for each individual. The best individuals will survive and are allowed to produce new individuals.

A common and obvious way for classifying an instance is from a sequence of questions, so that next question is asked with regard to this current question. Using trees are the most common representation way for theses question-answers. DT is used to create a classifier ensemble, expansively. Also, they are used for the application of data mining and clustering. Their functionality is understandable for human. Besides, unlike other methods such as ANN, they are very quick. It means their learning phase is quicker than other methods [6].

In the next section, we explain the proposed ensemble method in more details.

2 Proposed Method

Let us to assume that total number of obtained classifiers is denoted by M. Let the total number of labels (classes) be denoted by N. The solution of the selection problem encoded in the form of a chromosome which has $N \times M$ genes. First N genes belong to the first classifier. Second subsequent N genes belong to the second classifier. i-th N genes belong to the i-th classifier. The encoding of a chromosome is illustrated in Figure 1, for $N=10$. The genes of each chromosome have real data type. In the exemplary chromosome depicted in Figure 1, the first classifier is not allowed to vote for only fourth, fifth, eighth and tenth classes, And it is allowed to vote for first, second, third, sixth, seventh and ninth classes with coefficients 0.5, 0.6, 0.6, 1, 1 and 1 respectively.

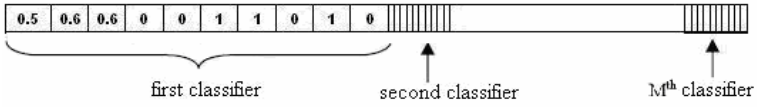


Fig. 1. Encoding of a chromosome of used GA, provided the number of classes is *N=10* in the problem

Let us denote a chromosome by *b*, an array of *N*×*M* numbers belonging to closed interval [0,1]. In the Figure 1, *b(i)* is the effect weight of *k*-th classifier to vote for selecting *j*-th class, where *k* and *j* is calculated according to the equation 1 and 2 respectively.

$$k = \lceil i / N \rceil \tag{1}$$

$$j = i \mod N \tag{2}$$

Because of non-normalization of the raw *b* chromosome, we first convert it to a normalized version according to the following equation. We denote this normalized version of chromosome *b* by *nb*.

$$nb(b) == \{b \to nb \mid b \in [0,1]^{N\times M}, nb \in [0,1]^{N\times M}, nb(b)_i = \frac{b_i}{\sum_{q=1}^{M} b_{(q-1)*k+j}}\} \tag{3}$$

where *k* and *j* are the same in the equation 1 and 2. The *nb* is employed in calculating confidences of classifier ensemble per classes for a data item *x*. These confidences are obtained according to the following equation.

$$conf(b,x) == (conf(b,x)_1, conf(b,x)_2, ..., conf(b,x)_N) \mid b \in [0,1]^c, \tag{4}$$
$$conf(b,x)_j = \sum_{i=1}^{M} C_{i,j}(x) * nb(b)_{(i-1)*k+j}\}$$

where *k* and *j* are the same in the equation 1 and 2, *c* is length of chromosome, i.e. *N*×*M*, and $C_{i,j}(x)$ is considered as output of *i*-th classifier for *j*-th class for data item *x*.

Now we define the following terms for the following usage. Normalization of an array of a number is defined as following equation.

$$normalize(a) = \{(normalize(a)_1, normalize(a)_2, ..., normalize(a)_c) \mid a \in [0,1]^c, \tag{5}$$
$$normalize(a)_i = \frac{a_i}{\sum_{j=1}^{c} a_j}\}$$

Label is a pre-assigned category of data item $\boldsymbol{x}$. It is denoted by *l*. $l_i(\boldsymbol{x})$ is a number which is considered as membership of $\boldsymbol{x}$ to the class *i*. If $\boldsymbol{x}$ belongs to *i*-th class, $l_i(\boldsymbol{x})$ is 1 and $l_j(\boldsymbol{x})$ is 0 for all $j \neq i$. The fitness of each chromosome (classifier ensembles) is defined as the amount of its accuracy on the evaluation set. The fitness function of a chromosome is computed as equation 6.

$$fitness(b, DV) = \sum_{x \in DV} \left\| normalize(conf(b, x)) - l(x) \right\| \quad (6)$$

where DV is validation dataset, l is label function. ||.|| is considered as one of norm function like Euclidean distance.

In all experiment, genetic parameters are fixed. Tournament selection is used for the reproduction phase. In this study, the crossover operator that has an important role in evolutionary computing, allowing them to explore the problem space by sharing different chromosomes information is two-point crossover. The mutation operator, allowing evolutionary computing algorithm to exploiting the problem space, is applied to each entry of the offspring chromosomes with a probability $p_{mut} = 0.01$. Probability of selection operator is $p_{cross} = 0.8$. The tournament size is fixed to 5. In the simulation experiments, the population size is selected as 200. It means that 200 different ensemble candidates evolved simultaneously. Pseudo-code of the GA used in the proposed method for evolving the classifier ensembles is shown in Figure 2.

```
Generate randomly an initial population of size POP_SIZE
For each chromosome in the population
        Compute fitness of the chromosome (as it will be    mentioned below)
For Iteration_Num = 1 .. GENERATION_NUM
        For Chromosome_Num = 1 .. POP_SIZE
                1-Select two parents from the old population
                2-Crossover the two parents to produce two offspring with probability
                P_cross
                3-Mutate each bit of each offspring with  probability P_mut
                4-Apply weighted majority to each of the offspring
                5-Compute fitness of each offspring (as it will be mentioned below)
        End for
        Replace the original population with the offsprings to form the new population
End for
Select the best chromosome as the resultant ensemble
```

Fig. 2. The GA used in the proposed method

We use two types of classifier in the ensembles: ANN, DT. The first classifier is ANN with N outputs which each of the outputs corresponds to a class.

The accuracy of each classifier ensembles is defined as the number of true estimation on the test data set.

In order to evaluate the proposed classifier selection approach, we compare it with weighted and unweighted static classifier selection for Hoda data set. Each chromosome is encoded as a string having M entries, one for each classifier with data types real and binary respectively in those two methods. In unweighted static classifier selection which has data type binary, if the value of a gene is 1; this means that the classifier is selected for being used in the corresponding ensemble. All the design parameters of the above-mentioned algorithm including population size, number of iterations, crossover and mutation rate etc. are kept the same. The Figure 3 illustrates the proposed method generally.

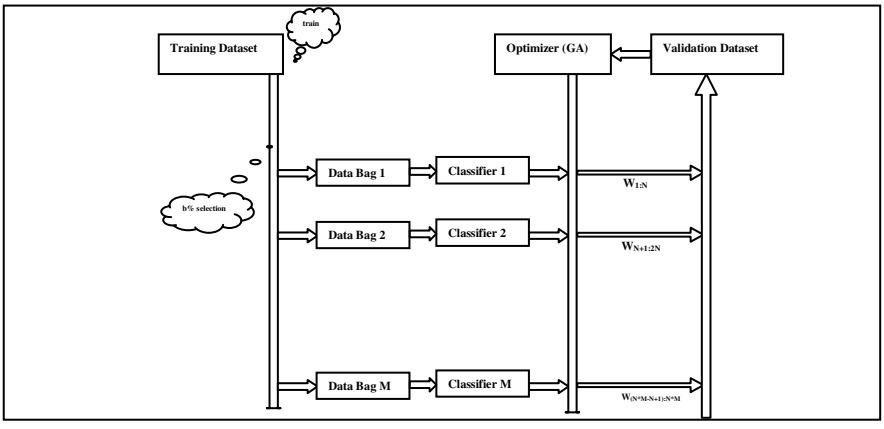


Fig. 3. Scheme of the weighted ensemble of classifiers

3 Experimental Results

Hoda data set [9] is a handwritten OCR data set. This data set contains 100000 data points. Some data instances are depicted in the Figure 4.

We have divided our data set into training, validation and test sets containing 60000, 20000 and 20000 data points, respectively. The validation data set acts as pseudo-testing for obtaining fitness of each chromosome as it was explained above. The ensemble is produced by bagging mechanism. Ensemble size is also set to 201. So, the training process is iterated 201 times for performing 201 different base classifiers, ANN and DT. Each classifier is trained over 10% of training dataset. As it is shown in Figure 5, the proposed method outperforms other methods and full ensemble. It shows that a full ensemble classification method can be optimized as well.

Fig. 4. Some instances of Farsi OCR data set, with different qualities

The method in column 2 in Figure 5, unweighted static classifier selection focuses only on selected classifiers which are allowed to vote. Because of unbalanced accuracy of classifiers in the ensemble, generally, the static classifier selection can give better results than the simple full ensemble. Usually the weighted approaches are doing better than unweighted ones as it is shown in the column 3 of the Figure 5. However, the results of weighted and unweighted approaches are close to each other, the weighted method slightly outperforms unweighted. It improves the result achieved by the full ensemble. Even though, a classifier is not able to achieve good accuracy in all classes; it may obtain a good accuracy on one special class. So, the proposed method has a good result. Figure 5 illustrates the same results in a diagram based representation.

Another aspect of the proposed approach is that its computational cost is very low. Although we can train just one MLP to reach to a good accuracy, it consumes many days for large data sets like Hoda. We need to train an MLP for some weeks to reach the accuracy approximately 98% on Hoda data set. The weak learners can converge to a good accuracy very soon, but the subsequent small improvements are very slow. In this approach, we have some weak base classifiers that are under-trained but ensemble of them is not under-trained. We provided 201 individual weak base MLPs or DTs as members in the ensemble. Also it is notable that both ensembles of MLPs and DTs are comparable, with fairly superior of DTs ensemble.

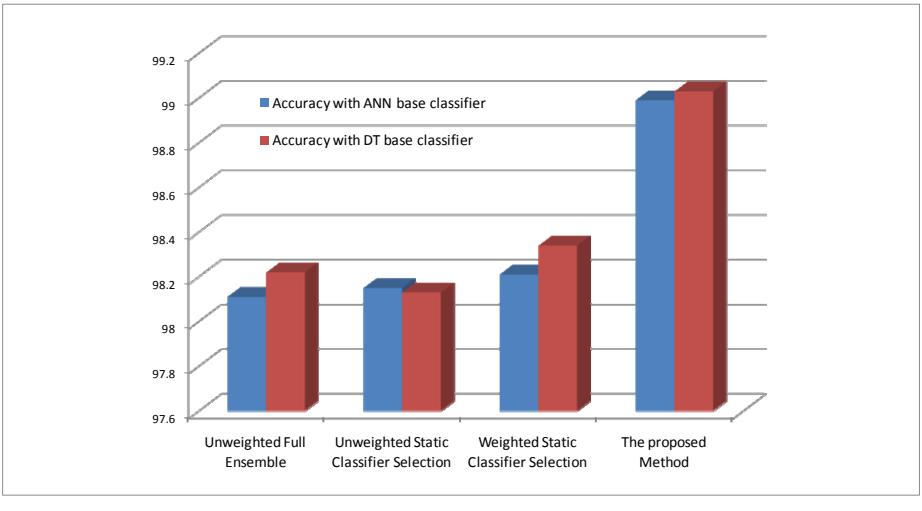


Fig. 5. Results of the proposed ensemble method

4 Conclusion

Because of their robustness and high performance, classifier ensemble methods are used for difficult problem solving. In this paper, a new ensemble algorithm is proposed, which is designed for building ensembles of bagging classifiers. The proposed

method is a weighted vote-based classifier ensemble like Random Forest method which employs DT and ANN as classifiers.

The empirical study on the very large dataset of Persian handwritten digits, Hoda shows that the proposed approach is superior to another combination of classifiers methods, as it is discussed. It effectively improves the accuracy of full ensemble of ANN or DT classifiers.

References

1. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
2. Breiman, L.: Random forests. MachineLearning 45, 5–32 (2001)
3. Davis, L.: Handbook of Genetic Algorithms. Van Nostrand Reinhold, New York (1991)
4. Dietterich, T.G.: Ensemble learning. In: Arbib, M.A. (ed.) The Handbook of Brain Theory and Neural Networks, 2nd edn. MIT Press, Cambridge (2002)
5. Dimililer, N., Varoğlu, E., Altınçay, H.: Vote-Based Classifier Selection for Biomedical NER Using Genetic Algorithms. In: Martí, J., Benedí, J.M., Mendonça, A.M., Serrat, J. (eds.) IbPRIA 2007. LNCS, vol. 4478, pp. 202–209. Springer, Heidelberg (2007)
6. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. John Wiley & Sons, NY (2001)
7. Efron, B., Tibshirani, R.: An Introduction to the Bootstrap. Chapman & Hall, New York (1993)
8. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of online learning and an application to boosting. In: Vitányi, P.M.B. (ed.) EuroCOLT 1995. LNCS, vol. 904, pp. 23–37. Springer, Heidelberg (1995)
9. Khosravi, H., Kabir, E.: Introducing a very large dataset of handwritten Farsi digits and a study on the variety of handwriting styles. Pattern Recognition Letters 28(10), 1133–1141 (2007)
10. Sanchez, A., Alvarez, R., Moctezuma, J.C., Sanchez, S.: Clustering and Artificial Neural Networks as a Tool to Generate Membership Functions. In: Proceedings of the 16th IEEE International Conference on Electronics, Communications and Computers (2006)
11. Zhou, Z.H., Wu, J.X., Jiang, Y., Chen, S.F.: Genetic Algorithm based Selective Neural Network Ensemble. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI 2001), Seattle, WA, vol. 2, pp. 797–802 (2001)
12. Zhou, Z.H., Wu, J.X., Tang, W.: Ensembling Neural Networks: Many Could Be Better Than All. Artificial Intelligence 137(1-2), 239–263 (2002)

Network-Scale Traffic Modeling and Forecasting with Graphical Lasso

Ya Gao, Shiliang Sun, and Dongyu Shi

Department of Computer Science and Technology
East China Normal University
500 Dongchuan Road, Shanghai 200241, P.R. China
ygao08@gmail.com, {slsun,dyshi}@cs.ecnu.edu.cn

Abstract. Traffic flow forecasting is an important application domain of machine learning. How to use the information provided by adjacent links more efficiently is a key to improving the performance of Intelligent Transportation Systems (ITS). In this paper, we build a sparse graphical model for multi-link traffic flow through the Graphical Lasso (GL) algorithm and then implement the forecasting with Neural Networks. Through a large number of experiments, we find that network-scale traffic forecasting with modeling by Graphical Lasso performs much better than previous research. Traditional approaches considered the information provided by adjacent links but did not extract the information. Thus, although they improved the performance to some extent, they did not make good use of the information. Furthermore, we summarize the theoretical analysis of Graphical Lasso algorithm. From theoretical and practical points of view, we fully verify the superiority of Graphical Lasso used in modeling for multi-link traffic flow forecasting.

Keywords: Traffic flow forecasting, Neural networks, Graphical Lasso.

1 Introduction

In recent years, research on statistics and computer science appears to intersect in the long-term goals. The most obvious area in this trend is that of graphical model. The graphical model is a family of probability distributions which defined according to a directed or undirected graph. Furthermore, the model provides a general methodology for approaching correlation problems. Often, the problems involve large-scale models with thousands or millions of random variables linked in complex ways [1]. Thus, graphical model makes a link between probability theory and graph theory [2]. It is used widely in many machine learning areas such as bioinformatics, information retrieval and image processing etc [1].

Graphical Lasso (GL) is an approach of estimating a sparse undirected graphical model through the use of L1 (lasso) regularization which has the basic model for continuous data obeying a multivariate Gaussian distribution with mean μ and covariance matrix Σ [3]. According to the components of inverse covariance matrix Σ^{-1}, we construct the graphical model. If the ij-th component of Σ^{-1} is zero,

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 151–158, 2011.

then there is no link between the two nodes corresponding to variables i and j in the graphical model. That is, variables i and j are conditionally independent given the other variables when the ij-th component of Σ^{-1} is zero. In next sections, we show the theoretical analysis of the procedure of Graphical Lasso algorithm.

Neural Networks (NNs) are well-known in machine learning for their good capability at modeling non-linear and uncertain problems [4]. They are widely used in traffic flow forecasting. From single-link single-task to multi-link multi-task traffic flow forecasting, or other various kinds of prediction algorithms, are all to improve the performance of forecasting. However, how to make good use of the information provided by adjacent links is the key. All the experiments performed in this paper are based on Neural Networks. More details about multi-link traffic flow forecasting can be seen in [5].

The remainder of this paper is organized as follows. In Section 2, we give the theoretical analysis of Graphical Lasso and its implementation in modeling sparse graphical model. In Section 3, combining with multi-link traffic flow forecasting, we show a large of experiments basing on Neural Networks with the model built by Graphical Lasso. Finally, conclusions and future works are given in Section 4.

2 Modeling Sparse Graphs

In modeling of a sparse graph, whether there is a link between two nodes is determined according to the corresponding component of the inverse covariance matrix. If the component is zero, there is no link between the two nodes. Otherwise, link the two nodes in the sparse graph. Thus, it makes sense to increase the sparsity of the inverse covariance matrix Σ^{-1}. In recent years, there are a number of researchers have taken an approximate or exact approach to this problem. For approximate approaches, sparse graphical model is estimated by fitting a lasso model to each variable, using the other variables as predictors [6]. The approaches adopt an AND rule to estimate whether a certain component of Σ^{-1} is zero or not. While the exact approaches are to maximize an L1-penalized log-likelihood problem [7-9]. Graphical Lasso is the exact approach of solving the problem to make the inverse covariance matrix Σ^{-1} more sparsely.

2.1 Problem Setup

Assume that there are N multivariate normal observations of dimension p, with mean μ and covariance Σ. Marking the empirical covariance matrix as S, the exact problem is to solve the L1-penalized log-likelihood problem

$$\Sigma^{-1} = \arg\max_{X \succ 0} \log\det X - trace(S \cdot X) - \rho \left\|X\right\|_1, \tag{1}$$

where $\left\|X\right\|_1$ is the L1 norm of matrix which is the sum of the absolute values of the elements of X, ρ is the penalty parameter which controls the extent of penalization [8].

Make a transformation of formula (1), we can write the problem as

$$\max_{X \succ 0} \min_{\|U\|_\infty \leqslant \rho} \log\det X + trace(X, S + U), \tag{2}$$

where $\|U\|_\infty$ denotes the maximum absolute value element of the symmetric matrix U. Exchange the max and the min, the dual problem of formula (2) is

$$\min_{\|U\|_\infty \leqslant \rho} -\log\det(S+U) - p, \tag{3}$$

where the relation between the primal and dual variables is: $X = (S+U)^{-1}$. To write neatly, let $M = S + U$ [8]. Then, the dual of our maximum likelihood problem is

$$\Sigma = \max\{\log\det M : \|M - S\|_\infty \leqslant \rho\}. \tag{4}$$

We can see that, through a series of transformations above, the inverse covariance matrix Σ^{-1} is estimated in the primal problem (1) while the covariance matrix Σ is estimated in the dual problem (4). Also, the diagonal element of Σ and S has the relation: $\Sigma_{ii} = S_{ii} + \rho$ for all i.

2.2 Block Coordinate Descent (BCD) Algorithm

According to formula (4), we begin to consider the estimation of Σ instead of Σ^{-1} as follows. Let W be the estimate of Σ. The algorithm is to optimize over each row and column of matrix W at a time, and repeats all columns until convergence. Details can be seen in [8]. Dividing W and S into blocks as

$$W = \begin{pmatrix} W_{11} & w_{12} \\ w_{12}^T & w_{22} \end{pmatrix}, S = \begin{pmatrix} S_{11} & s_{12} \\ s_{12}^T & s_{22} \end{pmatrix}.$$

The block coordinate descent algorithm solves the quadratic program

$$w_{12} = \arg\min_y \{y^T W_{11}^{-1} y : \|y - s_{12}\|_\infty \leqslant \rho\}, \tag{5}$$

for w_{12} at each iteration. Permuting the rows and columns to make the target column always be the last one, the BCD algorithm solves problem (5) for each column of W. Repeating until convergence, we will finally get a sparse matrix W which is also the covariance matrix Σ solving formula (4).

Furthermore, [8] gives the dual problem of problem (5) is

$$\min_\beta \{\frac{1}{2}\left\|W_{11}^{1/2}\beta - b\right\|^2 + \rho\|\beta\|_1\}, \tag{6}$$

where $b = W_{11}^{-1/2} s_{12}$. It is easy to find that formula (6) is similarly a lasso regression, and which is the launching point of our Graphical Lasso approach.

2.3 Graphical Lasso

In Graphical Lasso algorithm [3], let $W = S + \rho I$ firstly and the diagonal of W remains unchanged in the following. Then, for each row and column of W, solve the lasso problem (6) and obtain the solution β. Computing $w_{12} = W_{11}\beta$ and replacing the corresponding row and column with w_{12}. Like this, repeat until convergence.

After the whole procedure, we obtain the inverse matrix Σ^{-1}. In [3], the authors also mentioned a relatively cheap method for recovering the inverse covariance matrix from the obtained coefficient matrix in the Graphical Lasso algorithm.

According to the sparse matrix Σ^{-1}, we build the sparse undirected graphical model of the multi-variables included in the matrix. Data with the dimension of p has p nodes in the graphical model, and each dimension corresponds to a row and a column in the inverse covariance matrix. In the next section, combing with the practical application in traffic flow forecasting, we will get a more clear cognition of modeling with Graphical Lasso.

3 Experiments

3.1 Data Description

The datasets collected are the vehicle flow rates of discrete time series which were recorded every 15 min, gathering by the UTC/SCOOT system of the Traffic Management Bureau of Beijing along many road links. Vehicles per hour (vehs/h) is the unit of the data. In short-term traffic flow forecasting, we take 15 minutes as the prediction horizon and carry a one-step prediction. That is, we predict the traffic flow rates of the next 15 minutes interval using the historical data of a certain road link on the same time series [5].

Part of the real urban traffic map we selected is shown in Fig. 1. Each circle node in the figure denotes a road junction. The arrows show the directions of traffic flows from the upstream junctions to the corresponding downstream junctions. Paths without arrows represent no traffic flow records. Raw data is taken from March 1 to March 31, 2002, totaling 31 days [10]. Considering the malfunction of traffic flow detector, we wiped away the days with empty data. Finally, the remaining data we used are of 25 days and have totaling 2400 sample points. We divide the data into two parts, the first 2112 samples as training data and the rest as test data, in all the experiments we did.

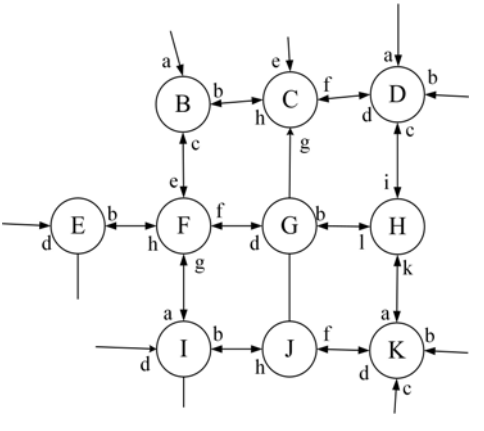


Fig. 1. The sketch map of road links used in the paper

3.2 Model Building

As can be seen from the data described above, the map has 31 links in all, in which every path with an arrow represents a link. Like what we did in previous experiments [5], we predict one interval traffic flow using traffic flows of the five continuous intervals before it on the same time series of the same link in single-link flow forecasting. While in multi-link flow forecasting, we use traffic flows of all the links with their each five intervals before the predicted interval.

In order to use the correlation between adjacent links more efficiently, we take traffic flows of 6 continuous intervals of each link to build the sparse graphical model through Graphical Lasso. There we will obtain the inverse variance matrix Σ^{-1} with dimension of 186 which is computed by 31 multiply 6, for there are 31 links and 6 selected traffic flows of each link. According to the actual meaning of traffic flow forecasting, we do not consider the correlation between the predicted traffic flow with the same interval traffic flows of all the other links. Thus, we ignore the components in the inverse covariance matrix corresponding to two same intervals traffic flows of each two links. Therefore, for one predicted traffic flow, there are at most 155 nodes linked in the graphical model.

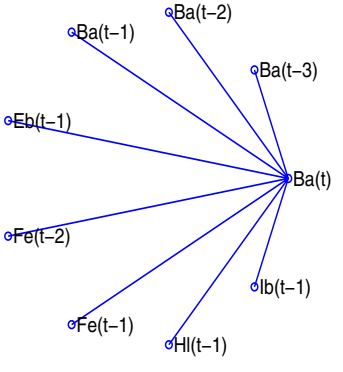


Fig. 2. The sparse graphical model identified for link Ba

Take link Ba as an example, in single-link traffic flow forecasting, we predict the traffic flow Ba(t) using the continuous traffic flows Ba(t-5), Ba(t-4), ..., Ba(t-1) of link Ba. However, in multi-link traffic flow, we consider the 5 intervals t-5, t-4, ..., t-1 of all the 31 links. That is, we predict the traffic flow of Ba(t) using the 155 intervals traffic flows of all the 31 links each with 5 intervals t-5,t-4, ..., t-1. In modeling for link Ba, we just consider whether the components of the corresponding column or row in the inverse covariance matrix are zero or not. Fig. 2 shows the sparse graphical model of Ba for predicting the traffic flow Ba(t), in which we ignore the nodes evaluated to have little relevance with Ba(t) by Graphical Lasso. We can see that there are only 8 nodes linked with node Ba(t) in Fig. 2. That is, 8 traffic flows are evaluated relevant to the predicting of Ba(t), which is much less than the original possible 155 traffic flows.

Comparing Fig. 2 with link Ba in traffic map Fig. 1, we can find the predicting of Ba(t) is not only relevant to the three traffic flows Ba(t-3), Ba(t-2), Ba(t-1) on link Ba but also with other five traffic flows Eb(t-1), Fe(t-2), Fe(t-1), Hl(t-1), Ib(t-1) which come from the four links Eb, Fe, Hl and Ib. In other words, the sparse graph modeled by Graphical Lasso, to one certain predicted traffic flow, indicates its relevance with traffic flows on the same link and points out its relevance with traffic flows on all the other links in the whole map as well. This model is superior to all previous approaches which only consider the relevance between intervals on the same link or relevance between links, the model built by Graphical Lasso combines the two traditional approaches.

3.3 Selection of Parameters and Design of Neural Networks

In the Graphical Lasso algorithm, the selection of the penalty parameter is referred to Section 2.3 in [7]. Basing the selected penalty parameter, for finite samples, the probability of error in estimating the graphical model is controlled. More details can be seen in [7]. Since the sparse graphical model we built is based on the inverse covariance matrix, we need a criterion to determine whether there is an effective correlation between each two nodes. In our experiments, we set the component to be 0 while it is less than 5e-4, that is, we think there is little relevance between the two nodes when the corresponding component is so small.

In the design of Neural Networks, we choose a three-layer neural network model for which can approximates arbitrary bounded and continuous function [11]. On the other hand, less layers make the network less complex and then less time-consuming. As we all know, Back Propagation (BP) networks have perfect self-learning ability. Thus, in all our experiments, BP networks with three layers were selected. The transfer and train functions and all the related parameters of the BP networks are all the same as our previous studies [5], for they are the common foundation of all the comparisons.

3.4 Results

According to the model built by Graphical Lasso, we process the dataset to the corresponding form. Still take link Ba as an example, according to Fig. 2, we choose the related eight traffic flows as inputs and the corresponding Ba(t) as the output. Since there are different numbers of related nodes in sparse graphical models for different links, we make different data processing respectively. Then, basing the common foundation of Neural Networks, we compared our experiment with previous multi-link multi-task learning (MMTL) and multi-link single-task learning (MSTL) approaches [5]. The results are shown in Table 1, in which we represent our approach as GL_NN for it models the traffic flow with Graphical Lasso and does the predicting basing on Neural Networks. All the predicted performance is evaluated by root mean square error (RMSE).

From the results shown in Table 1, we can see that, in the total 31 road links, there are 21 links showing our approach GL_NN performs better than MSTL and 20 links even outperforms MMST. In order to highlight the advantages of

GL_NN, we also compared the sum of the RMSE of the whole 31 links corresponding to the three different approaches in Table 2. Uniting the two comparisons in Table 1 and Table 2, we can obviously see the superiority of our approach GL_NN.

Table 1. RMSE of all the 31 road links corresponding to the three approaches

RMSE	MMTL	MSTL	GL_NN
Ba	147.79	150.81	**139.71**
Bb	72.59	73.60	**70.46**
Bc	97.65	98.80	**91.89**
Ce	53.57	54.73	**52.70**
Cf	84.58	86.79	**81.62**
Cg	49.19	49.51	53.27
Ch	63.13	63.48	**64.02**
Da	77.15	82.28	95.47
Db	53.49	54.60	63.75
Dc	87.69	88.32	**73.39**
Dd	65.07	68.61	**55.99**
Eb	165.58	168.14	**150.17**
Ed	199.36	208.95	**179.67**
Fe	119.94	122.73	**112.40**
Ff	83.23	83.88	103.15
Fg	92.40	93.12	**87.67**
Fh	136.23	141.46	144.00
Gb	83.34	83.64	103.03
Gd	155.08	153.39	**144.28**
Hi	87.10	87.23	95.11
Hk	131.61	131.72	158.27
Hl	129.67	130.04	**108.92**
Ia	88.13	88.60	100.65
Ib	129.45	132.83	**124.16**
Id	133.13	135.06	**113.36**
Jh	148.88	148.23	**130.23**
Jf	119.46	120.33	**108.42**
Ka	76.45	75.72	**75.60**
Kb	130.85	134.27	159.13
Kc	378.47	385.35	**365.17**
Kd	161.21	163.50	**159.61**

Table 2. Sum of the RMSE of the 31 road links corresponding to the three different approaches

	MMTL	MSTL	GL_NN
RMSE	3601.47	3659.74	3565.27

4 Conclusions and Future Works

Traditional multi-link traffic flow forecasting just considers the correlation between directly adjacent links and does not extract the correlation information. The approach we proposed efficiently extracts the information provided by adjacent links and considers all the directly or indirectly adjacent links. That is, we consider more comprehensively and efficiently than traditional approaches. Through the comparison with multi-link multi-task learning in traffic flow forecasting, it is shown that our approach has further superiority in traffic flow forecasting.

In the future, further research would take on the combination of Graphical Lasso with other predicting methods, or other applications not just in traffic flow forecasting. In our paper, we verified the high efficiency of modeling with Graphical Lasso. While model building used widely in machine learning, we believe that the Graphical Lasso can also be used efficiently in many other areas.

Acknowledgments. This work is supported in part by the National Natural Science Foundation of China under Project 61075005, 2011 Shanghai Rising-Star Program, and the Fundamental Research Funds for the Central Universities.

References

1. Jordan, M.I.: Graphical Models. Statistical Science 19(1), 140–155 (2004)
2. Murphy, K.P.: An introduction to graphical models. Citeseer (2001)
3. Friedman, J., Hastie, T., Tibshirani, R.: Sparse inverse covariance estimation with the graphical lasso. Biostatistics 9(3), 432–441 (2008)
4. Rich, C., Virginia, R.S.: Benefiting from the variables that variable selection discards. Machine Learning Research 3, 1245–1264 (2003)
5. Gao, Y., Sun, S.: Multi-link traffic flow forecasting using neural networks. In: The 6th International Conference on Natural Computation, vol. 1, pp. 398–401 (2010)
6. Meinshausen, N., Bühlmann, P.: High dimensional graphs and variable selection with the lasso. The Annals of Statistics 34(3), 1436–1462 (2006)
7. Yuan, M., Lin, Y.: Model election and estimation in the Gaussian graphical model. Biometrika 94(1), 19–35 (2007)
8. Banerjee, O., El Ghaoui, L., d'Aspremont, A.: Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. Machine Learning Research 9, 485–516 (2008)
9. Dahl, J., Vandenberghe, L., Roychowdhury, V.: Covariance selection for nonchordal graphs via chordal embedding. Optimization Methods and Software 23(4), 501–520 (2008)
10. Sun, S., Zhang, C.: The selective random subspace predictor for traffic flow forecasting. IEEE Transactions on Intelligent Transportation Systems 8(2), 367–373 (2007)
11. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Citeseer (2001)

Learning Curve Model for Torpedo Based on Neural Network

Min-quan Zhao[1,2], Qing-wei Liang[1], Shanshan Jiang[1], and Ping Chen[2]

[1] The College of Marine, Northwestern Polytechnical University, Xi'an, Shaanxi, China
[2] No. 92785 Unit, People's Liberation Army, Qinhuangdao, Hebei, China
{zh8015,liangqingwei,brendaforever,yangmei0715}@163.com.cn

Abstract. Producing cost of Torpedo decreases with the increase of turnout. Based on the theory of approaching discretional function by three layers BP neural network, a learning curve model for torpedo based on neural network is set up. Result indicates that this model can achieve satisfactory precision. At the same time, it has practical value.

Keywords: Torpedo; BP Neural Networks; Producing Cost; Output; Learning Curve.

1 Introduction

Influenced by factors of society, economy, and so on, all kinds of cost of torpedo are increasing rapidly. How to use the limited financial resources to obtain efficient torpedo becomes more and more important to the developing department, the purchasing department, and the users. To solve this problem, the life cycle cost analysis has been proposed.

Life cycle cost (LCC) is the total cost in the whole life cycle period from research, developing, and production, to use and warranty, till ex-service. The research cost and the ex-service cost is little in all the life cycle cost, so the research cost and the ex-service cost is often not considered, and the life cycle cost of torpedo is divided into developing cost, producing cost, and maintaining cost.

The producing cost is represented by purchasing cost. With more independent of the purchasing department, more competitive of the purchasing course, more marketable of the purchasing mechanism, the purchasing cost should be thought much. No matter the developing department or the purchasing department, the sensitivity to the purchasing cost will boost up remarkably.

2 Learning Curve

The relationship between producing cost and cumulative turnout is often described by learning curve. The learning curve denotes that the producing cost of the first product is the highest, and it will decrease regularly by a certain rate with the increase of the

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 159–164, 2011.

turnout. The rate is high at the beginning, and then slower slowly, at last reach stabilization. That is to say, producing cost will reduced by a certain rate with the increase of turnout. The learning curve denotes the decreasing cost rule of the repeated product when the producing course carries out ceaselessly.

The existing learning curve [1], like Wright formula and Carwford formula, are experiential formula. The first developed formula is Wright formula:

$$C = C_1 N^b \tag{1}$$

where, C is the cumulative average producing cost when the total turnout is N; C_1 is the producing cost of the first product; N is the cumulative turnout; b is learning index.

Carwford formula has the same form with Wright formula, but the meaning of C is unlike:

$$C = C_1 N^b \tag{2}$$

where, C is the producing cost of product; C_1 is the producing cost of the first product; N is the cumulative turnout; b is learning index.

Index b is the slope of the learning curve:

$$b = \log S / \log 2 \tag{3}$$

where, S is the slope of learning, that is, the decreasing rate of producing cost with the increasing of turnout. It can be gained by historical data. The relationship of the two learning curve forms is shown as fig. 1 [2].

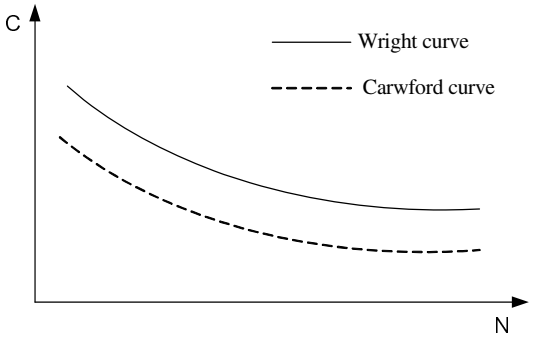


Fig. 1. The relationship of the two learning curve form

The Carwford formula is often modified or developed to get the learning curve by many company of Europe.

The learning curves differ with the aim and the Scopes. It can be gained by the history data of the producing cost, and then the producing cost will be forecasted by the curve.

The relationship between the producing cost and the cumulative turnout is nonlinear, and neural network is very fit for disposing nonlinear relation. So, neural network is adopted to set up the model of producing cost and cumulative turnout for torpedo.

3 Neural Network Model

Neural network [3][4] is directional map composed by many neural nerve cells. It imitates the structure of the biological neural system and has the ability of learning by itself and adapting by itself. The knowledge which learned by itself hide in the structure of the net, and it needn't to get distinct formula. So the disposal mode of neural network to the complicated nonlinear system has essential difference compared with the traditional method.

The neural cell is the basic constitution (shown as fig. 2). A neural cell has three elements: (1) a set of connection weight, which shows the connection intensity. Positive weight shows stimulant, and negative weight shows restrained; (2) an adding cell, which is used as the weighted sum of the input information; (3) a nonlinear stimulant function, whose function is nonlinear mapping, and confining the output signal in a certain region (within [0,1] or [-1,+1]). Besides, there is a threshold. There are many representative neural network models, such as Perceptron, BP neural network, RBF neural network, Self-Organizing network, Feedback network [5].

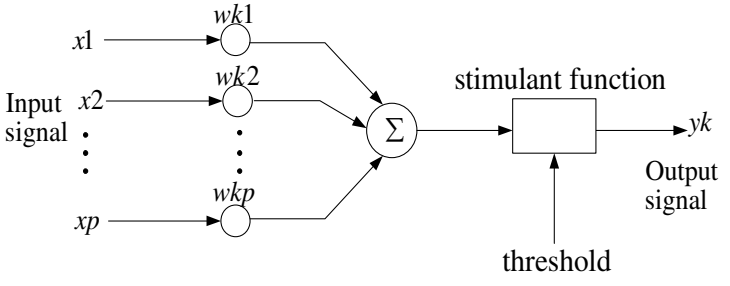


Fig. 2. Neural cell model

BP neural network is a kind of multilayer neural network, and it is consisted by input layer, hidden layer, output layer. Error back propagation learning algorithm is adopted to adjust the weight. The transform function of nerve cell is Sigmoid function:

$$\varphi(v) = 1/\left(1+\exp\left(-av\right)\right) \tag{4}$$

whose output signal is continuous value within 0 and 1. When the structure of the BP neural network is confirmed, the BP neural network is trained by the sample of input and output. That is to say, the weight and the threshold are trained and adjusted, so the neural network can achieve the arbitrary nonlinear mapping of input signal and output signal. The BP neural network, which has been trained, can gain the output signal no mater the input signal is in the sample or not. So the BP neural network has the strong ability of generalization. BP neural network is the clearest understood and the

broadest applied neural network by now, so it is the most important model of neural network. This paper adopts BP neural network to set up learning curve model.

4 The Learning Curve Model Based on Neural Network

The relationship between the producing cost and the cumulative turnout is nonlinear. Three layers BP neural network can approach arbitrary function, and the precision can be controlled. So it is adopted to set up the model of the learning curve for torpedo.

The principle of the learning curve model for torpedo based on neural network is: the known cumulative turnout is adopted as the input signal of the BP neural network, and the producing cost is adopted as the output signal. The neural network is trained by enough samples (the more the samples are, the more accurate the neural network will be). The output signal will be distinct with the distinct input signal. The weight and the threshold of the trained neural network are denotation of the relationship between producing cost and cumulative turnout. Every layer is buildup by several neural cells. The neural cells in the same layer have no connection with each other, and the neural cells in different layer have complete connection with each other. The training course is buildup by the onwards propagation and backwards propagation. In the course of the onwards propagation, the input signal spread from input layer, pass hidden layer, then reach the output layer. The state of each layer only influents the state of next layer. If the expected result is not gotten in the output layer, then the neural network turns to the backward propagation. The error of the output signal will be feedback by the primary access. The weight of every layer is adjusted to make the error be least. Then the trained weight and the threshold will be gotten and can be used. Taking the cumulative turnout as the input signal, the output signal of producing cost will be gotten by onward propagation.

5 Example

Certain torpedo is produced 7 groups, the recorded producing cost and the corresponding cumulative turnout is shown in table 1. Considering time value, the cost has been converted to 1990 money year. The learning curve model for torpedo based on neural network should be set up.

Table 1. Producing Cost and the Corresponding Cumulative Turnout Unit:10 000 RMB￥, (first fiscal year)

Groups	1	2	3	4	5	6	7
Cumulative Turnout	10	25	45	80	120	180	240
Producing Cost	440	430	410	400	390	386	383

Training the model with the date of former 5 groups, and validate the model with the date of the 6th, 7th groups. Because the BP neural network needs large numbers of training sample, the median method is adopted to enlarge the number of the sample. The model is validated time after time. 3 is selected as the node number of hidden

layer, Sigmoid is selected as the transfer function of input layer and hidden layer, liner transfer function is selected as the transfer function of output layer. The weight and the threshold are adjusted by the BP arithmetic, whose study ratio is variable. The maximal study number is set as 5000, the study ratio is set as 0.001, and the study aim is set as error sum of square, premnmx function is used to make the date of sample unitary, so the date of sample will within [-1,1].

Matlab [6] is used as the emulational tool. The constringency of the model can be gotten. The net pinches at 4324 step, and the anticipant error is satisfied. The effect of training is show as fig.3. After the training is accomplished, the date of the 6th, 7th groups is emulated, and the result is in table 2.

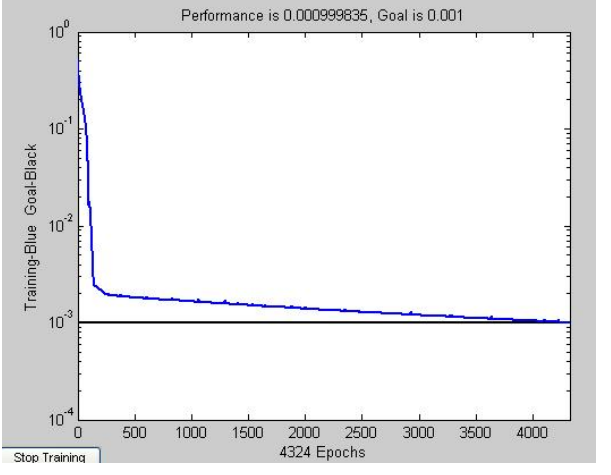


Fig. 3. The effect of training

Because the torpedo is build up group by group, the average cost of the fist group is regarded as the first producing cost C_1, and the estimate of Carwford curve can be get from the cumulative turnout of 2th group, the 3th group, the 4th group, the 5th group with the cost of them.

$$C = 440 * N^{-0.02} \tag{5}$$

The producing cost of 6th, 7th group can be predicted according to the Carwford curve. The result is in table 2.

Table 2. The contrast of the prediction of 6th, 7th group

Groups	Cumulative Turnout	Producing Cost			Relative Error	
		Real value	Prodiction of Carwford curve	Prodiction of Neural Network	Carwford Curve	Neural Network
6	180	386	396.5950	385.4933	0.0301	0.0013
7	240	383	394.3197	378.3230	0.0300	0.0122

From Table. 2 we can see, for the date of the 6th, 7th groups, the average of the relative error of Carwford curve is 0.0300, somewhat bigger than the average of the relative error of BP neural network, 0.0068.

6 Conclusion

Neural network is adopted to set up the learning curve model for torpedo. The essential is to find the nonlinear relationship between the producing cost and the cumulative turnout. Using the trained weight and threshold, more accurate result will be gained than parameter method. So using the neural network to set up learning curve model is a new way for those problems.

References

1. Wu, Z.: The guiding role of learning curve theory for aerospace industry. The Manufacture for Aerospace Industry 7, 13–14 (1996)
2. Zhao, Y., Lou, S., Zhang, Y.: The prediction model and analysis of missile cost. The Theory and the Practice of the System Engineering 9, 117–121 (2003)
3. Zhou, Z., Cao, C.: Neural network and applications. Qinghua university publishing company, Beijing (2004)
4. Li, X., Zhang, X.: A introduction to neural network and neural computers. Polytechnical university publishing company, Xi'an (1995)
5. Zhang, N., Yan, P.: Neural network and fuzzy control. Qinghua university publishing company, Beijing (1998)
6. Lou, S., Shi, Y.: The system analysis and design based on MATLAB – neural network. Xian electron science and technology university publishing company, Xi'an (2000)

An Efficient EM Approach to Parameter Learning of the Mixture of Gaussian Processes

Yan Yang and Jinwen Ma*

Department of Information Science,
School of Mathematical Sciences & LMAM Peking University,
Beijing, 100871, P.R. China

Abstract. The mixture of Gaussian processes (MGP) is an important probabilistic model which is often applied to the regression and classification of temporal data. But the existing EM algorithms for its parameter learning encounters a hard difficulty on how to compute the expectations of those assignment variables (as the hidden ones). In this paper, we utilize the leave-one-out cross-validation probability decomposition for the conditional probability and develop an efficient EM algorithm for the MGP model in which the expectations of the assignment variables can be solved directly in the E-step. In the M-step, a conjugate gradient method under a standard Wolfe-Powell line search is implemented to learn the parameters. Furthermore, the proposed EM algorithm can be carried out in a hard cutting way such that each data point is assigned to the GP expert with the highest posterior in the E-step and then the parameters of each GP expert can be learned with these assigned data points in the M-step. Therefore, it has a potential advantage of handling large datasets in comparison with those soft cutting methods. The experimental results demonstrate that our proposed EM algorithm is effective and efficient.

Keywords: Mixture of Gaussian processes, Leave-one-out cross-validation, EM algorithm, Conjugate gradient method.

1 Introduction

As an extension of the mixture of experts (ME) architecture, the mixture of Gaussian processes (MGP) is a combination of several single Gaussian processes by a gating network. With the help of the divide-and-conquer strategy, the MGP model is more flexible for modeling a temporal dataset than a single Gaussian process. Moreover, Gaussian process has been shown to have a good performance on regression and classification. However, just as many other powerful tools, Gaussian process is not so perfect and has two main limitations. First, a Gaussian process has a stationary covariance function and this characteristic cannot be adapted in the cases of temporal datasets which have varying noises in different times. Second, the computational cost of the parameter learning or inference is

* Corresponding author, jwma@math.pku.edu.cn

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 165–174, 2011.

very high since it is involved in the computation of the inversion of an $n \times n$ matrix where n is the number of the training dataset.

Since the MGP model was firstly investigated by Tresp [11], there have appeared some variants of the MGP model and the corresponding learning methods have been established ([3-4], [12], etc.). For clarity, we summarize all these investigations from two aspects: gating network and inference method. In fact, there are three kinds of gating networks in the literature. Firstly, as the MGP model is inspired by the ME model, the gating network of the ME model can be straightforward inherited [7, 9, 11]. The second kind of gating network is just a set of mixing coefficients which are assumed to follow a Dirichlet distribution [6] or be generated from a Dirichlet process [4] (in this case, the finite mixture model can be generalized to an infinite one). The third kind of gating network is based on the distribution of the data points from the input space. In this situation, data points from one GP expert space are assumed to be subject to a Gaussian distribution [3], or a Gaussian Mixture distribution [12].

With the diversity of gating networks, there have developed two main inference methods: the Bayesian inference method and the non-Bayesian parameter estimation method. By the Bayesian inference method, all the parameters are assumed to have some prior distributions and certain sophisticated techniques like the Markov Chain Monte Carlo methods are used for the parameter learning or estimation [3-4], [12]. On the other hand, since the well-known EM algorithm has been successfully implemented to learn the ME model [1-2], several implementations of the EM algorithm have been proposed to learn the parameters of the MGP model (e.g., [7], [9], [11]). However, since the outputs of the MGP model are not independent as those of the ME model, it becomes a very difficult problem to compute the posterior probability that a data point belongs to each GP expert. Actually, the computation schemes of the posterior probabilities in the existing EM algorithms are heuristic, in lack of theoretical proofs, and often lead to a low efficiency.

In this paper, in order to solve this difficult problem more efficiently, we utilize the leave-one-out cross-validation probability decomposition for these conditional probabilities and develop an efficient EM algorithm for the MGP model in which the expectations of the assignment variables can be computed directly. In fact, the leave-one-out cross-validation probability decomposition was already used for the parameter learning in the single GP model [5], [10], but it has not been used for the parameter learning of the MGP model. Here, as the conditional probability of the output with respect to the input and the parameters is expressed by the leave-one-out cross-validation probability decomposition, we can get a novel expression of the posterior probability that each data point belongs to a GP expert in the E-step. In the M-step, we implement a conjugate gradient method under a standard Wolfe-Powell line search to maximize the log likelihood with the gradients being computed via the expressions given by Sundararajan et al. [10].

As compared with the Bayesian inference methods, the existing EM algorithms must use all the data points for inferring each GP expert. This may

cause a great computation cost in dealing with a large dataset. To get rid of this difficulty, we further modify the proposed EM algorithm in a hard cutting way by assigning every data point to the GP expert with the highest posterior in the E-step. Then, in the M-step, only these assigned data points are used to learn the parameters of each expert. Therefore, the modified EM algorithm is more adapted to deal with the learning problem of a large dataset. To demonstrate the proposed algorithms in this situation, we conduct experiments on the motorcycle dataset.

The remainder of this paper is organized as follows. In Section 2, we introduce the MGP model and the leave-one-out cross-validation probability decomposition. The new EM algorithm is derived and investigated in Section 3, with the experimental results being illustrated in Section 4. In Section 5, we make a brief conclusion.

2 MGP and Leave-One-Out Cross-Validation Probability Decomposition

We begin with a brief introduction to the Gaussian Process according to the work by Rasmussen and Williams [5]. Give a set of training data $X = [x_1^T, \cdots, x_n^T]^T$ as inputs and $Y = [y_1^T, \cdots, y_n^T]^T$ as the corresponding outputs, where n is the number of the training data. This dataset is said to follow a Gaussian Process if $Y \sim \mathcal{N}(m(X), K_y(X, X))$, where $m(X)$ is a prior defined mean function and $K_y(X, X)$ is a covariance matrix function with its element $K_y(x_p, x_q)$ being a kernel function. For simplicity, we assume that the mean function $m(X)$ is zero. There are some varying forms for the covariance function and here we use the common one named the *squared exponential* (SE) covariance function as follows:

$$K_y(x_p, x_q) = l^2 \exp\{-\frac{\sigma_f^2}{2}||x_p - x_q||^2\} + \delta_{pq}\sigma_n^2, \tag{1}$$

where l, σ_f and σ_n are nonzero real values. $\delta_{pq} = 1$ if $p = q$; otherwise, $\delta_{pq} = 0$.

The MGP model, as an extension of mixture of experts (ME) architecture, is a combination of several single Gaussian processes by a gating network $g(x|\phi)$, where ϕ denotes the set of all the parameters in the gating network. The gating network aims to divide the input space into regions for specific Gaussian processes making predictions. As described in [3], we assume that data points in the input space are i.i.d. and those from the same GP expert are Gaussian distributed.

Suppose that there is a training dataset $\{Y, X\} = \{y_t, x_t\}_{t=1}^N$ being generated from a mixture of Gaussian processes containing M single components. The covariance matrix K_j of the j-th GP component is specified by the parameters $\theta_j = \{l_j, \sigma_{fj}, \sigma_{nj}\}$ and each Gaussian component in the input space (i.e., $\mathbb{R}^d$) is specified by the parameters $\phi_j = \{\nu_j, \Sigma_j\}$. Let Y_{-t} and X_{-t} be the corresponding datasets leaving out y_t and x_t, respectively. The leave-one-out cross-validation

probability decomposition can be given by

$$p(Y, X, \Theta) = \prod_{t=1}^{N} \sum_{j=1}^{M} \alpha_{tj} p(y_t | x_t, Y_{-t}, X_{-t}, A, \theta_j) p(x_t | \phi_j), \tag{2}$$

where $A = \{\alpha_{tj}\}$, α_{tj} is the probability that (y_t, x_t) belongs to the j-th component, under the constraint that $\sum_{j=1}^{M} \alpha_{tj} = 1$. In our consideration, the gating network is set by $g(x|\phi) = [p(x|\phi_1), \cdots, p(x|\phi_M)]^T$. Specifically, we have

$$p(x_t|\phi_j) = \frac{1}{(2\pi)^{d/2}|\Sigma_j|^{1/2}} \exp\{-1/2(x_t - \nu_j)^T \Sigma_j^{-1}(x_t - \nu_j)\}. \tag{3}$$

For any pair of y_p and y_q in given Y_{-t}, the covariance of them can be written as $K(x_p, x_q) = Cov(y_p, y_q) = \sum_{i=1}^{M} \alpha_{pi}\alpha_{qi} K_i(x_p, x_q)$, where K_i is the covariance function of the i-th GP component. Under the assumption that y_t belongs to the j-th component, the covariance of y_t and any y_p in Y_{-t} is $Cov(y_p, y_t) = \alpha_{pj} K_j(x_p, x_t)$. Therefore, we have

$$\begin{bmatrix} Y_{-t} \\ y_t \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X_{-t}, X_{-t}) & \beta_{tj} \\ \beta_{tj}^T & K_j(x_t, x_t) \end{bmatrix}\right)$$

where $\beta_{tj}(x_p, x_t) = \alpha_{pj} K_j(x_p, x_t)$. Hence, we further get

$$p(y_t | x_t, Y_{-t}, X_{-t}, A, \theta_j) \sim \mathcal{N}(\mu_{tj}, \sigma_{tj}^2) \tag{4}$$

where

$$\mu_{tj} = \beta_{tj}^T K(X_{-t}, X_{-t})^{-1} Y_{-t}, \tag{5}$$

$$\sigma_{tj}^2 = K_j(x_t, x_t) - \beta_{tj}^T K(X_{-t}, X_{-t})^{-1} \beta_{tj}. \tag{6}$$

Until now we have specified the leave-one-out cross-validation probability decomposition (2), and in the following analysis we will try to maximize it under an EM framework.

3 Proposed EM Algorithm for MGP

Let $\{Y, X\} = \{y_t, x_t\}_{t=1}^{N}$ be a dataset drawn from a MGP model which contains M components, where N is the number of training data points. In order to carry out the EM algorithm for MGP, we first consider a set of binary variables $Z = \{z_{tj}\}$ such that $z_{tj} = 1$, if (y_t, x_t) is drawn from the j-th GP expert; otherwise, $z_{tj} = 0$. Obviously, $\sum_{j=1}^{M} z_{tj} = 1$. Let Y^j and X^j denote the output and input data points of the j-th GP, and Y_{-t}^j and X_{-t}^j be the corresponding datasets leaving out y_t and x_t, respectively.

Suppose that all the values of the binary variables $Z = \{z_{tj}\}$ are known, the joint probability eqn. (2) can be written as

$$p(Y, X|\Theta, \Phi) = \prod_{t=1}^{N} \prod_{j=1}^{M} (\alpha_{tj}(p(x_t|\phi_j)))^{z_{tj}} p(y_t | x_t, X_{-t}, Y_{-t}, Z, \theta_j), \tag{7}$$

where $p(y_t|x_t, X_{-t}, Y_{-t}, Z, \theta_j)$ is obtained by replacing α_{ti} with z_{tj} in eqn. (4). We then get the log likelihood function as follows:

$$l_0(\Theta, \Phi; Y, X, Z) = \sum_{t=1}^{N} (\log p(y_t|x_t, X_{-t}, Y_{-t}, Z, \Theta) + \sum_{j=1}^{M} z_{tj} \log \alpha_{tj} p(x_t|\phi_j)). \quad (8)$$

In this situation, the missing data are the hidden variables Z, the observed data are $\{Y, X\}$ and l_0 is the log likelihood of the complete data which we aim to maximize. We further define a so-called Q function as the expectation of the log likelihood w.r.t. the missing data Z:

$$Q(\Theta, \Phi|\Theta^{(k)}, \Phi^{(k)}) = E_Z\{l_0(\Theta^{(k)}, \Phi^{(k)}; X, Y, Z)\}, \quad (9)$$

where $\Theta = \{\theta_j\}$, $\Phi = \{\phi_j\}$. In the EM framework, we actually do not maximize the log likelihood function directly. Instead, we try to maximize the Q function. In order to compute the Q function, we need to compute the posteriors of each data point (y_t, x_t) belonging to the j-th GP expert, denoted by $h_j(t)$:

$$h_j(t) = \frac{\hat{\alpha}_{tj}^{(k)} p(y_t|x_t, Y_{-t}, X_{-t}, \hat{\alpha}_{tj}, \theta_j^{(k)}) p(x_t|\hat{\nu}_j^{(k)}, \hat{\Sigma}_j^{(k)})}{\sum_{l=1}^{M} \hat{\alpha}_{tl}^{(k)} p(y_t|x_t, Y_{-t}, X_{-t}, \hat{\alpha}_{tj}, \theta_l^{(k)}) p(x_t|\hat{\nu}_l^{(k)}, \hat{\Sigma}_l^{(k)})}. \quad (10)$$

As z_{tj} is replaced by $h_j(t)$ in l_0, we get

$$Q = \sum_{t=1}^{N} \log p(y_t|x_t, X_{-t}, Y_{-t}, \{h_j(t)\}, \hat{\Theta}) + \sum_{t=1}^{N} \sum_{j=1}^{M} h_j(t) \log \hat{\alpha}_{tj} p(x|\hat{\phi}_j). \quad (11)$$

Since there are no common parameters in the first and second terms of the Q function, we can deal with the maximization of the Q function on the two terms independently. We can find an analytical solution to the maximization of the second term by taking the derivatives to zero, that is,

$$\hat{\alpha}_{tj} = h_j(t), \qquad \hat{\nu}_j = \frac{1}{\sum_{t=1}^{N} h_j(t)} \sum_{t=1}^{N} h_j(t) x_t, \quad (12)$$

$$\hat{\Sigma}_j = \frac{1}{\sum_{t=1}^{N} h_j(t)} \sum_{t=1}^{N} h_j(t)(x_t - \hat{\nu}_j)(x_t - \hat{\nu}_j)^T. \quad (13)$$

For convenience, we denote the first term of the Q function as Q_1. It is rather difficult to find a an analytical solution to the maximization of Q_1. Here we try to develop a numerical method to get a maximum of Q_1 via the conjugate gradient method for a standard Wolfe-Powell line search. To implement the conjugate gradient method, we first need to get the derivatives w.r.t. the parameters $\theta_j = \{l_j, \sigma_{fj}, \sigma_{nj}\}_{j=1}^{M}$.

As investigated by Sundararajan et al. [10], Q_1 can be written as: $Q_1 = -\sum_{t=1}^{N}((y_t - \mu_t)^2/(2\sigma_t^2) + \log \sigma_t^2/2 + \log(2\pi)/2)$, where the predictive mean μ_t

and variance σ_t^2 can be expressed as $\mu_t = y_t - [K^{-1}Y]_t/[K^{-1}]_{tt}$, $\sigma_t^2 = 1/[K^{-1}]_{tt}$, and the notations $[\cdot]_t$, $[\cdot]_{tt}$ stand for the tth element of the specified vector and the tth diagonal element of the specified matrix, respectively. K is the covariance function defined by $K(x_p, x_q) = \sum_{j=1}^{M} h_j(p)h_j(q)K_j(x_p, x_q)$. Its gradient can be given by

$$\frac{\partial Q_1}{\partial \theta_j} = \sum_{t=1}^{N} \frac{\alpha_t [Z_j \alpha]_t}{[K^{-1}]_{tt}} - \frac{\alpha_t^2 [Z_j K^{-1}]_{tt}}{2[K^{-1}]_{tt}^2} - \frac{[Z_j K^{-1}]_{tt}}{2[K^{-1}]_{tt}}, \tag{14}$$

where $\alpha = K^{-1}Y$, $Z_j = K^{-1}\partial K/\partial\theta_j$ and $\partial K/\partial\theta_j = h_j(p)h_j(q)\partial K_j/\partial\theta_j$. According to the definition of the covariance function eqn. (1), we have

$$\frac{\partial K_j(p,q)}{\partial l_j} = 2l_j \exp\{-\frac{\sigma_{fj}^2}{2}||x_p - x_q||^2\}, \qquad \frac{\partial K_j(p,q)}{\partial \sigma_{nj}} = 2\delta_{pq}\sigma_{nj},$$

$$\frac{\partial K_j(p,q)}{\partial \sigma_{fj}} = -\sigma_{fj}||x_p - x_q||^2 l_j^2 \exp\{-\frac{\sigma_{fj}^2}{2}||x_p - x_q||^2\}.$$

With the above preparations, we now give our new EM algorithm for MGP as follows.

1. Initialize the parameters $\{\alpha_{tj}\}$, $\{\theta_j\}$, $\{\phi_j\}$.
2. Calculate the posteriors according to eqn. (10).
3. Calculate $\hat{\alpha}_{tj}$, $\hat{\nu}_j$, $\hat{\Sigma}_j$ according to eqn. (12, 13). Calculate θ_j by maximizing Q_1 using a conjugate gradient method under a standard Wolfe-Powell line search.
4. Repeat step 2-4, until convergence.

In the E-step, we compute the posteriors by eqn. (10). In the M-step, we estimate the parameters α_{tj}, $\hat{\nu}_j$, $\hat{\Sigma}_j$ by eqn. (12 & 13) and θ_j by implementing the conjugate gradient method with the help of eqn. (14). Repeat the two steps until convergence.

As an disadvantage of the non-Bayesian methods [11] in comparison with the Bayesian methods [3-4],[12], the computation complexity problem is so knotty that they require the inverse of an $n \times n$ matrix for every GP expert, where n is the number of the training data points. In order to overcome this complexity problem, we can modify our proposed EM algorithm in a hard cutting mode. That is, in the E-step, after getting all the posteriors, we assign each data point to the GP expert with the largest posterior. In the M-step, only the assigned data points are used for learning each GP expert. In such a way, the computation complexity of the modified hard cutting EM algorithm is reduced considerably.

4 Experimental Results

To test the performance of our proposed EM algorithm for MGP, we conduct some experiments on an artificial toy dataset given in [3] and the motorcycle dataset given in [8]. The artificial toy dataset consists of four continuous functions which have different levels of noise. The four continuous functions are:

$f_1(a_1) = 0.25a_1^2 - 40 + \sqrt{7}n_t$, $f_2(a_2) = -0.0625(a_2 - 18)^2 + 0.5a_2 + 20 + \sqrt{7}n_t$, $f_3(a_3) = 0.008(a_3 - 60)^3 - 70 + \sqrt{4}n_t$, $f_4(a_4) = -\sin(a_4) - 6 + \sqrt{2}n_t$, where $a_1 \in (0, 15)$, $a_2 \in (35, 60)$, $a_3 \in (45, 80)$, $a_4 \in (80, 100)$ and $n_t \sim \mathcal{N}(0, 1)$ that denotes a standard Gaussian distribution (with zero mean and variance 1). We generate 200 samples (50 samples for each function) from this toy model. We apply a mixture of four Gaussian Processes to model this dataset and implement the EM algorithm to learn the parameters of the mixture. The experimental results are shown in Figure 1. The noise values of each expert learned by our proposed EM algorithm are very close to the true ones: 7.04, 6.69, 3.98, 1.59. In the input space the centroids of the experts are 7.28, 46.65, 64.22, 90.90.

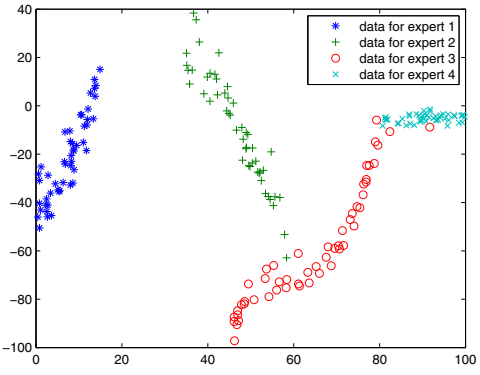
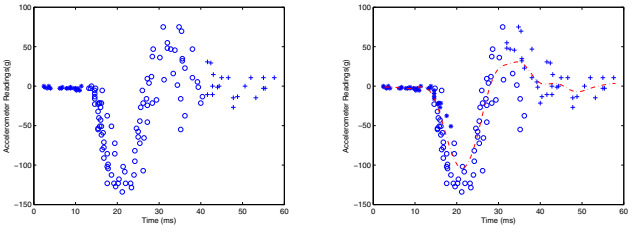


Fig. 1. Experimental results of the proposed EM algorithm on the toy dataset. The notations '*', '+', 'o', 'x' represent samples from four expert.

The motorcycle dataset consists of 133 observations of accelerometer readings taken through time. These observations belong to three strata and we present them in terms of intervals along the time axis: [2.4, 11.4], (11.4 40.4] and (40.4, 57.6]. In the left plot of Figure 2, we illustrate the dataset and denote those belonging to the same stratum by notations 'o', '*' and '+', respectively.

In this case, we set the number of GP experts as 3, and then implement the proposed EM algorithm for MGP on the dataset. For convenience, we begin to initialize the posteriors rather than the parameters, as in the MGP model, the prediction task is impossible to be done only with the parameters. In the M-step, the conjugate gradient method under a standard Wolfe-Powell line search is applied to estimating the parameters in the GP experts. In this situation, the conjugate gradient method is considered to get a maximum solution when the absolute values of the derivatives w.r.t all the parameters are less than 0.01. We repeat the E-step and the M-step until convergence. In this particular case we stop the algorithm as long as the average norm of the difference of the parameters in the latest two iterations is less than 0.1.

We list the estimated parameters learned by the proposed EM algorithm in Table 1. It shows clearly that three GP experts divide the input space and model the corresponding data points, respectively. They have different degree of noises



(a) (b)

Fig. 2. (a) Three strata of the motorcycle dataset denoted by 'o', '*' and '+'. (b) Clustering result by the proposed EM algorithm for MGP. Three clusters are denoted by the notations 'o', '*' and '+'. We illustrate the predictive medians by the dash-dot line '-·', with 100 samples at each of the 84 equispaced locations according to the posterior distribution.

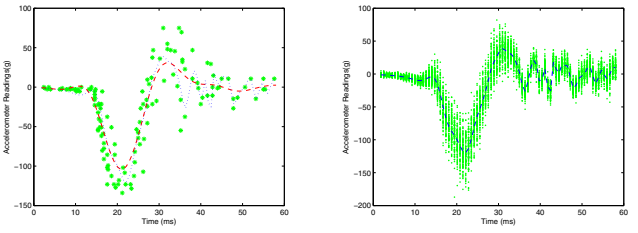
Table 1. The parameters of the MGP model on the motorcycle dataset estimated by the EM algorithm

	l	σ_f	σ_n	$\hat{\mu}$	$\hat{\sigma}^2$
GP expert 1	0.937	0.139	1.106	8.916	15.653
GP expert 2	30.902	0.319	24.212	23.060	53.370
GP expert 3	13.719	1.218	7.833	42.470	73.352

(σ_n^2) according to the varying intervals. The GP expert 1 mainly model the data points at the beginning of the dataset where the data points seem flat. Therefore, the noise in this area learned by the EM algorithm is much smaller than those in the other areas.

To show the flexibility of our leave-one-out cross-validation MGP model, we illustrate the predictive median of the predictive distribution using a dotted line on the left plot in Figure 3. Meanwhile, we use a dashed line to represent the median of the predictive distribution of a single stationary covariance GP model. According to the difference between the two lines, we can observe that the dotted line performs better especially in the intervals where time < 12ms and > 40ms. As speculated in [3], our model also performs well by not inferring an "flat" GP expert [4] at the beginning of the dataset where time < 11ms. In the interval where time > 45ms, the data points are not as dense as those around at 30ms. As compared with the prediction in this interval in [3], the predictive mean of our model almost passes through every data point. The experimental result shows that our leave-one-out cross-validation model may be in more agreement with the idea of the Gaussian process regression that the more closer in the input space the more closer in the output space. The right plot in Figure 3 shows a set of samples drawn from the predictive distribution. We select 84 equispaced locations along the time axis and draw 100 samples at each location.

We further apply the modified hard cutting algorithm on the motorcycle dataset and illustrate the result in the right plot in Figure 2. Three clusters



(a) (b)

Fig. 3. (a) Two medians of the predictive distributions based on the single stationary covariance GP model (dashed line) and the proposed leave-one-out cross-validation MGP model (dotted line). (b) A sample data drawn from the predictive distribution based on the leave-one-out cross validation MGP model. 20 samples for each of the equispaced 84 locations.

are denoted by three notations 'o', '*' and '+'. The dash-dot line represents the predictive median of the posterior distribution. We draw 100 samples from the posterior distribution at each of the 84 equispaced locations. We can see from the clustering results that, the hard cutting method performs very well except for the area where data points from different clusters are very close. We find in the experiment that the clustering results and the computation time are sensitive to the initialization. A proper initialization would lead to a stable performance and a short computation cost. In this case, we assign three GP experts with 15 data points (about ten percent of the training dataset) from the beginning, the middle and the end of the dataset, respectively, and all the remaining data points are assigned randomly. The total computation time is around 34s, in comparison with one hour in [4] and 20s in [12]. Notice that we use the whole 133 data points on the training, but only 40 data points were used for training in [12]. As for any other dimensional input space, the utilization of the information from a rough segment of the input space for initialization will help a lot to achieve good performance and cut down computation cost.

5 Conclusions

We have established an efficient EM algorithm to learn the MGP model. By utilizing the leave-one-out cross-validation probability decomposition, we efficiently compute the posteriors of data points belonging to each GP expert and make the Q function into two independent terms. We further modify the algorithm by assigning data points to the GP expert with the highest posterior and learning each GP expert with the assigned data points. This modification would lead the proposed EM algorithm to the learning problems with large datasets. The experimental results show the efficiency of the EM algorithm and a competitive computation time of the modified hard cutting EM algorithm.

Acknowledgments

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

References

1. Jordan, M.I., Jacobs, R.A.: Hierarchies mixtures of experts and the EM algorithm. Neural Computation 6, 181–214 (1994)
2. Jordan, M.I., Xu, L.: Convergence Results for the EM Approach to Mixtures of Experts Architectures. Neural Computation 8(9), 1409–1431 (1995)
3. Meeds, E., Osindero, S.: An Alternative Infinite Mixture of Gaussian Process Experts. Advances in Neural Information Processing System 18, 883–890 (2006)
4. Rasmussen, C.E., Ghahramani, Z.: Infinite Mixtures of Gaussian Process Experts. Advances in Neural Information Processing System 14, 881–888 (2002)
5. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. MIT Press, Cambridge (2006)
6. Shi, J.Q., Murray-Smith, R., Titterington, D.M.: Bayesian Regression and Classification Using Mixtures of Gaussian Processes. International Journal of Adaptive Control and Signal Processing 17(2), 149–161 (2003)
7. Shi, J.Q., Wang, B.: Curve Prediction and Clustering with Mixtures of Gaussian Process Functional Regression Models. Statistics and Computing 18, 267–283 (2008)
8. Silverman, B.W.: Some aspects of the spline smoothing approach to non-parametric regression curve fitting. Journal of the Royal Statistical Society. Series B 47(1), 1–52 (1985)
9. Stachniss, C., Plagemann, C., Lilienthal, A., Burgard, W.: Gas distribution modeling using sparse Gaussian process mixture models. In: Proc. of Robotics: Science and Systems (RSS), Zurich, Switzerland (2008)
10. Sundararajan, S., Keerthi, S.S.: Predictive Approaches for Choosing Hyperparameters in Gaussian Processes. Neural Computation 13(5), 1103–1118 (2001)
11. Tresp, V.: Mixtures of Gaussian Processes. Advances in Neural Information Processing System 13, 654–660 (2001)
12. Yuan, C., Neubauer, C.: Variational Mixture of Gaussian Process Experts. Advances in Neural Information Processing System 21, 1897–1904 (2008)

Boundary Controller of the Anti-stable Fractional-Order Vibration Systems

Yanzhu Zhang, Xiaoyan Wang, and Yanmei Wang

College of Information Science and Engineering, Shenyang ligong University, Shenyang, China
syzd710471@sina.com

Abstract. This paper discusses two models of the anti-stable vibration system. The anti-stable vibration system can be expressed the integer wave model and the fractional wave model. Many of engineering physical phenomenons can be modeled more accurately and authentically using the fractional order differential equations. The fractional wave equation is obtained from the standard integer wave equation by replacing the first-order time derivative with a fractional derivative of order *b*. The boundary controller of the two models of string vibration systems will be considered. This paper presents a boundary control method of the anti-stable fractional-order vibration systems. Numerical simulations are used to illustrate the improvements of the proposed control method for the fractional vibration systems.

Keywords: fractional calculus, adaptive control, boundary control.

1 Introduction

Time and space fractional wave equations have been considered by several authors for different purposes[1][2]. One of physical purpose for adopting and investigating wave equations is to describe vibration systems. Fractional calculus is the field of mathematical analysis, which deals with the investigation and applications of integrals and derivatives of arbitrary order, which can be real or complex derivatives and integrals to arbitrary orders are referred to as differ integrals by many authors. It starts to play an important role in many branches of science during the last three decades. It is well known that many of engineering physical phenomenons can be modeled more accurately and authentically using the fractional order differential equations[3][4].

In this paper, it is mainly discussed that two models of the string vibration system. The string vibration system can be expressed the integer wave model and the fractional wave model. The fractional wave equation is obtained from the standard integer wave equation by replacing the first-order time derivative with a fractional derivative of order *b*. The boundary controller of the two models of string vibration systems will be considered. This paper presents a boundary control method of anti-stable vibration systems. Numerical simulations are used to illustrate the improvements of the proposed control method for the fractional vibration systems.

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 175–181, 2011.

2 Theory of the Anti-stable String Vibration System

It is well known that a sting vibration system can be governed by the integer-order wave equation, fixed at one end, and stabilized by a boundary control at the other end, The system can be represented by:

$$\begin{aligned} &u_{tt}(x,t) = u_{xx}(x,t) \\ &u_x(0,t) = -a_1 u_t(0,t) \\ &u_x(1,t) = f_1(t) \end{aligned} \tag{1}$$

where $f(t)$ is the boundary control force at the free end of the string and $u(x,t)$ is the displacement of the string, a is a constant parameter.

For the different value of a, equation (1) can model different string vibration systems. For $a = 0$, equations (1) model a string which is free at the end x = 0 and is actuated on the opposite end. For $a < 0$, the system (1) model a string which is fixed at one end , and stabilized by a boundary controller $u = -ku(1,t), k > 0$ at the other end. In this paper we study the string system with $a > 0$, the system (1) model a string which is fixed at one end, and the free end of the string is negatively damped, so that all eigen values located on the right hand side of the complex plane, so the open-loop plant of this kind of string system is "anti-stable". So we call the equation (1) as the model of the anti-stable string vibration systems.

Many authors found the anti-stable string vibration systems can be modeled more accurately and authentically using the fractional order differential equations. The anti-stable fractional-order wave equations are obtained from the classical wave equations by replacing the second order time derivative term by a fractional order derivative (1,2),so the system (1) can be represented by:

$$\begin{aligned} &\frac{\partial^\alpha u(x,t)}{\partial t^\alpha} = \frac{\partial^2 u(x,t)}{\partial x^2} = u_{xx}(x,t) \qquad 1 < \alpha < 2 \\ &u_x(0,t) = -a_2 u_t(0,t) \\ &u_x(1,t) = f_2(t) \end{aligned} \tag{2}$$

The definitions of fractional derivative include Riemann-Liouville,Grunwald-Letnikow,Weyl,Caputo,Marchaud,and Riesz fractional derivatives[5]. Here, we adopt the Caputo definition for fractional derivative of any function $f(t)$, for m to be the smallest integer that exceeds α , the Caputo fractional derivative $\alpha > 0$ is define as:

$${}_0^C D_t^\alpha f(t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{f^{m+1}(\tau)}{(t-\tau)^\alpha} d\tau \tag{3}$$

Where Γ is the gamma function, ${}_0^C D_t^\alpha f(t)$ is the fractional derivative of order α of $f(t)$.

Based on the definition of (5), the Laplace transform of the Caputo fractional derivative is

$$L\left\{{}_0D_t^{\alpha}f(t)\right\}=s^{\alpha}f(s)-\sum_{k=0}^{m-1}s^{\alpha-1-k}\left[\frac{d^k f(t)}{dt^k}\right]_{t=0^+} \tag{4}$$

where $m-1<\alpha<m$

3 Boundary Control of the Anti-stable String Vibration System

In this section, we study the constant parameter $\boldsymbol{a} > 0$, the string vibration system is anti-stable system, so we want to transfer the anti-stable plant to the stable system. For the integer-order model of the anti-stable string vibration system, we want to map the equation (1) into the following target system:

$$\begin{aligned} &v_{tt}(x,t)=v_{xx}(x,t)\\ &v_x(0,t)=b_1v_t(0,t)\\ &v_x(1,t)=f_d(t)=-ku(1,t) \end{aligned} \tag{5}$$

which is exponentially stable for b>0 and k>0 . As will be shown later, the transformation (6) is invertible in a certain norm, so that stability of the target system ensures stability of the closed loop system.

To deal with the boundary control problem of the anti-stable system, we employ the following transformation invented by A.Smyshlyaev [6] (for known q):

$$v(x,t)=u(x,t)+\frac{a_1+b_1}{1+a_1b_1}(-a_1u(0,t)+\int_0^x u_t(y,t)dy) \tag{6}$$

Differentiating (6) with respect to x, setting x=1, and using the boundary condition of the equation (1), we can get the following boundary controller of the anti-stable integer-order vibration system:

$$\begin{aligned} f(t)=&\frac{ka_1(a_1+b_1)}{1+a_1b_1}u(0,t)-ku(1,t)\\ &-\frac{a_1+b_1}{1+a_1b_1}u_t(1,t)-\frac{k(a_1+b_1)}{1+a_1b_1}\int_0^1 u_t(y,t)dy) \end{aligned} \tag{7}$$

This result on stabilization is given by A.Smyshlyaev. The boundary control law (7) has been used in the boundary control of anti-stable wave equation with anti-damping on the uncontrolled boundary, its effectiveness when applied to the boundary control of integer-order wave equation is also proved by the author A.Smyshlyaev.

We will consider the fractional order can be used as a parameter to model the vibration systems, the fractional wave equation is obtained from the standard wave equation by replacing the first-order time derivative with a fractional derivative of order b. It mainly discusses the system in the using fractional calculus tool. We want to transfer the anti-stable fractional-order plant to the stable system. We map the equation (2) into the target system (8):

$$\begin{aligned}&\frac{\partial^{\alpha} v(x,t)}{\partial t^{\alpha}}=\frac{\partial^{2} v(x,t)}{\partial x^{2}}=v_{xx}(x,t) \qquad 1<\alpha<2\\&v_x(0,t)=b_2 v_t(0,t)\\&v_x(1,t)=f_d(t)=-k_d u(1,t)\end{aligned} \tag{8}$$

Consider the follow transformation for the anti-stable fractional order wave equation:

$$v(x,t)=u(x,t)-\int_0^x m(x,y)u_t(y,t)dy-\int_0^x n(x,y)u_x(y,t)dy) \tag{9}$$

where the gains $\boldsymbol{m}(\boldsymbol{x},\boldsymbol{y}),\boldsymbol{n}(\boldsymbol{x},\boldsymbol{y})$ are to be determined.

Substituting (9) into (2) we obtain:

$$\begin{aligned}D_t^{\alpha}(v(x,t))&=\frac{1}{\Gamma(2-\alpha)}\int_0^t \frac{v_{tt}(x,\tau)}{(t-\tau)}d\tau\\&=D_t^{\alpha}(u(x,t))-D_t^{\alpha}(\int_0^x m(x,y)u_t(y,t)dy\\&\quad-\int_0^x n(x,y)u_x(y,t)dy)\\&=D_t^{\alpha}(u(x,t))-D_t^{\alpha}(A(x,t)-B(x,t))\\A(x,t)&=\int_0^x m(x,y)u_t(y,t)dy\\B(x,t)&=\int_0^x n(x,y)u_x(y,t)dy\end{aligned} \tag{10}$$

$$v_{xx}(x,t)=u_{xx}(x,t)-(A(x,t)-B(x,t))_{xx} \tag{11}$$

From (10) (11), we obtain:

$$\begin{aligned}D_t^{\alpha}(v(x,t))&=v_{xx}(x,t))+(A(x,t)-B(x,t))_{xx}\\&\quad-D_t^{\alpha}(A(x,t)-B(x,t))\\&=v_{xx}(x,t))+(A_{xx}(x,t)-D_t^{\alpha}(A(x,t)))\\&\quad+(D_t^{\alpha}(B(x,t))-B_{xx}(x,t))\end{aligned} \tag{12}$$

Matching all the terms, we get two equations as follow:

$$\begin{aligned}&A_{xx}(x,y)=D_t^{\alpha}A(x,y) \quad 1<\alpha<2\\&B_{xx}(x,y)=D_t^{\alpha}B(x,y) \quad 1<\alpha<2\end{aligned} \tag{13}$$

Substituting (4) into the boundary condition of the equation (2), we obtain:

$$\begin{aligned}0&=v_x(0,t)-b_2 v_t(0,t)\\&=(a_2 n(0,0)-m(0,0)-a_2-b)u_t(0,t)\end{aligned} \tag{14}$$

To solve the equations (13) (14),we obtain the transformation(9) can be written in the following form:

$$v(x,t)=u(x,t)-\frac{a_2+b_2}{a_2^2-1}\int_0^x u_t(y,t)dy-\frac{a_2(a_2+b_2)}{a_2^2-1}\int_0^x u_x(y,t)dy) \tag{15}$$

Differentiating with respect to x, we get the following controller:

$$\begin{aligned} f_d(t)=&\frac{k_d a_2(a_2+b_2)}{1+a_2b_2}u(0,t)-k_d u(1,t)\\ &-\frac{a_2+b_2}{1+a_2b_2}u_t(1,t)-\frac{k_d(a_2+b_2)}{1+a_2b_2}\int_0^1 u_t(y,t)dy) \end{aligned} \tag{16}$$

We can see that equation(7) and equation(16) in the same form. Although the boundary control law (7) has been used in the boundary control of anti-stable wave equation with anti-damping on the uncontrolled boundary, its effectiveness when applied to the boundary control of fractional wave equation is still unknown. We will present some simulation results to show the effectiveness of the boundary control law.

First, let us fix the value $k=1$, we compared the displacement response of the different orders $\alpha=1.25, 1.75, 2.00$ of string vibration system application of integer order boundary controller at the free end of string. The simulation results are shown in figure 1. From the figure, we can see that the integer order controller is still able to use for fractional vibration and stability control systems, however, the figure shows the integer-order controller is applied to some of the fractional order system, the settling time is very long, but there is also a more large displacement movement, which is harmful for some actual systems. When $\alpha=2$, the convergence of the controller gain $k_d=1$ makes the shortest time.

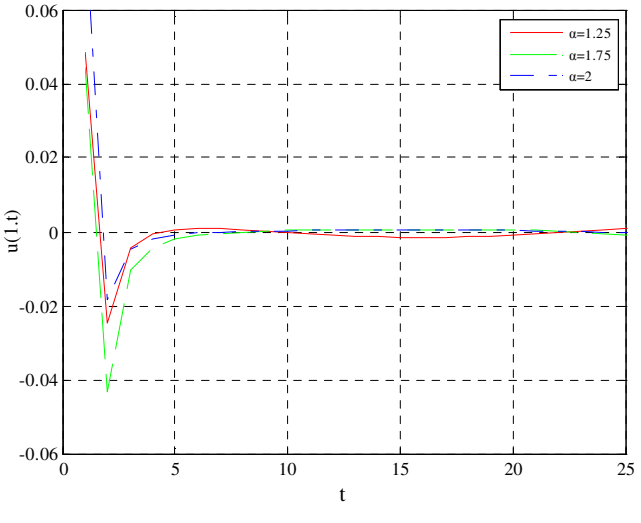


Fig. 1. Tip end movement overtime for different α's

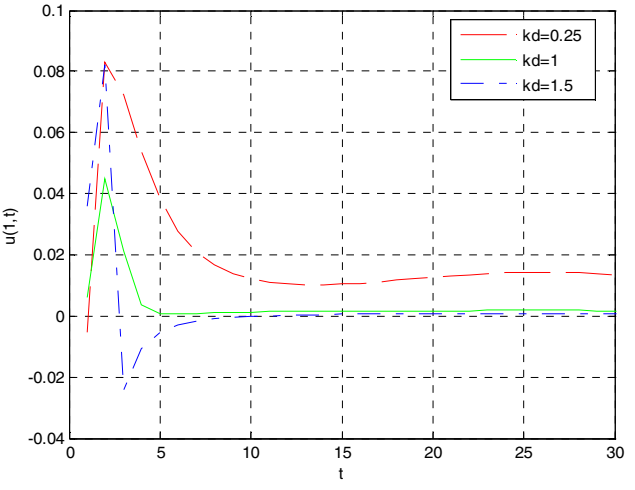


Fig. 2. Tip end movement over time for different gains

Secondly, we study the response of controller gain $k_d = 0.25, 1.00, 1.50$ for the anti-stable fractional-order string vibration system. Simulation results show that shown in figure 2. Seen from the figure 2, k_d increased from 0.25 to 1.50, the system response changed from lack damping state to off damping state. The simulation results show that the integer order controller is still able to control the vibration equation of fractional order.

4 Conclusions

In this paper, two models of the anti-stable string vibration system are be considered. A boundary control method of anti-stable vibration systems is presented in the paper. Numerical simulations are used to compared the displacement response of the different order of string vibration system application of integer order boundary controller at the free end of string. We introduced a new integral transformation for wave equations and used it to obtain the integer-order controllers of a wave equation with negative damping at the boundary. The application of the presented boundary controller to other hyperbolic systems is very promising and will be the subject of future work.

References

1. Chen, G.: Energy decay estimates and exact boundary value controllability for the wave equation in a bounded domain. J. Math. Pure Appl. 58, 249–273 (1979)
2. Komornik, V., Zuazua, E.: A direct method for the boundary stabilization of the wave equation. J. Math. Pure Appl. 69, 33–54 (1990)

3. Krstic, M., Guo, B.-J., Balogh, A., Smyshlyaev, A.: Outputfeedback stabilization of an unstable wave equation. Automatica 44, 63–74 (2008)
4. Liang, J., Chen, Y., Guo, B.-Z.: A hybrid symbolic-numeric simulation method for some typical boundary control problems. In: Proceedings of the IEEE American Control Conference, Boston, USA (2004)
5. Smyshlyaev, A., Krstic, M.: Backstepping observers for a class of parabolic PDEs. Systems and Control Letters 54, 613–625 (2005)
6. Smyshlyaev, A., Krstic, M.: Closed form boundary state feedbacks for a class of 1-D partial integro-differential equations. IEEE Trans. on Automatic Control 49(12), 2185–2202 (2004)
7. Liang, J., Chen, Y.Q., Fullmer, R.: Simulation studies on the boundary stabilization and disturbance rejection for fractional di®usion-wave equation. In: 2004 IEEE American Control Conference (2003)

Stochastic p-Hub Center Problem with Discrete Time Distributions

Kai Yang, Yankui Liu*, and Xin Zhang

College of Mathematics & Computer Science,
Hebei University
Baoding 071002, Hebei, China
yangk09@sina.com, yliu@hbu.edu.cn, xinzhang08@126.com

Abstract. This paper considers a stochastic p-hub center problem, in which travel time is characterized by discrete random vector. The objective of the problem is to minimize the efficient time point of total travel time. For Poisson travel time, the problem is equivalent to a deterministic programming problem by finding the quantiles of the related probability distribution functions. For general discrete distributed travel time, the proposed problem is equivalent to a deterministic mixed-integer linear programming problem. So, we can employ conventional optimization algorithms such as branch-and-bound method to solve the deterministic programming problem. Finally, one numerical example is presented to demonstrate the validity of the proposed model and the effectiveness of the solution method.

Keywords: p-hub center problem, Random travel time, Service level, Mixed-integer programming.

1 Introduction

The p-hub center problem is to locate p hubs in a network and to allocate non-hub nodes to hub nodes so that the maximum travel time between any origin-destination (o-d) pair is minimized. Hubs can serve as consolidation, switching and sorting centers, and allow for the replacement of direct connections between all nodes with fewer, indirect connections [1,2,3]. The p-hub center problem applications can be found in the delivery of perishable or time sensitive systems, such as express mail services and emergency services [4,5,6], in which the maximum travel time represents the best time guarantee that can be offered to all customers. To be competitive, it is important that this value is as low as possible. We remark that every node is allocated to exactly one hub. There is also a multi-assignment version of the problem in which each node is connected to at least one hub node. In the current development we do not consider the case and omit the term "single-assignment" in the rest of the paper.

The p-hub center problem was introduced in [7,8], where [8] formulated the p-hub center problem as a quadratic program. Several linearizations of quadratic

* Corresponding author.

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 182–191, 2011.

programs were proposed by Kara and Tansel [9], who also provided an NP-completeness proof for single allocation case and numerical comparisons for the linearizations. On the basis of the concept radius of hubs, Ernst *et al.* [10] proposed a mixed-integer linear programming for the single and multiple-allocation p-hub center problem.

In the literature, most research has focused on deterministic problems. It is evident that in real applications the travel time can not be considered deterministic since their values may vary because of traffic condition, speed ambulances, time of day, climate conditions, and land and road type. Thus, a more accurate model should take explicitly into account uncertainty by including random travel time rather than deterministic one. As travel time is typically uncertain in reality, Sim *et al.* [11] first attempted to tackle p-hub center problem with stochastic time and service-level constraints involving mutually independent normal distributions; some analytical results and solution heuristics were also discussed. Note that discrete distributions arise frequently in applications, which may also available through experience distribution or approximating continuous probability distribution. So, in the current development, we incorporate random travel time with discrete distributions into p-hub center problem, and suggest alternative methods to model and solve the problem.

The rest of the paper is organized as follows. In Section 2, we present the problem formulation with random travel times. Section 3 analyzes the problem with Poisson distributions and general discrete distributions. The solution method to the proposed problem with the state-of-the-art commercial code LINGO is discussed in Section 4. In Section 5, we present the computational results. Finally, Section 6 summarizes the conclusions in the paper.

2 Problem Formulation

The stochastic p-hub center problem is to locate p hubs in the network and to allocate non-hub nodes to hub nodes so that the maximum travel time between any origin-destination (o-d) pair is minimized for a given service-level β. It is reasonable to assume that the service-level β is close to 1 such as 0.95. For modeling the problem, we adopt the following notation:

- $N = \{1, 2, \ldots, n\}$: the set of nodes in the network;
- T_{ij}: random variable representing the travel time on the link from node i to node j;
- α: discount factor on links between hubs;
- p: the number of hubs to be selected.

For each pair $i, k \in N$, we define the following binary decision variables,

$$X_{ik} = \begin{cases} 1, \text{ if node } i \text{ is assigned to hub } k \\ 0, \text{ otherwise.} \end{cases}$$

When $i = k$, the variable X_{kk} represents the establishment or not establishment of a hub at node k.

We define additional binary decision variables X_{iklj} that represent path in network from node i to node j through hub k first then hub l, i.e.,

$$X_{iklj} = \begin{cases} 1, \text{if exists a path from node } i \text{ to } j \text{ through hub } k \text{ first then } l \\ 0, \text{otherwise.} \end{cases}$$

Objective Function

The objective function includes the following travel time:

The total travel time on a valid path $i \to k \to l \to j$ is

$$(T_{ik} + \alpha T_{kl} + T_{lj}) X_{iklj}, \forall i, j, k, l \in N.$$

Given a service level $\beta \in (0, 1)$, the objective is to minimize the β-efficient time point of total random time in the sense that

$$\min\{\varphi \mid \Pr\{(T_{ik} + \alpha T_{kl} + T_{lj}) X_{iklj} \leq \varphi\} \geq \beta, \forall i, j, k, l \in N\}.$$

Constrains

I: Constraint (1) ensures that path $i \to k \to l \to j$ is a valid path in network if and only if nodes i and j are assigned to hubs k and l, respectively, i.e., $X_{ik}{=}X_{jl}{=}1$,

$$X_{iklj} \geq X_{ik} + X_{jl} - 1, \tag{1}$$

where $X_{iklj} \in \{0, 1\}$.

II: Constraint (2) requires that exactly p hubs are established in the network,

$$\sum_{k \in N} X_{kk} = p. \tag{2}$$

III: Constraint (3) states that a non-hub node i can only be assigned to an open hub at node k,

$$X_{ik} \leq X_{kk}. \tag{3}$$

VI: Constraint (4) imposes the single-assignment rule,

$$\sum_{k \in N} X_{ik} = 1, \tag{4}$$

where $X_{ik} \in \{0, 1\}$.

Using the notation above, we present a new critical value approach to formulating a meaningful p-hub center problem, which is formally stated as follows:

$$\begin{cases} \min & \min\{\varphi \mid \Pr\{(T_{ik} + \alpha T_{kl} + T_{lj}) X_{iklj} \leq \varphi\} \geq \beta, \forall i, j, k, l \in N\} \\ \text{subject to:} & \\ & X_{iklj} \geq X_{ik} + X_{jl} - 1, \forall i, j, k, l \in N \\ & \sum_{k \in N} X_{kk} = p \\ & X_{ik} \leq X_{kk}, \forall i, k \in N \\ & \sum_{k \in N} X_{ik} = 1, \forall i \in N \\ & X_{ik} \in \{0, 1\}, \forall i, k \in N \\ & X_{iklj} \in \{0, 1\}, \forall i, j, k, l \in N. \end{cases} \tag{5}$$

In the current development, we assume that travel times T_{ik}, T_{kl} and T_{lj} are random variables. For simplicity of presentation, we define

$$f(X_{iklj}, \xi_{iklj}) = (T_{ik} + \alpha T_{kl} + T_{lj}) X_{iklj}, \forall i, j, k, l \in N,$$

where $\xi_{iklj} = (T_{ik}, T_{kl}, T_{lj})$ is a random vector with finite support.

To compute the objective in problem (5), it is required to deal with the following critical value function:

$$C : X_{iklj} \rightarrow \min\{\varphi \mid \Pr\{f(X_{iklj}, \xi_{iklj}) \leq \varphi\} \geq \beta, \forall i, j, k, l \in N\},$$

where β is a prescribed probability service level.

If the critical value function can be converted into its deterministic form, then we can obtain equivalent deterministic models. However, in generally case, we cannot do so. It is thus more convenient to deal with the general case by stochastic simulation [12].

In order to compute the critical value function $C(X_{iklj})$, we generate ω^n_{iklj} from a probability space $(\Omega, \mathcal{A}, \Pr)$ and produce random samples $\xi^n_{iklj} = \xi(\omega^n_{iklj})$ for $n = 1, 2, \ldots, N_{iklj}$. Equivalently, we generate random samples ξ^n_{iklj} for $n = 1, 2, \ldots, N_{iklj}$ according to the probability distribution of ξ_{iklj}. Now we define

$$h(X_{iklj}, \xi_{iklj}) = \begin{cases} 1, \text{ if } f(X_{iklj}, \xi_{iklj}) \leq \varphi \\ 0, \text{ otherwise} \end{cases}$$

for $n = 1, 2, \ldots, N_{iklj}$, which are random variables such that $E[h(X_{iklj}, \xi_{iklj})] = \beta$ for all n. By the strong law of large numbers, we obtain

$$\frac{1}{N_{iklj}} \sum_{n=1}^{N} h(X_{iklj}, \xi_{iklj}) \rightarrow \beta$$

in the sense of almost sure as N towards infinity. Note that $\sum_{n=1}^{N_{iklj}} h(X_{iklj}, \xi_{iklj})$ is the number ξ^n_{iklj} satisfying $f(X_{iklj}, \xi^n_{iklj}) \leq \varphi$ for $n = 1, 2, \ldots, N_{iklj}$. Thus φ is the N'_{iklj}th smallest element in the sequence $\{f(X_{iklj}, \xi^k_{iklj}), k = 1, \ldots, N_{iklj}\}$, where N'_{iklj} is the integer part of βN_{iklj}.

Now let us consider problem (5) from a different point of view. By introducing an additional variable φ, we have the following equivalent formulation:

$$\begin{cases} \min & \varphi \\ \text{subject to:} & \\ & \Pr\{(T_{ik}+\alpha T_{kl}+T_{lj}) X_{iklj} \leq \varphi\} \geq \beta, \forall i, j, k, l \in N \\ & X_{iklj} \geq X_{ik} + X_{jl} - 1, \forall i, j, k, l \in N \\ & \sum_{k \in N} X_{kk} = p \\ & X_{ik} \leq X_{kk}, \forall i, k \in N \\ & \sum_{k \in N} X_{ik} = 1, \forall i \in N \\ & X_{ik} \in \{0, 1\}, \forall i, k \in N \\ & X_{iklj} \in \{0, 1\}, \forall i, j, k, l \in N. \end{cases} \tag{6}$$

The equivalence with (5) is immediate by noting that for each fixed feasible solution to (6), it is sufficient to take into account the minimal φ in the constraint, this minimal φ is just $\min\{\varphi \mid \Pr\{f(X_{iklj}, \xi_{iklj}) \leq \varphi\} \geq \beta, \forall i, j, k, l \in N\}$.

Problem (6) clearly belongs to the class of probabilistic constraint programming problems [13]. The traditional solution methods require conversion of probabilistic constraints to their respective deterministic equivalents. As we know, this conversion is usually hard to perform and only successfully for special case. We will discuss the equivalent formulation of problem (6) in the the case when random travel time are characterized by discrete distributions.

3 Equivalent Mixed-Integer Programming

First, we consider the case when travel times T_{ik}, T_{kl} and T_{lj} are mutually independent Poisson random variables with parameters λ_{ik}, λ_{kl} and λ_{lj}, respectively. It is known that the sum of a finite number of independent Poisson random variables is also a Poisson variable. Hence, the total travel time on a valid path $i \rightarrow k \rightarrow l \rightarrow j$ in problem (6) can be modeled by a Poisson random variable with mean $(\lambda_{ik} + \alpha\lambda_{kl} + \lambda_{lj})X_{iklj}$.

Now we consider the following service level constraint with $\beta \in (0, 1)$,

$$\Pr\{f(X_{iklj}, \xi_{iklj}) \leq \varphi\} \geq \beta, \forall i, j, k, l \in N. \tag{7}$$

According to Poisson probability distribution, the service level constraint (7) can then be rewritten as

$$\sum_{k \leq \varphi} \Pr\{\xi_{iklj} = k\} \geq \beta, \forall i, j, k, l \in N,$$

which is equivalent to

$$\sum_{k=0}^{\varphi} e^{-(\lambda_{ik}+\alpha\lambda_{kl}+\lambda_{lj})X_{iklj}} \frac{((\lambda_{ik} + \alpha\lambda_{kl} + \lambda_{lj})X_{iklj})^k}{k!} \geq \beta, \forall i, j, k, l \in N.$$

As a consequence, we can express service level constraint (7) as

$$Q^{-}_{\xi_{iklj}}(\beta)X_{iklj} \leq \varphi, \forall i, j, k, l \in N, \tag{8}$$

where $Q^{-}_{\xi_{iklj}}(\beta)$ denotes the left end-point of the closed interval of β-quantiles of the probability distribution function of ξ_{iklj}.

Therefore, problem (6) with Poisson travel time is transformed into the deterministic programming problem with constraint (8) replacing (7), which can be solved by radial heuristic algorithm [11].

We next consider the case when travel times T_{ik}, T_{kl} and T_{lj} are general discrete random variables. For the sake of simplicity of presentation, we denote $\xi_{iklj} = (T_{ik}, T_{kl}, T_{lj})$, which is a discrete random vector with the following probability distribution

$$\begin{pmatrix} (\hat{T}_{ik}^{1}, \hat{T}_{kl}^{1}, \hat{T}_{lj}^{1}) \cdots (\hat{T}_{ik}^{N_{iklj}}, \hat{T}_{kl}^{N_{iklj}}, \hat{T}_{lj}^{N_{iklj}}) \\ p_{iklj}^{1} \quad \cdots \quad p_{iklj}^{N_{iklj}} \end{pmatrix},$$

where $p_{iklj}^{n} > 0$, $n = 1, 2, \ldots, N_{iklj}$, and $\sum_{n=1}^{N_{iklj}} p_{iklj}^{n} = 1$, $\forall i, j, k, l \in N$.

In this case, consider the following service level constraint with $\beta \in (0, 1)$,

$$\Pr\{f(X_{iklj}, \xi_{iklj}) \leq \varphi\} \geq \beta, \forall i, j, k, l \in N.$$

By introducing a "big enough" constant M, one has

$$(\hat{T}_{ik}^{n} + \alpha \hat{T}_{kl}^{n} + \hat{T}_{lj}^{n}) X_{iklj} - M \leq \varphi, \forall i, j, k, l \in N, n = 1, 2, \ldots, N_{iklj}.$$

In addition, we introduce a vector z_{iklj} of binary variables whose components z_{iklj}^{n}, $n = 1, 2, \ldots, N_{iklj}$ take value 0 if the corresponding constraint has to be satisfied and 1 otherwise.

As a consequence, stochastic p-hub center problem (6) can be turned into the following equivalent mixed-integer programming model

$$\begin{cases} \min & \varphi \\ \text{subject to:} & \\ & (\hat{T}_{ik}^{n} + \alpha \hat{T}_{kl}^{n} + \hat{T}_{lj}^{n}) X_{iklj} - M \cdot z_{iklj}^{n} \leq \varphi, \forall i, j, k, l \in N, \\ & \qquad\qquad n = 1, 2, \ldots, N_{iklj} \\ & \sum_{n=1}^{N_{iklj}} p_{iklj}^{n} z_{iklj}^{n} \leq (1 - \beta), \forall i, j, k, l \in N \\ & X_{iklj} \geq X_{ik} + X_{jl} - 1, \forall i, j, k, l \in N \\ & \sum_{k \in N} X_{kk} = p \\ & X_{ik} \leq X_{kk}, \forall i, k \in N \\ & \sum_{k \in N} X_{ik} = 1, \forall i \in N \\ & X_{ik} \in \{0, 1\}, \forall i, k \in N \\ & X_{iklj} \in \{0, 1\}, \forall i, j, k, l \in N \\ & z_{iklj}^{n} \in \{0, 1\}, \forall i, j, k, l \in N, n = 1, 2, \ldots, N_{iklj}, \end{cases} \tag{9}$$

where $\sum_{n=1}^{N_{iklj}} p_{iklj}^{n} z_{iklj}^{n} \leq (1 - \beta), \forall i, j, k, l \in N$, define a binary knapsack constraint ensuring that violation of stochastic service level constraints is limited to $(1 - \beta)$.

4 Solution Method

Since problem (9) is a mixed-integer linear programming problem with binary variables, one possibility for solving it is to use branch-and-bound method [14], which is a solution procedure that systematically examines all possible combinations of discrete variables. The solution process is described as follows.

Consider the relaxation problem of mixed-integer programming problem (9)

$$\begin{cases} \min & \varphi \\ \text{subject to:} & \\ & (\hat{T}_{ik}^n + \alpha\hat{T}_{kl}^n + \hat{T}_{lj}^n)X_{iklj} - M \cdot z_{iklj}^n \le \varphi, \forall i,j,k,l \in N, \\ & \qquad\qquad\qquad\qquad\qquad\qquad n = 1,2,\ldots,N_{iklj} \\ & \sum_{n=1}^{N_{iklj}} p_{iklj}^n z_{iklj}^n \le (1-\beta), \forall i,j,k,l \in N \\ & X_{iklj} \ge X_{ik} + X_{jl} - 1, \forall i,j,k,l \in N \\ & \sum_{k\in N} X_{kk} = p \\ & X_{ik} \le X_{kk}, \forall i,k \in N \\ & \sum_{k\in N} X_{ik} = 1, \forall i \in N \\ & 0 \le X_{ik} \le 1, \forall i,k \in N \\ & 0 \le X_{iklj} \le 1, \forall i,j,k,l \in N \\ & 0 \le z_{iklj}^n \le 1, \forall i,j,k,l \in N, n = 1,2,\ldots,N_{iklj}. \end{cases} \tag{10}$$

Let P denote the set of problems derived from the original mixed-integer programming. Initially, P will include only the continuous relaxation. As we proceed toward the solution, P will include problems with added constraints as the integer restrictions are imposed. Let p_0 denote the relaxation problem (10).

Then, the process of branch and bound method includes the following several steps (see [14]):

Initialize: Set $U = +\infty$ and $P = p_0$. Solve p_0. If the solution is 0 or 1, set U equal to the optimal value and terminate; or if there is no feasible solution, terminate; else select problem.

Select problem: Remove from P problem p having a solution that fails to satisfy some 0-1 constraint and has an objective function value greater than or equal to U and choose variable; or if there is no such problem in P, terminate.

Choose variable: Choose an 0-1 constrained variable x_i having not value 0 or 1 in the solution to problem p, and branch on x_i.

Branch on x_i: Add to P the problem p' and p'' formed by adding to p the constraints x_i=0 and x_i=1, respectively. If an solution to p' or p'' is obtained with objective function value less than U, set U equal to the new objective function value, and select problem.

Terminate: If $U = +\infty$, then there is no feasible solution; otherwise, the solution corresponding to the current value of U is optimal.

The code LINGO is a state-of-the-art commercial general branch-and-bound IP-code, which works in conjunction with the linear, nonlinear, and quadratic solvers [15]. The structure of the constraints in the problem makes the use of modeling language particularly appropriate. This yields a rather efficient solution method for this kind of problem. In the next section, we will consider a specific application. There we will rely on LINGO to solve the problem.

5 Numerical Experiments

In this section, we present an application example about stochastic p-hub center problem. We only consider the case when travel times are discrete random variables. Assume that there are 6 cities in a region whose travel time T_{ij} from city i to city j and location are given in Table 1 and Figure 1, respectively, where travel times are symmetric and mutually independent random variables. We also assume that travel time T_{ij} only has three realizations: optimistic arrival time, mean arrival time and pessimistic arrival time. Obviously, the probability of mean arrival time is much larger than the other two values.

In the tests, we consider $p = 2, 3$ and service-level parameters $\beta = 0.80, 0.90$, and 0.95, respectively. All computational tests have been carried out on a personal computer. Given the service level parameter β, we employ LINGO 8.0 software to solve the equivalent mathematical programming model. In order to illustrate parameter's influence to efficiency, we also compare solutions with different values of parameter β, and the computational results are reported in Table 2.

In Figure 2, we depict a valid path 1→4→5→3, from which we find that the optimal locations of p-hubs tend to form a certain structure, where one hub is located in the center of the region (Figure 2). In addition, when the service-level parameter β in the network increases, the maximum travel time becomes longer.

Table 1. Travel Time T_{ij} from City i to j

T_{ij}	1	2	3	4	5	6
1	0	$\begin{pmatrix} 13 & 15 & 16 \\ 0.1 & 0.6 & 0.3 \end{pmatrix}$	$\begin{pmatrix} 18 & 21 & 23 \\ 0.1 & 0.7 & 0.2 \end{pmatrix}$	$\begin{pmatrix} 8 & 10 & 12 \\ 0.1 & 0.8 & 0.1 \end{pmatrix}$	$\begin{pmatrix} 16 & 17 & 18 \\ 0.2 & 0.6 & 0.2 \end{pmatrix}$	$\begin{pmatrix} 14 & 16 & 17 \\ 0.1 & 0.8 & 0.1 \end{pmatrix}$
2		0	$\begin{pmatrix} 8 & 9 & 10 \\ 0.3 & 0.6 & 0.1 \end{pmatrix}$	$\begin{pmatrix} 11 & 12 & 13 \\ 0.1 & 0.8 & 0.1 \end{pmatrix}$	$\begin{pmatrix} 10 & 11 & 12 \\ 0.2 & 0.7 & 0.1 \end{pmatrix}$	$\begin{pmatrix} 15 & 17 & 19 \\ 0.1 & 0.8 & 0.1 \end{pmatrix}$
3			0	$\begin{pmatrix} 14 & 15 & 17 \\ 0.2 & 0.7 & 0.1 \end{pmatrix}$	$\begin{pmatrix} 7 & 9 & 10 \\ 0.1 & 0.8 & 0.1 \end{pmatrix}$	$\begin{pmatrix} 13 & 15 & 16 \\ 0.1 & 0.8 & 0.1 \end{pmatrix}$
4				0	$\begin{pmatrix} 6 & 8 & 10 \\ 0.1 & 0.8 & 0.1 \end{pmatrix}$	$\begin{pmatrix} 6 & 7 & 8 \\ 0.2 & 0.7 & 0.1 \end{pmatrix}$
5					0	$\begin{pmatrix} 5 & 6 & 7 \\ 0.1 & 0.8 & 0.1 \end{pmatrix}$
6						0

Table 2. Computational Results for Different Parameters

p	β	Optimal objective value	Time CPU (sec)
2	0.80	23.600000	301
2	0.90	23.600000	291
2	0.95	23.900000	239
3	0.80	20.500000	211
3	0.90	21.600000	359
3	0.95	21.900000	212

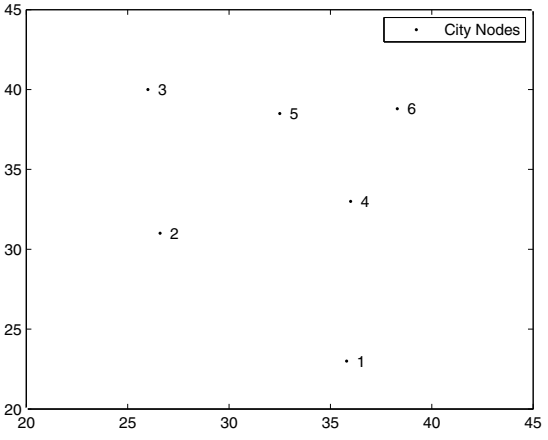


Fig. 1. The Locations of Six Cities

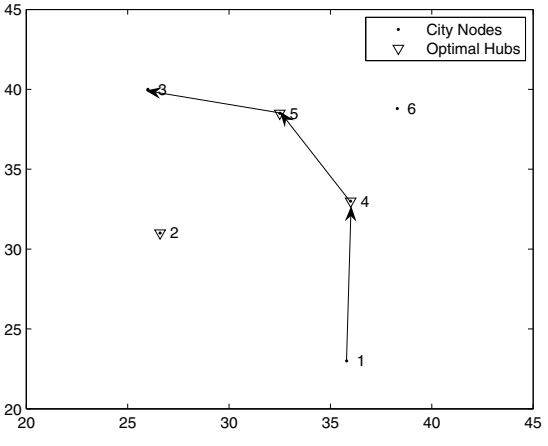


Fig. 2. Plot of the Optimal Hub Locations with p=3, and β=0.80

6 Conclusions

This work studied the p-hub center problem with discrete random travel time, which seeks to configure a network to minimize efficient time point. For Poisson travel time, we showed the stochastic problem is equivalent to a deterministic programming problem. For general discrete random travel time, the problem could be formulated as an equivalent mixed-integer programming model by introducing auxiliary binary variables. We employed LINGO to solve this equivalent programming problem.

From computational point of view, it may be infeasible to employ LINGO solver for quite large instances of the p-hub center problem. Therefore, other solution heuristics such as a combination of approximation method and neural network should be further developed, which will be addressed in our future research.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No.60974134), and the Natural Science Foundation of Hebei Province (No. A2011201007).

References

1. Hakimi, S.L.: Optimum Distribution of Switching Centres in a Communication Network and Some Related Graph Theoretic Problems. Oper. Res. 13, 462–475 (1965)
2. Klincewicz, J.G.: Hub Location in Backbone/Tributary Network Design: the State of the Art. Eur. J. Oper. Res. 4, 139–154 (2008)
3. Alumur, S., Kara, B.Y.: Network Hub Location Problem: a Review. Location Sci. 6, 307–335 (1998)
4. Bryan, D.L., O'Kelly, M.E.: Hub-and-Spoke Networks in Air Transportation: an Analytical Review. J. Regional Sci. 39, 275–295 (1999)
5. Chung, S.H., Myung, Y.S., Tcha, D.W.: Optimal Design of a Distributed Network with a Two-Level Hierarchical Structure. Eur. J. Oper. Res. 62, 105–115 (1992)
6. Ernst, A.T., Krishnamoorthy, M.: Efficient Algorithm for the Uncapacited Single Allocation p-Hub Median Problem. Location Sci. 4, 139–154 (1996)
7. O'Kelly, M.E., Miller, H.J.: Solution Strategies for the Single Facility Minimax Hub Location Problem. The J. RSAI 70, 367–380 (1991)
8. Campbell, J.F.: Integer Programming Formulations of Discrete Hub Location Problems. Eur. J. Oper. Res. 72, 387–405 (1987)
9. Kara, B.Y., Tansel, B.: On the Single Assignment p-Hub Center Problem. Eur. J. Oper. Res. 125, 648–655 (2000)
10. Ernst, A.T., Hamacher, H.W., Jiang, H., Krishnamoorthy, M., Woeginger, G.: Uncapacitated Single and Multiple Allocation p-Hub Center Problems. Comput. Oper. Res. 36, 2230–2241 (2009)
11. Sim, T., Lowe, T.J., Thomas, B.W.: The Stochastic p-Hub Center Problem with Service-Level Constraints. Comput. Oper. Res. 36, 3166–3177 (2009)
12. Rubinstein, R.Y., Melamed, B.: Modern Simulation and Modeling. John Wiley & Sons, Chichester (1998)
13. Kall, P., Mayer, J.: Stochastic Linear Programming: Models, Theory and Computation. Kluwer Academic Publishers, Dordrecht (2005)
14. Walker, R.C.: Introduction to Mathematical Programming. Pearson Eduction Inc., London (1999)
15. Atamtürk, A., Savelsbergh, M.W.P.: Integer-Programming Software Systems. Ann. Oper. Res. 140, 67–124 (2005)

Orthogonal Feature Learning for Time Series Clustering

Xiaozhe Wang[1] and Leo Lopes[2]

[1] School of Management, La Trobe University, Melbourne, VIC, 3086, Australia
[2] School of Mathematical Sciences, Monash University, Clayton, VIC 3800, Australia

Abstract. This paper presents a new method that uses orthogonalized features for time series clustering and classification. To cluster or classify time series data, either original data or features extracted from the data are used as input for various clustering or classification algorithms. Our methods use features extraction to represent a time series by a fixed-dimensional vector whose components are statistical metrics. Each metric is a specific feature based on the global structure of the time series data given. However, if there are correlations between feature metrics, it could result in clustering in a distorted space. To address this, we propose to orthogonalize the space of metrics using linear correlation information to reduce the impact on the clustering from the correlations between clustering inputs. We demonstrate the orthogonal feature learning on two popular clustering algorithms, k-means and hierarchical clustering. Two benchmarking data sets are used in the experiments. The empirical results shows that our proposed orthogonal feature learning method gives a better clustering accuracy compared to all other approaches including: exhaustive feature search, without feature optimization or selection, and without feature extraction for clustering. We expect our method to enhance the feature extraction process which also serves as an improved dimension reduction resolution for time series clustering and classification.

Keywords: time series, clustering and classification, feature learning, orthogonalization.

1 Introduction

In this paper, we present an algorithm of orthogonal learning on features extracted for time series clustering or classification. Clustering time series and other sequences of data has become an important topic, motivated by several research challenges including similarity search of medical and astronomical sequences, as well as the challenge of developing methods to recognize dynamic changes in time series [1]. Features extraction from the original series has been widely used and serves as a dimensionality reduction techniques to improve the clustering efficiency.

Among various statistical-based dimension reduction techniques, principal component analysis has been most commonly used as the dimensionality-reduction tool for the feature space [2]. In general, the number of principal components should be known as a predetermined parameter, which may be difficult to select in practice. Hidden markov models have also been used in time series clustering and classification [3]. Based on assumed probability distributions, they are capable to capture both the dependencies

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 192–198, 2011.

between variables and the serial correlations in the measurements. However, their performance in clustering long time series data was not promising compared to many other methods [4]. In 2006, wang et al. [5] proposed a characteristic-based clustering method that transform the original time series into a fixed-dimensional vector. Our previous work has demonstrated the advantages of handling time series data of various lengths, with missing values or long (or high dimensional) time series. However, if there are correlations between feature metrics, it could result in clustering in a distorted space. To address this, we propose to orthogonalize the space of metrics using linear correlation information to reduce the impact on the clustering from the correlations between clustering inputs. We demonstrate the orthogonal feature learning on two popular clustering algorithms, k-means and hierarchical clustering.

We assume the statistical features used to represent the time series' global structure form a 'super set', i.e., a set consisting all best possible features that could be considered for extraction from the original time series. The focus of this paper is on how to optimize the features to improve the clustering or classification performance rather than to seek potential features candidates. More specifically, we aim to produce an optimized feature set that can effective in feature extraction (or dimension reduction). In addition, it should be both algorithm independent and domain data independent.

We present our orthogonal features learning method for time series data in Section 2. Then in Section 3 we explain the features identified for the feature extraction. In Section 4 we introduce the clustering algorithms and data sets used for the evaluation and explain how our orthogonal feature learning can be applied to those algorithms. Finally, we show the results from the evaluation with promising clustering accuracy, and discuss some future plans in in Section 5.

2 Orthogonal Features Learning

Let Y_i denotes a univariate time series, which is often written as $Y_i = y_1, \ldots, y_t$ for a series with t observations. In a data set, regardless of the variation in the length t, a finite vector consists of n metrics to be extracted based on the global structure of the time series Y_i. Each $Y_i = y_1, \ldots, y_t$ is transformed into a fixed-dimensional feature vector V, $(v_1, \ldots, v_n)$. In this paper, we demonstrate our method with 13 feature vectors, (i.e. the value for n is 13). Each feature vector is based on a statistical metric extracted from the original time series. As discussed in following Section 3, we selected a finite number of statistical metrics to describe the time series quantitatively, and these metrics should be informative in measuring time series global structure, such as trend, seasonality, and many others. (i.e., 13 statistical metrics were computed from Y to form the feature vector). Then, the time series data set D of m time series objects $\mathbf{Y} = Y_1, \ldots, Y_m$ for a clustering or classification task is transformed into a matrix $\mathbf{D} := (v_{i,j})_{m*n}$.

There are a few reasonable approaches based on the idea of a super set of features:

- Use the matrix **D** directly as the input for a clustering algorithm.
- If the data set comes from a particular domain with certain known background knowledge, a learning procedure such as a feed-forward algorithm can be used to select an optimized subset of features from the 'super set', and then construct a

new set $V^{'}$. Let $(v_1, \ldots, v_s)$ denote the new vector, where $s \leq n$. Then the matrix becomes $\mathbf{D}^{'} := (v_{i,j})_{m*s}$;

– The most common problem in the features is their correlation. We can remove the linear correlation between the features and transform each $\mathbf{d}_i$ into a vector in an orthogonalized space, in which the most independent features are weighted (or selected) more heavily, V^*, $(v_1^*, \ldots, v_n^*)$, then the input matrix is: $\mathbf{D}^* := (v_{i,j}^*)_{m*n}$.

To orthogonalize the space, we use the fact that correlation between two vectors can be interpreted as the cosine between the vectors. Let $\mathbf{Z}$ be a mean-shifted version of $\mathbf{D}$ where each column has mean zero. Then $\mathbf{D}^* := \mathbf{Z}(\mathbf{Z}^t\mathbf{Z})^{\frac{1}{2}}$ has mean zero, and since its columns are orthonormal, its correlation matrix is I. Note that no training data is needed to perform the transform.

3 Global Structure Based Statistical Features Extraction on Time Series Data

Time series can be described using a variety of adjectives such as seasonal, trending, noisy, non-linear, chaotic,etc. The extracted statistical features should carry summarized information of time series data, capturing the *global picture* based on the structure of the entire time series. A novel set of characteristic metrics were proposed by Wang et al. [5] to represent univariate time series and their structure-based features. This set of metrics not only includes conventional features (for example, trend), but also covers many advanced features (for example, chaos) which are derived from research on new phenomena in physics [6]. In their work, 13 metrics were extracted based on the following structure-based statistical features to form a rich portrait of the nature of a time series: *Trend, Seasonality, Serial Correlation, Non-linearity, Skewness, Kurtosis, Self-similarity, Chaotic, and Periodicity*.

Trend and Seasonality. Trend and seasonality are common features of time series, and it is natural to characterize a time series by its degree of trend and seasonality. In addition, once the trend and seasonality of a time series has been measured, we can de-trend and de-seasonalize the time series to enable additional features such as noise or chaos to be more easily detectable. Y_t is the original data, $X_t = Y_t^* - T_t$ denotes de-trended data after a Box-Cox transformation, and $Z_t = Y_t^* - S_t$ denotes the de-seasonalized data after a Box-Cox transformation. $Y_t^{'} = Y_t^* - T_t - S_t$ is a time series after trend and seasonality adjustment. Then, $1 - \mathrm{Var}(Y_t^{'})/\mathrm{Var}(Z_t)$ is the measure of trend and $1 - \mathrm{Var}(Y_t^{'})/\mathrm{Var}(X_t)$ is the measure of seasonality.

Periodicity and Serial Correlation. The periodicity is very important for determining the seasonality and examining the cyclic pattern of the time series.

Non-linear Autoregressive Structure. Non-linear time series models have been used extensively in recent years to model dynamics not adequately represented by linear models. For example, the well-known *sunspot* data set and *lynx* data set have non-linear structure.

Skewness and Kurtosis. Skewness is a measure of symmetry, or more precisely, the lack of symmetry in a distribution or a data set. Kurtosis is a measure of whether the data are peaked or flat relative to a normal distribution. A data set with high kurtosis tends to have a distinct peak near the mean, declines rather rapidly, and has heavy tails.

Self-similarity. Processes with long-range dependence have attracted a good deal of attention from theoretical physicists and in 1984, Cox [7] first presented a review of second-order statistical time series analysis. The subject of self-similarity (or *long-range dependence*) and the estimation of statistical parameters of time series in the presence of long-range dependence are becoming more common in several fields of science.

Chaos. Many systems in nature that were previously considered as random processes are now categorized as chaotic systems. Nonlinear dynamic systems often exhibit chaos, which is characterized by sensitive dependence on initial values, or more precisely by a positive lyapunov exponent. Recognizing and quantifying chaos in time series are important steps toward understanding the nature of random behavior, and reveal the dynamic feature of time series [6].

4 Clustering Algorithms and Data Sets for Evaluation

We choose two of the most popular clustering algorithms (k-means [8] and hierarchical clustering) in the evaluation and use two benchmarking data sets for the convenience of comparison.

k-means Clustering algorithms
k-means clustering has been recognized as a fast method compared to other clustering algorithms [9]. It includes the steps: 1) Decide the value of k and initialize the k cluster centers randomly; 2) Decide the class memberships of the N objects by assigning them to the nearest cluster center; 3) Re-estimate the k cluster centers, by assuming the memberships found are correct; 4) When none of the N objects changed their membership in the last iteration, exit. Otherwise go to step 2. The objective to achieve is to minimize total intra-cluster variance, or, the squared error function: $V = \sum_{i=1}^{k} \sum_{x_j \in S_i} (x_j - \mu_i)^2$, where there are k clusters $S_i, i = 1, 2, \ldots, k$, and μ_i is the centroid or mean point of all the points $x_j \in S_i$. In our experiments, we used two k-means methods: one by Hartigan and Wong [10], the other the most commonly-used method given by MacQueen [11].

Hierarchical Clustering algorithms
Hierarchical clustering provides a cascading series of partitions of the objects to be grouped. It subdivides into two types of methods: agglomerative and divisive, of which the agglomerative method is most commonly used. Single-linkage, complete-linkage and average-linkage clusterings [12] are three different algorithms are known in the agglomerative method family defined by three different techniques used as distance measures in clustering process. We used all three methods in the experiments and Ward distance [13] which is a weighted group average criterion that keeps the within-cluster variance as small as possible at each of the agglomerative steps.

Data Sets

For a fair comparison, we used two benchmark datasets from the UCR Time Series Data Mining Archive in the experiments [14], "reality check" and "18 pairs" have been tested for clustering by other researchers and used as benchmarking for comparison. "Reality check" consists of data from space shuttle telemetry, exchange rates, and artificial sequences. The data is normalized so that the minimum value is 0 and the maximum is 1. There are fourteen time series and each contains 1,000 data points. In the "18 pairs" dataset, thirty-six time series with 1,000 data points each come in eighteen pairs. The original data from these two data sets are shown in Figure 1.

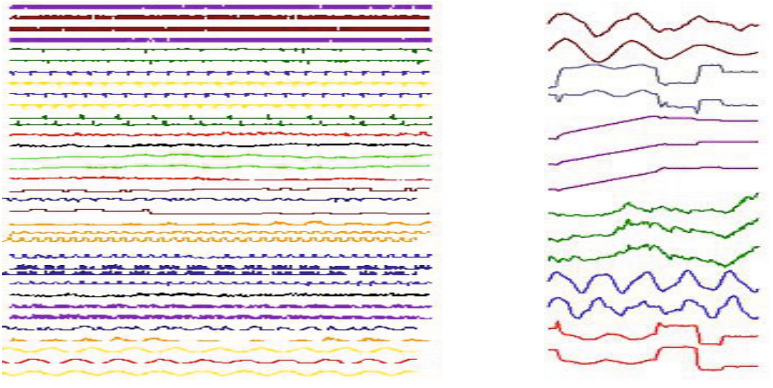


Fig. 1. Data set '18-pairs' (left) and Data set 'reality check' (right)

5 Our Results and Discussion

The accuracy of each clustering algorithm can be measured using a cluster purity, P, which is a percentage count: $P = (C_d/C_e) * 100\%$, where C_d is the number of the *dominant* class objects within that cluster, and C_e is the *expected* number of objects in that cluster. We implemented the proposed 'orthogonal feature learning' method on both k-means clustering and hierarchical clustering algorithms in R [15]. Because the two data sets used in our experiments were previously studied with known class labels as ground-truth, the accuracy of clustering results can be calculated using P for evaluation directly. In both experiments using hierarchical clustering and k-means clustering, two types of matrices are used as clustering inputs: one is the original matrix $\mathbf{D}$ that is derived from the feature vectors without orthogonal learning, and the other one is the orthogonalized matrix $\mathbf{D}^*$.

As shown in both Table 1 and Table 2, orthogonal feature learning can provide much better accuracy compared to using the original feature matrix. Given the five algorithms and two data sets used in our experiments, there is only one experiment with Single-linkage on reality check data set, whose clustering accuracy decreased after the orthogonal treatment. All other results have shown some strong improvement in clustering accuracy through the proposed orthogonal learning. In conclusion, the experiments have

shown that the proposed orthogonal learning on time series statistical features can certainly improve the clustering accuracy by reducing the correlations between different features. The promising results from the empirically study also provides evidence that the orthogonal feature learning can be a flexible and efficient technique for dimensional reduction for time series clustering and classification. Mathematically, any algorithm must perform better on features that are orthogonal, since they provide more information about the data than correlated features do.

The practical impact of orthogonality may vary depending on which measures the algotrithm takes internally to address the issue (i.e. by using alternative linear algebra) and on how strongly correlated the features are. Thus, to establish that orthogonalization will improve results it is not necessary to run computatational experiments at all. The theoretical exposition is sufficient. To establish the practical impact, it would be ideal to have more diverse data sets with different correlation structures and we intend to explore that further in future research. Such a study must carefully consider the fact that correlation between n features is an $(n-1)$-dimensional quantity and thus is beyond the scope of this paper. To evaluate our method for algorithm and data dependency or independence, we plan to do more empirical study with more algorithms and data sets in the future.

Table 1. Evaluation results with hierarchical clustering algorithms (P %), column **D** is the clustering experiment using original matrix as input after feature extraction, and column $\mathbf{D}^*$ is the clustering experiment using orthogonalized features as inputs

Data	**Complete-linkage**		**Average-linkage**		**Single-linkage**		***Mean***	
Set	**D**	$\mathbf{D}^*$	**D**	$\mathbf{D}^*$	**D**	$\mathbf{D}^*$	**D**	$\mathbf{D}^*$
reality check	76.8	*76.8*	76.8	*76.8*	76.8	*63.4*	**76.8**	***72.3***
18 pairs	44.4	*55.6*	44.4	*55.6*	44.4	*61.1*	**44.4**	***57.4***
both	51.5	*60.2*	51.5	*60.2*	51.5	*61.6*	**51.5**	***60.7***
Mean	**57.6**	***64.2***	**57.6**	***64.2***	**57.6**	***62.0***	**57.6**	***63.5***

Table 2. Evaluation results with k-means clustering algorithms (P %), column **D** is the clustering experiment using the original matrix as input after feature extraction, and column $\mathbf{D}^*$ is the clustering experiment using orthogonalized features as inputs

Data	**Hartigan-Wong**		**MacQueen**		***Mean***	
Set	**D**	$\mathbf{D}^*$	**D**	$\mathbf{D}^*$	**D**	$\mathbf{D}^*$
reality check	76.8	*76.8*	36.8	*83.4*	**56.8**	***80.1***
18 pairs	44.4	*55.6*	33.3	*55.6*	**38.9**	***55.6***
both	51.5	*60.2*	34.1	*61.6*	**42.8**	***60.9***
Mean	**57.6**	***64.2***	**34.7**	***66.9***	**46.2**	***65.5***

References

[1] Scargle, J.: Timing: New Methods for Astronomical Time Series Analysis. American Astronomical Society, 197th AAS Meeting,# 22.02; Bulletin of the American Astronomical Society 32, 1438 (2000)
[2] Trouve, A., Yu, Y.: Unsupervised clustering trees by non-linear principal component analysis. Pattern Recognition and Image Analysis 2, 108–112 (2001)
[3] Owsley, L., Atlas, L., Bernard, G.: Automatic clustering of vector time-series for manufacturing machine monitoring. In: Proc. of ICASSP IEEE Int. Conf. Acoust. Speech Signal Process., vol. 4, pp. 3393–3396 (1997)
[4] Keogh, E., Kasetty, S.: On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration. DMKD 7(4), 349–371 (2003)
[5] Wang, X., Smith, K., Hyndman, R.: Characteristic-Based Clustering for Time Series Data. Data Mining and Knowledge Discovery 13(3), 335–364 (2006)
[6] Lu, Z.: Estimating Lyapunov Exponents in Chaotic Time Series with Locally Weighted Regression. PhD thesis, University of North Carolina at Chapel Hill (1994)
[7] Cox, D.: Long-range dependence: a review. Statistics: an appraisal. In: David, H.A., David, H.T. (eds.) 50th Anniversary Conf., Iowa State Statistical Laboratory, pp. 55–74. The Iowa State University Press (1984)
[8] Dasu, T., Swayne, D.F., Poole, D.: Grouping Multivariate Time Series: A Case Study. KDD-2006 workshop report: Theory and Practice of Temporal Data Mining, ACM SIGKDD Explorations Newsletter 8(2), 96–97 (2006)
[9] Bradley, P., Fayyad, U.: Refining Initial Points for K-Means Clustering. In: Proc. 15th International Conf. on Machine Learning, vol. 727 (1998)
[10] Hartigan, J., Wong, M.: A K-means clustering algorithm. JR Stat. Soc., Ser. C 28, 100–108 (1979)
[11] MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297 (1967)
[12] Lance, G., Williams, W.: A general theory of classificatory sorting strategies. I. Hierarchical systems. Computer Journal 9(4), 373–380 (1967)
[13] Ward, J.: Hierarchical grouping to optimize an objective function. Journal of the American Statistical Association 58(301), 236–244 (1963)
[14] Keogh, E., Folias, T.: The ucr Time Series Data Mining Archive (2002), http://www.cs.ucr.edu/~eamonn/TSDMA/index.html
[15] Ihaka, R., Gentleman, R.: R: A Language for Data Analysis and Graphics. Journal of Computational and Graphical Statistics 5(3), 299–314 (1996)

A Text Document Clustering Method Based on Ontology

Yi Ding and Xian Fu

The college of computer science and technology
Hubei normal university
Huangshi , China
{teacher.dingyi,teacher.fu}@yahoo.com.cn

Abstract. Text clustering as a method of organizing retrieval results can organize large amounts of web search into a small number of clusters in order to facilitate users' quickly browsing. In this paper, we propose A text document clustering method based on ontology which is different from traditional text clustering and can improve clustering results performance. We have shown how to include background knowledge in form of a heterarchy in order to generate different clustering views onto a set of documents. We have compared our approach against a sophisticated baseline, achieving a result favorable for our approach.

Keywords: document clustering, ontology, document preprocessing.

1 Introduction

Most of existing search engines often returns a long list of search results, ranked by their similarity to the given query. Web users have to go through the list and title to find their required results. When multiple sub-topics of the given query are mixed together, it would be a time consuming task to find the satisfied results he wants. A possible solution to this problem is to cluster these results into different groups and enable users to find their required clusters at a glance. However, most traditional clustering algorithms cannot be directly used for search results clustering, for some practical issues.

In this paper we have shown how to include background knowledge in form of a heterarchy in order to generate different clustering grand-total from a set of documents. We have compared our approach against a sophisticated baseline, achieving a result favorable for our approach. In addition, we have shown that it is possible to automatically produce results for diverging views of the same input. Thereby, the user can rely on a heterarchy to control and possibly interpret clustering results.

2 Related Works

Most existing text clustering methods use the Vector Space Model (VSM) to represent whole documents. The VSM represents a document as a feature vector of terms, or words appeared in it. Each feature vector contains term-weights, usually term frequency[1,2], of the terms appeared in that document.

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 199–206, 2011.

Similarity between documents is measured using one of several similarity measures such as the Cosine measure, the Jaccard measure. Then use a fast algorithm (such as k-means [3]) to deal with the datasets. Clustering methods based on this model make use of only single-term analysis and seldom make use of any word relation analysis.

In order to improving this problem, the most relevant works are that of Zamir et. al. [4,5]. They proposed a phrase-based document clustering approach based on Suffix Tree Clustering (STC). The method basically involves the use of a tree structure to represent share suffix, which generate base clusters of documents and then combined into final clusters based on connect-component graph algorithm.

3 Ontology-Based Text Document Clustering

3.1 Definition Ontology

An ontology in our framework is defined by:

Definition 1. An ontology is a sign system $O := (V, C, P, H, ROOT)$, which consists of:

(1) A vocabulary V (contains a set of terms); $v_i \subset V$ can map to $c_i \subset C$. In general, one term may refer to several concepts and one concept may be referred to by several terms;
(2) A set of concept C ;
(3) Properties P, one concept C may have several properties;
(4) A hierarchy H : Concepts are taxonomically related by the directed, acyclic, transitive, reflexive relation $H \subset C * C$. H(c_1,c_2) means that c_1 is a subclass of c_2;
(5) A top concept ROOT, For all $c \subset C$ it holds' H(c, ROOT).

3.2 Document Preprocessing: Term Choice (DPTC)

Documents may be represented by a wide range of different feature descriptions. The most straightforward description of documents relies on term vectors. A term vector for one document specifies how often each term from the document set occurs in that document. The immediate drawback of this approach for clustering is the size of the feature vectors. In our example evaluation, the feature vectors computed by this method were of size 46,947, which made clustering inefficient and difficult in principle, as described above.

Term choice, the approach we use here for preprocessing, is based on the feature vectors from SiVeR, but focuses on few terms; hence, it produces a low dimensional representation. Choice of terms is based on the information retrieval measure *tfidf*:

Definition 2. (*tfidf*) *Let tf(i,j) be the term frequency of term j in a document* $d_i \in D^*$, *i= 1,…,N. Let df(j) be the document frequency of term j that counts in how many*

documents term j appears. Then tfidf (*term frequency / inverted document frequency*) *of term j in document is defined by*:

$$tfidf(i, j) = tf(i, j) * \log\left(\frac{N}{df(j)}\right) \tag{1}$$

tfidf weighs the frequency of a term in a document with a factor that discounts its importance when it appears in almost all documents. Therefore terms that appear too rarely or too frequently are ranked lower than terms that hold the balance and, hence, are expected to be better able to contribute to clustering results.

For DPTC, we produce the list of all terms contained in one of the documents from the corpus D^* except of terms that appear in a standard list of stop-words. Then, DPTC choice the dim best terms j that maximize and produces a *dim* dimensional vector for document d_i containing the *tfidf* values, *tfidf*(*i*,*j*) for the *dim* best terms.

$$W(j) = \sum_{i=1}^{N} tfidf(i, j) \tag{2}$$

3.3 Document Preprocessing: Concept Choice and Grand-Total (CCAG)

Our approach for preprocessing, involves two stages. First, CCAG maps terms onto concepts using a shallow and efficient natural language processing system. Second, CCAG uses the concept heterarchy to propose good grand-total for subsequent clustering.

Therefore, we have looked for heuristics to further reduce the number of features. The principal idea of our algorithm lies in navigating the heterarchy top-down splitting the concepts with most *support* into their sub-concepts and abandoning the concepts with least support. Thus, the algorithm generates lists of concepts that appear neither too often nor too rarely. The rationale is that too (in-) frequent concept occurrences are not appropriate for clustering.

$$Support(i, C) := \sum_{\substack{B \in C^* \\ H(B,C)}} cf(i, B) \tag{3}$$

$$Support(C) := \sum_{i=1}^{N} Support(i, C) \tag{4}$$

3.4 Logarithmic Values and Normalized Vectors

The document representations described so far use absolute frequency values for concepts or terms (possibly weighted by *idf*). Considering that the occurrence of terms forms a hyperbolic distribution and, hence, most terms appear only rarely, using the logarithmic value log(x+1) instead of the absolute value x itself seemed reasonable to improve clustering results. Indeed, for all preprocessing strategies given here, we

found that results were only improved compared to absolute values. Hence, all results presented subsequently assume the logarithmic representation of term or concept frequencies. Furthermore, we compared normalized vector representations against absolute or logarithmic values. For this latter comparison, we could not find any interesting differences, with respect to our measure.

Algorithm 1

```
Input: number of dimensions dim, Ontology O with top
concept ROOT document set D*
1 begin
2  set Agenda is Root
3  while continue is true
4   set E is the first common Agenda processing
function;
5   set Agenda is the rest common Agenda processing
function;
6   if E is a concept without sub-concept;
7   then set continue is flase;
8   else
9    if E is not a list then set E is an arbitrarily
ordered list of direct sub-concepts of E;end if;
10   set NE is the Element of E with maximal E
11   set RE is NE from E
12   if RE is not empty then set Agenda is sort s RE
which may be a single concept or a list of concept, as
a whole into Agenda ordering according to; end if;
13   set Agenda is sort s NE which may be a single
concept or a list of concept, as a whole into Agenda
ordering according to;
14   if the Length of Agenda > dim then  set Agenda is
a list identical to Agenda, but excluding the last E;
end if;
15  end if;
16  if the length of Agenda= dim then Output the
Agenda; end if;
17 until continue is FALSE;
18 end
Output: Set of lists consisting of single concepts and
lists of concepts, which describe feature choices
corresponding to different representations of the
document corpus D*.
```

4 Performance Evaluation Approaches

This section describes the evaluation of applying K-Means to the preprocessing strategies SiVeR, DPTC, and CCAG introduced above.

4.1 Setting

We have performed all evaluations on a document set from the tourism domain [7]. For this purpose, we have manually modeled an ontology *O* consisting of a set of

concepts C (C 719), and a word lexicon consisting of 350 stem entries. The heterarchy H has an average depth of 4.2, the longest un-directed path from root to leaf is of length 8.

Our document corpus D^* has been crawled from a WWW provider for computer information (URL: http://www.pconlie.com) consisting now of 234 HTML documents with a total sum of over 170,000 terms.

4.2 Silhouette Coefficient

In order to be rather independent from the number of features used for clustering and the number of clusters produced as result; our main comparisons refer to the silhouette coefficient [6]:

Definition 3 (Silhouette Coefficient). Let $D_M = \{\overline{D_1}, \ldots, \overline{D_k}\}$ describe a clustering result, i.e. it is an exhaustive partitioning of the set of documents D*. The distance3 of a document $d \in D^*$ to a cluster $\overline{D_1} \in D_M$ is given as

$$dist(d, \overline{D_i}) = \frac{\sum_{p \in \overline{D_i}} dist(d, p)}{\left|\overline{D_i}\right|} \tag{5}$$

Let further be $a(d, D_M) = dist(d, \overline{D_i})$ the distance of document d to its cluster $\overline{D_i} (d \in \overline{D_i})$, and $b(d, D_M) = \min_{\overline{D_i} \in D_M, d \notin \overline{D_i}} dist(d, \overline{D_i})$ the distance of document d to the nearest neighbor cluster.

The silhouette $\upsilon(d, D_M)$ of a document d is then defined as:

$$s(d, D_M) = \frac{b(d, D_M) - a(d, D_M)}{\max\{a(d, D_M), b(d, D_M)\}} \tag{6}$$

The silhouette coefficient as:

$$SC(D_M) = \frac{\sum_{p \in D^*} s(p, D_M)}{\left|D^*\right|} \tag{7}$$

The silhouette coefficient is a measure for the clustering quality, that is rather independent from the number of clusters, k. For comparison of the three different preprocessing methods we have used standard K-Means4.However, we are well aware that for high-dimensional data approaches like[2] may improve results – very likely for all three preprocessing strategies. However, in preliminary tests we found that in the low-dimensional realms where the silhouette coefficient indicated

reasonable separation between clusters, quality measures for standard and improved K-Means coincided.

The general result of our evaluation using the silhouette measure was that K-Means based on CCAG preprocessing excelled the comparison baseline. K-Means based on DPTC, to a large extent. K-Means based on SiVeR was so strongly handicapped by having to cope with overly many dimensions that its silhouette coefficient always approached 0 – indicating that no reasonable clustering structures could be found.

4.3 Varying Number of Features Dim and Clusters k

Then we explored how CCAG and DPTC would fare when varying the number of features used and the number of clusters produced by K-Means.

Figure1,2 depicts the dependency between the number of features, *dim*, used and the preprocessing method for a fixed number of clusters. $k = 10$. The line for CCAG shows the silhouette coefficient for the best grand-total from the ones generated by algorithmic 1. We see that for DPTC and CCAG the quality of results decreases as expected for the higher dimensions [1], though CCAG still compares favorably against DPTC.

We have not included the lower bound of CCAG in Figure 1,2. The reason is that so far we have not been very attentive to optimize algorithmic 1 in order to eliminate the worst grand-total up front. This, however, should be easily possible, because we observed that the bad results are produced by grand-total that contain too many overly general concepts like.

In our real-world application we experienced that it is useful to include the user's viewpoint for deriving the number of dimensions with respect to the actual problem. In general one may propose the following upper bound for the number of useable dimensions: The silhouette coefficient decreases below 0.25 using more than 6 dimensions. Thus, using more than 6 dimensions may not be useful, because no meaningful clustering structure may be discovered.

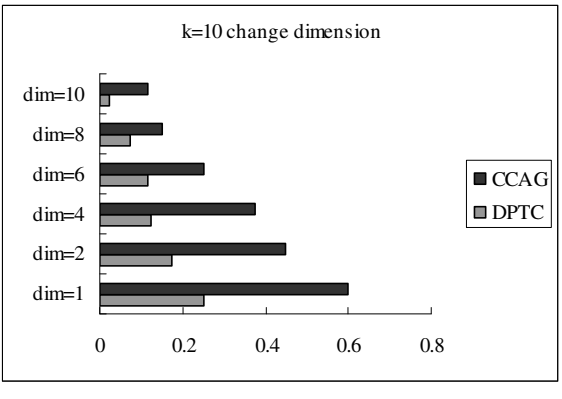


Fig. 1. Comparing DPTC and the grand-total of CCAG

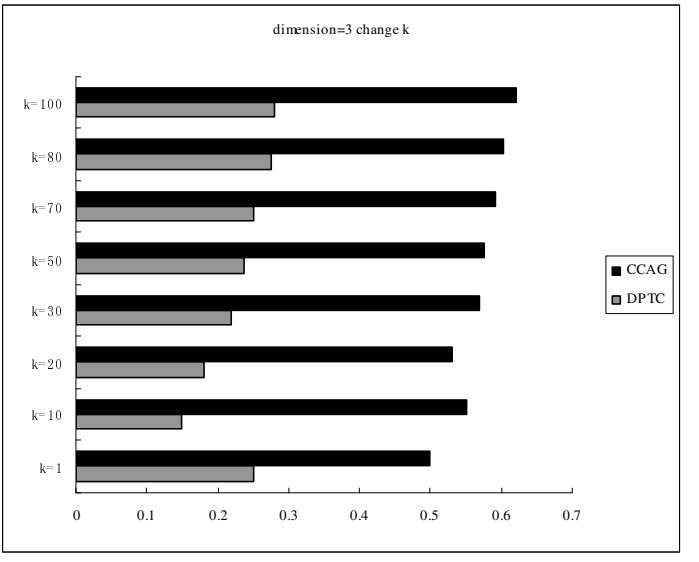


Fig. 2. Comparing DPTC and the grand-total of CCAG

5 Conclusion

In this paper we have shown how to include background knowledge in form of a heterarchy in order to generate different clustering grand-total from a set of documents. We have compared our approach against a sophisticated baseline, achieving a result favorable for our approach. In addition, we have shown that it is possible to automatically produce results for diverging views of the same input. Thereby, the user can rely on a heterarchy to control and possibly interpret clustering results.

Acknowledgment. This work was supported by the project of educational commission of Hubei Province of China from Cooperation of Industry, Academe and Research (Grant No. CXY2009B031), scientific research project of Hubei Normal University (Grant NO.2010C28), education research project of Hubei Normal University (Grant NO. 201031).

References

1. Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U.: When is 'nearest neighbor' meaningful. In: Proceedings of ICDT 2004, pp. 217–235 (2004)
2. Bradley, P., Fayyad, U., Reina, C.: Scaling clustering algorithms to large databases. In: Proceedings of KDD 2003, pp. 9–15. AAAI Press, Menlo Park (2003)

3. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. Journal of the American Society of Information Science 41(6), 391–407 (1990)
4. Devaney, M., Ram, A.: Efficient feature selection in conceptual clustering. In: Proceedings of ICML 1998, pp. 92–97. Morgan Kaufmann, San Francisco (1998)
5. Hinneburg, A., Wawryniuk, M., Keim, D.A.: Hd-eye: visual mining of high-dimensional data. IEEE Computer Graphics and Applications 19(5), 22–31 (1999)
6. Kaufman, L., Rousseeuw, P.: Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, New York (1990)
7. Schuetze, H., Silverstein, C.: Projections for efficient document clustering. In: Proceedings of SIGIR 1997, pp. 74–81. Morgan Kaufmann, San Francisco (1997)

Visual Tracking Using Iterative Sparse Approximation

Huaping Liu[1,2], Fuchun Sun[1,2], and Meng Gao[3]

[1] Department of Computer Science and Technology, Tsinghua University, P.R. China
[2] State Key Laboratory of Intelligent Technology and Systems, Beijing, P.R. China
[3] Shijiazhuang Tiedao University, Hebei Province, China

Abstract. Recent research has advocated the use of sparse representation for tracking objects instead of the conventional histogram object representation models used in popular algorithms. In this paper we propose a new tracker. The core is that the tracking results is iteratively updated by gradually optimizing the sparsity and reconstruction error. The effectiveness of the proposed approach is demonstrated via comparative experiments.

Keywords: Visual tracking, sparse approximation.

1 Introduction

Object tracking is an important problem with extensive application in domains including surveillance, behavior analysis, and human-computer interaction and has attracted significant interests. The particle filter is a popular approach which recursively constructs the posterior probability distribution function of the state space using Monte Carlo integration[3]. However, when a good dynamic model is not available, or the state dimension of the tracked object is high, the number of required samples becomes large and the particle filtering can be computationally prohibitive. In [2], the mean-shift, which is a non-parametric kernel density estimator that can be used for finding the modes of a distribution, is used for visual tracking. At each iteration, the offset between the previous location and the new kernel-weighted average of the samples points is used to find the mean-shift vector that defines the path leading to a stationary point of the estimated density. However, it has been shown that for tracking scenario, constant subspace assumption is more reasonable than constant brightness or color assumptions[5]. Currently a lot of works have been developed to construct suitable subspace representation[5]. But how to get a robust representation remains an open challenge problem.

Recently sparse signal reconstruction has gained considerable interests[1]. Variations and extensions of l_1 minimization have been applied to many vision tasks[9] [4] [8]. In [9], the face recognition problem was solved by l_1 optimization. Ref.[4] and [8] gave further results to deal with registration error and light illumination changes. In [6], a robust visual tracking framework is developed

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 207–214, 2011.

by casting the tracking problem as finding a sparse approximation in a template subspace. During tracking, a target candidate is represented as a linear combination of the template set composed of both target templates and trivial templates. Intuitively, a good target candidate can be efficiently represented by the target templates. This leads to a sparse coefficient vector, since coefficients corresponding to trivial templates tend to be zeros. The sparse representation is achieved through solving an l_1-regularized least squares problem, which can be done efficiently through convex optimization. Then the candidate with the smallest target template projection error is chosen as the tracking result. This approach has been tested on some benchmark dataset and shows advantages over some existing approaches. In addition, Ref.[10] also proposed sparse representation approach for visual tracking, but their approach requires off-line training. Both [6] and [10] utilize the particle filter as the tracker. For each particle and a sparse representation needs to be calculated. Therefore the time cost is very large. To tackle this problem, we give a different way of treating tracking task as iterative optimization problem. The advantages of this approach is that only several iterations need to be calculated. In addition, an extensive experimental comparison shows that the proposed approach is more robust than the approach proposed in [6] and some other classical approaches, such as [2].

2 Proposed Approach

Similar to [6], we assume we have a target template set $\mathbf{T} = [\mathbf{t}_1, \cdots, \mathbf{t}_n] \in \mathbb{R}^{d \times n}$. The template $\mathbf{t}_i$ is generated by stacking template image columns to form a 1D vector.

To incorporate the effect of noise, following the scheme in [9] and [6], we introduce the trivial templates $\mathbf{I} \in \mathbb{R}^{d \times d}$ which is essentially an identity matrix and approximate a potential candidate $\mathbf{t}$ as

$$\mathbf{t} = \mathbf{Bc} \tag{1}$$

where $\mathbf{B} = [\mathbf{T} \;\; \mathbf{I} \;\; -\mathbf{I}] \in \mathbb{R}^{d \times (n+2d)}$ and $\mathbf{c}$ represents the coefficient vector. The role of negative templates $-\mathbf{I}$ has been illustrated in [6].

The introduction of trivial templates $\mathbf{I}$ makes the problem (1) to be underdetermined and there does not exist unique solution to $\mathbf{c}$. A natural approach is to get the sparse solution to (1). To this end, Ref.[6] constructed the following weighted optimization problem:

$$\min \;\; ||\mathbf{t} - \mathbf{Bc}||_2^2 + \lambda ||\mathbf{c}||_1 \tag{2}$$

where $||\cdot||_1$ and $||\cdot||_2$ denote the l_1 and l_2 norms respectively, and λ is the weighting parameter. Furthermore, the reconstruction error $||\mathbf{t} - \bar{\mathbf{B}}\mathbf{c}||_2$ is used for likelihood function evaluation, where $\bar{\mathbf{B}} = [\mathbf{B} \; \mathbf{0} \; \mathbf{0}] \in \mathbb{R}^{d \times (n+2d)}$. The basic assumption of the tracking method developed in [6] is that good candidate can be represented as a sparse linear combination of the templates $\mathbf{B}$. Using this assumption, we re-formulate the tracking task as the following energy minimization procedure:

$$\begin{aligned} \min_{\mathbf{c}} \quad & ||\mathbf{t} - \bar{\mathbf{B}}\mathbf{c}||_2 + \lambda||\mathbf{c}||_1 \\ \text{s.t.} \quad & \mathbf{t} = \mathbf{B}\mathbf{c}. \end{aligned} \tag{3}$$

This optimization problem can be easily solved by using CVX package, which can be downloaded from http://www.stanford.edu/˜ boyd/cvx/.

The proposed tracking is based on the assumption that the solution of the optimization problem corresponds to the target location we seek, starting from an initial guess to the target. Then the problem to be solved is that given an initial guess $\mathbf{t}_0$, how to get the final solution to solve the optimization problem (3). Therefore (3) is further transformed as

$$\begin{aligned} \min_{\mathbf{c},\delta u,\delta v} \quad & ||\mathbf{t}_0 + \delta u\frac{\partial \mathbf{F}_k}{\partial x} + \delta v\frac{\partial \mathbf{F}_k}{\partial y} - \bar{\mathbf{B}}\mathbf{c}||_2 + \lambda||\mathbf{c}||_1 \\ \text{s.t.} \quad & \mathbf{t}_0 + \delta u\frac{\partial \mathbf{F}_k}{\partial x} + \delta v\frac{\partial \mathbf{F}_k}{\partial y} = \mathbf{B}\mathbf{c}. \end{aligned} \tag{4}$$

where $\mathbf{F}_k$ is the current image frame; $\frac{\partial \mathbf{F}_k}{\partial x}$ and $\frac{\partial \mathbf{F}_k}{\partial y}$ can be obtained by the central difference; δu and δv are the motion parameter to be estimated. The weighting factor λ can be set to 1. In practical tracking scenario, this optimization procedure should be repeated for several iterations. The whole tracking algorithm is summarized in Algorithm 1, where EPS and MAX_NUM are the error tolerance and the maximum iteration number, respectively. They should be prescribed by the designer.

Algorithm 1. Proposed visual tracking algorithm

Given: Template matrix $\mathbf{B}$; Previous tracking position $\hat{x}_{k-1}$,$\hat{y}_{k-1}$, and scale $\hat{s}_{k-1}$; Current image frame $\mathbf{F}_k$.
OUTPUT: Current tracking position $\hat{x}_k$, $\hat{y}_k$.

Initialization: $u^0 = 0$, $v^0 = 0$, $i = 0$.
Do:
- $x = \hat{x}_{k-1} + u^i$, $y = \hat{y}_{k-1} + v^i$, $s = \hat{s}_{k-1}$
- Extract sample $\mathbf{t}$ from the box determined by x, y, and s.
- Solve the following optimization problem:

$$\begin{aligned} \min_{\mathbf{c},\delta u,\delta v} \quad & ||\mathbf{t} + \delta u\frac{\partial \mathbf{F}_k}{\partial x} + \delta v\frac{\partial \mathbf{F}_k}{\partial y} - \overline{\mathbf{B}}\mathbf{c}||_2 + \lambda||\mathbf{c}||_1 \\ \text{s.t.} \quad & \mathbf{t} + \delta u\frac{\partial \mathbf{F}_k}{\partial x} + \delta v\frac{\partial \mathbf{F}_k}{\partial y} = \mathbf{B}\mathbf{c}. \end{aligned} \tag{5}$$

- $u^{i+1} = u^i + \delta u, v^{i+1} = v^i + \delta v$.
- err $= ||[\delta u \;\; \delta v]||_2$.
- $i \leftarrow i + 1$.

While ((err $<$ EPS) AND ($i <$ MAX_NUM))
$\hat{x}_k = x$, $\hat{y}_k = y$.

The above iteration converges within about 3 to 4 iterations. Once we have obtained final estimates of x_k and y_k, we can further estimate the scale. In this paper, we use a heuristic approach to estimate the scale. Denote $\hat{s}_{k-1}$ as the estimated scale at the last frame. After running Algorithm 1 to get the estimated central point $\hat{x}_k, \hat{y}_k$, we extract three samples from the bounding boxes

determined by three different scales $(1-\alpha)\hat{s}_{k-1}$, $\hat{s}_{k-1}$, and $(1+\alpha)\hat{s}_{k-1}$. After calculating the resulting reconstruction errors made by the three samples, we select the scale which leads to the minimum reconstruction error to be the estimated scale $\hat{s}_k$. In practical scenario, The coefficient α is usually set to be 0.1 and we find this setting is very effective in practice. Finally, although the above algorithm description models the motion of the target as pure translation, it can be easily extended to other motions, such as rotation. For brevity we donot give further discussions.

Remark 1. The construction of template set is similar to [6]: At initialization, the first target template is manually selected from the first frame. The rest target templates are created by perturbation of one pixel in four possible directions at the corner points of the first template in the first frame.

It should be noted that the proposed optimization scheme will not perform a global search to find the image region that matches the stored representation. Rather, given an initial guess, it will refine the position and reconstruction.

In the following we give a brief comparison between the most related work[6] and the proposed approach. The core difference between the two approaches is a little like the difference between conventional mean-shift approach[2] and particle filter approach[7]. In [6], the reconstruction error is used to evaluate the likelihood function. However, how to construct the likelihood function is still an open problem and needs setting some tuning parameters. In addition, the performance of particle filter in [6] strongly depends on some other parameters, such as the parameters in dynamic equation. In the proposed approach, however, the tracking is totally automatically realized by optimizing the objective function (4). On the other hand, the time costs are rather different. In [6], assume we have N particles, then each particle needs to be represented as the sparse combination of the templates, and therefore N times sparse decompositions are required. In practical implementation of particle filter, the number of particles N is usually set to be a large number. This leads to time-consuming computations. In our approach, the algorithm usually converges within $3\tilde{}\ 4$ iterations.

Finally, it should be emphasized that in practical tracking scenarios, the template set $\mathbf{B}$ should be updated to incorporate the changes of appearance, illumination, and so on. In [6], a heuristic update approach is proposed and it also applies in our algorithm. However, due to the space limitation, we can only focus on the tracking itself but not update method. In spite of this, we will show in the next section that the proposed approach which utilizes fixed template set still performs rather well on many public available testing sequences.

3 Experimental Results

We make extensive experimental comparisons and provide some representative results in this section. For comparison, we consider the conventional color-based mean-shift approach[2](**MS**) and the approach in [6](**MEI**). The former is a

classical approach, which utilizes the color information to guide the shift direction. In [6], a particle filter is utilized and therefore some parameters should be given. We define the states of the particle filter to be $\mathbf{x}_k = [x_k, y_k, s_k]$, where x_k, y_k indicates the locations of the object; s_k is the corresponding scale. The dynamics can be represented as $\mathbf{x}_k = \mathbf{x}_{k-1} + \mathbf{v}_k$, where $\mathbf{v}_k$ is a multivariate zero-mean Gaussian random variables. The variance is set by $[\sigma_x, \sigma_y, \sigma_s] = [2, 2, 0.01]$. The particle filter is assigned to 50 particles. The authors of [6] suggested that the likelihood function to be formulated from the error approximated by the target templates but they did not give precise form. Therefore we construct the likelihood using $\exp(-err)$, where err is the reconstruction error. On the other hand, since in this work we do not include model update module, we also remove update stages of **MEI**. Please note that we do not compare to [10], because it requires a lot of training samples to be off-line collected. In all of the presented images, the *magenta*, *blue* and *red* boxes correspond to the **MS**, **MEI** and the proposed approach, respectively.

Table 1. Quantitative comparisons (The averaged position errors in pixels)

Sequence	I(0)	II(0)	III(2)	IV(3)	V(7)	VI(0)	VII(18)	VIII(3)	VIIII(2)	X(0)
Number of frames	382	295	234	272	285	855	771	398	310	559
MS	5.4	9.0	8.9	10.9	8.6	11.4	43.1	37.2	37.7	54.4
MEI	8.4	22.0	50.5	63.6	34.9	16.8	62.0	70.6	9.9	10.8
Proposed	1.3	5.2	2.4	3.6	4.6	8.9	10.4	8.7	3.8	4.0

The first sequence was captured by us using a camera which is mounted in the top of a car. The tracking results are shown in Fig.1. In this sequence, there is a car running in a tunnel and the lighting condition changes rapidly. **MS** tracker quickly loses the target after tracking a few frames (see Frame 61) and never recovers. **MEI** approach can track the target for about 200 frames and our algorithm is able to track the car throughout the entire 530 frames of the sequence.

The second sequence (*ThreePastShop2cor*) comes from CAVIAR project(see http://homepages.inf.ed.ac.uk/rbf/CAVIAR/). The tracking results are shown in Fig.2. In this sequence, 3 persons walk in the corridor. We initialize the detection results at Frame 398 and the leftmost man in black clothes is selected as the target. In the following frames, this man walks from left to the right. From Frame 465 to 512, there occurs some occlusions. During the occlusion period, the performance of **MS** is deteriorated and the tracking result is even wrongly locked onto the man in red clothes (Note that color information is not used by **MEI** and us). After this, the target walks along the corridor and the scale is smaller and smaller. Nevertheless, our approach accurately tracks in both position and scale (see Frames 627 and 939). Since CAVIAR dataset provides detailed annotations, we can quantitatively compare the three approaches. The left part of Fig.Fig.4 gives the tracking error, which is defined as the position errors between the centers of the tracking results and that of the ground truth. It shows that only the proposed approach succeeds in tracking during the whole period.

Fig. 1. A car in the tunnel. LEFT to RIGHT: Frames 1, 61, 242, 378, 529.

Fig. 2. CAVIAR *ThreePastShop2cor*. LEFT to RIGHT: Frames 398, 465, 512, 627, 939.

Fig. 3. CAVIAR *OneLeaveShopReenter2cor*. LEFT to RIGHT: Frames 2, 207, 225, 245, 559.

The third sequence (*OneLeaveShopReenter2cor*) has been investigated in Ref.[6]. In [6] this sequence is initiated at Frame 137 and the approach in Ref.[6] is able to track up to the 271st frame. However, They did not mention the performance for the remaining frames. In our experiment, this sequence is initialized at the first frame and the whole 559 frames are investigated. In this case, Both **MEI** and our approach are able to track during the period when the woman is partially occluded by the man (from Frame 190 to Frame 238) while **MS** fails. After the occlusion, Both **MEI** and our approach still track the woman. However, our algorithm is able to track the object more stably, and hence closer to the ground truth than **MEI**(See Fig.3 for some selected frames and Fig.5 for performance comparison).

Table I gives a quantitative comparison on 10 sequences. Seq.I is *EnterExitCrossingPaths1cor*, Seq.II is *OneLeaveShop2cor*, Seq.III is *OneLeaveShopReenter1cor*, Seq.IV is *OneShopOneWait1cor*, Seq.V is *OneShopOneWait2cor*, Seq.VI is *OneStopNoEnter2cor*, Seq.VII is *ShopAssistant2cor*, Seq.VIII is *TwoEnterShop2cor*, Seq. VIIII is *TwoEnterShop3cor*, Seq.X is *OneLeaveShopReenter2cor*. The number in the parenthese is the object ID which is defined by CAVIAR ground truth. We use fixed parameter setting $[\sigma_x, \sigma_y, \sigma_s] = [2, 2, 0.01]$ for all sequences and see that the proposed approach clearly outperforms the other methods. Based on the un-optimized MATLAB implementation, the average time cost per frame is 50 seconds for **MEI**, and 15 seconds for our approach. The performance can be further improved by operating the tracking at multiple levels of resolution. An extensive research on improving the efficiency will be the subject of future works.

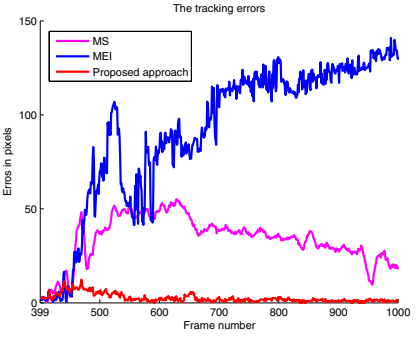


Fig. 4. Performance comparison for Sequence *ThreePastShop2cor*

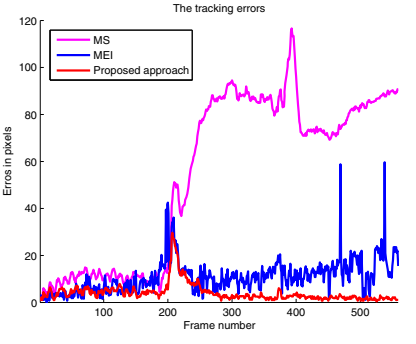


Fig. 5. Performance comparison for Sequence *OneLeaveShopReenter2cor*

4 Conclusion

We presented a robust algorithm to track object in video sequences using sparse representation and successive optimization. The proposed approach shows better performance than existing mean-shift tracker and the recently proposed particle filter using l_1 minimization.

Acknowledgments

This work is jointly supported by the National Natural Science Foundation of China(Grants No. 90820304, 61075027), the National Key Project for Basic Research of China(Grant No. G2007CB311003), and the Natural Science Foundation of Hebei Province(F2010001106).

References

1. Candes, E., Tao, T.: Near-optimal signal recovery from random projections: Universal encoding strategies? IEEE Trans. Information Theory, 5406–5425 (2006)
2. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Trans. on Pattern Analysis and Mechine Intelligence 25, 564–577 (2003)
3. Canton-Ferrer, C., Casas, J., Pardas, M.: Particle filtering and sparse sampling for multi-person 3D tracking, In: Proc. of Int. Conf. on Image Processing, pp. 2644–2647 (2008)
4. Huang, J., Huang, X., Metaxas, D.: Simultaneous image transformation and sparse representation recovery. In: Proc. of Computer Vision and Pattern Recognition, pp.1–8 (2008)
5. Lim, J., Ross, D., Lin, R., Yang, M.: Incremental learning for visual tracking, In: Proc. of Neural Information Processing Systems, pp. 793–800 (2004)
6. Mei, X., Ling, H.: Robust visual tracking using l_1 minimization. In: Proc. of Int. Conf. on Computer Vision, pp.1–8 (2009)
7. Pérez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-based probabilistic tracking. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 661–675. Springer, Heidelberg (2002)
8. Wagner, A., Wright, J., Ganesh, A., Zhou, Z., Ma, Y.: Towards a practical face recognition system: Robust registration and illumination by sparse representation, In: Proc. of Computer Vision and Pattern Recognition, pp. 597–604 (2009)
9. Wright, J., Yang, A., Ganesh, A., Sastry, S., Ma, Y.: Robust face recognition via sparse representation. IEEE Trans. on Pattern Analysis and Machine Intelligence 31, 210–227 (2009)
10. Zhang, J., Cai, W., Tian, Y., Yang, Y.: Visual tracking via sparse representation based linear subspace model, In: Proc. of Int. Conf. on Computer and Information Technology, pp.166–171 (2009)

Orientation Representation and Efficiency Trade-off of a Biological Inspired Computational Vision Model

Yuxiang Jiang and Hui Wei

Cognitive Algorithm Laboratory, School of Computer Science,
Fudan University,
825 Huatuo Road, Shanghai, China
jiangyuxiang@gmail.com,
wehui@fudan.edu.cn

Abstract. Biological evolution endows human vision perception with an "optimal" or "near optimal" structure while facing a large variety of visual stimuli in different environment. Mathematical principles behind the sophisticated neural computing network facilitate these circuits to accomplish computing tasks sufficiently as well as at a relatively low energy consumption level. In other words, human visual pathway, from retina to visual cortex has met the requirement of "No More Than Needed" (NMTN). Therefore, properties of this "nature product" might cast a light on the machine vision. In this work, we propose a biological inspired computational vision model which represents one of the fundamental visual information — orientation. We also analyze the efficiency trade-off of this model.

Keywords: computational vision model, orientation representation, efficiency trade-off.

1 Introduction

In the very beginning of human vision pathway, retina deals with visual stimuli with six interconnected layers [1]. Responses output by the final layer are collected by Lateral Geniculate Nuleus (LGN) and further transmitted to visual cortex for semantic extraction and representation. Around 75% of ganglion cells in retina contribute to Parvocellular vision pathway [2]. These ganglion cells highly response to certain wavelength stimuli and their center-periphery antagonized response patterns are well simulated by Differences of Gaussian (DoG) functions [5]. According to Hubel's description [3], simple cells that packed in primary visual cortex are selectivie to orientation which was considered to be the most basic component for making up more complicated visual representation. For image processing, objects can be considered as combinations of linear segments as shown in figure 2. These tiny segments are modeled as *linear segments* in our model which are the basic (meta) components of rebuilding apicture.

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 215–224, 2011.

Biologically, how does this representation emergy still remains unclear for scientists. Hubel and Wiesel [3] proposed a hypothesis that explained the structure of a simple cell's receptive field and some anatomical evidence have been collected [4] seemingly to support it. Receptive fields of ganglion cells, in this case, have to align strictly inside each simple cell's receptive field. Although it was an elaborate structure, models built under this assumption will certainly suffer from rigid geometric requirements which are very expensive to implement on hardwares. We alleviate the spatial requirement and propose an alternative model in which "Simple Cell like" (SC-like) neurons can compute and get the orientation information with fewer properly located ganglion cells.

Also, it is worth noting that biological evolution deals with the trade-off between performance and energy consumption extremely well. Since better performance usually yields to higher energy consumption, biological structures have been working in a mechanism that best fit the environment. This means if the current structure has just satisfied surviving requirements (e.g. visual computing requirement for visual structures), there is no need to evolve into better ones which may cost more energy and resources. This principle is termed as "No More Than Needed" (NMTN) in this paper. Much evidence can be found in retina to support it. For example, foveal center area has been evolved to have exceptionally high resolution for precisely discerning objects where we are focusing. However, this high performance can not be afforded or needed throughout the entire retina. We will follow this principle in the design of our computational model.

The entire computational model will be developed in section 2. We will discuss the geometry constraints for ganglion cells to achieve the NMTN principle and explain how does an SC-like neuron work in this section. In section 3, we analysis the energy efficiency of different geometry pattern and section 4 will show some picture-rebuilding results produced by this model. Finally, we will end up with a discussion and future works about this model in section 5.

2 Computational Vision Model

We proposed a three-layer computational vision model, see figure 1. Photoreceptors collect information from an input picture; ganglion cells (GC) pre-process

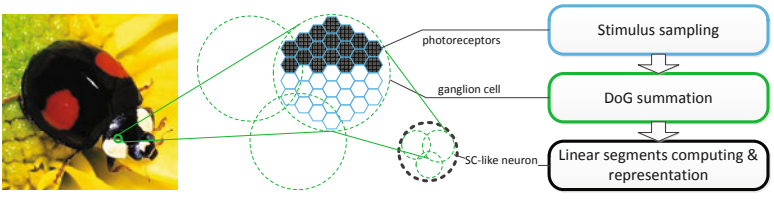


Fig. 1. Ganglion cells (marked as green dash circles) get inputs from photoreceptors (marked as blue hexagons within a ganglion cell's receptive field) and several neighboring ganglion cells work together making up the receptive field of an SC-like neuron (marked as a black dash circle) to compute orientation representation

the responses from photoreceptors with respect to a DoG model, and final representations are computed by SC-like neurons.

2.1 Unstable Part of a Ganglion Cell's Response Curve

Decades ago, Rodieck [5] proposed the DoG (Differences of Gaussian) model for ganglion cells which simulate biological data quite smoothly [6]. Basically, we employ the standard two dimensional DoG function $\mathcal{F}$, which is defined as:

$$\mathcal{F}(\boldsymbol{x}|\sigma_0,\sigma_1) = \mathcal{N}(\boldsymbol{x}|\sigma_0) - \mathcal{N}(\boldsymbol{x}|\sigma_1) \quad (\sigma_0 < \sigma_1) \tag{1}$$

$$= \frac{1}{2\pi\sigma_0^2}\exp\left(-\frac{x_1^2+x_2^2}{2\sigma_0^2}\right) - \frac{1}{2\pi\sigma_1^2}\exp\left(-\frac{x_1^2+x_2^2}{2\sigma_1^2}\right), \tag{2}$$

where the parameters σ_0, σ_1 denote the standard deviations of inner (positive) and outer (negative) gaussian function respectively. This function models the weighting distribution of inputs to a single ganglion cell. Since the integral over the entire domain is zero, a uniform stimulus within the receptive field will make a ganglion cell to output 0.

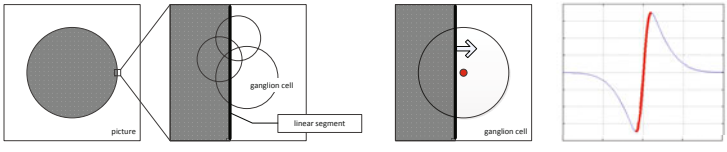


Fig. 2. A segment of an object's boundary can be viewed as a straight short line locally. We term it as "linear segments" which is marked as a thick line in the second subfigure from left. A ganlgion cell's response curve of such a segment crossing over its receptive field is plotted in the last subfigure. The middle part (red part) of that curve which caused by a segment crossing over the positive region (marked as a red blob) is so-called the "unstable part".

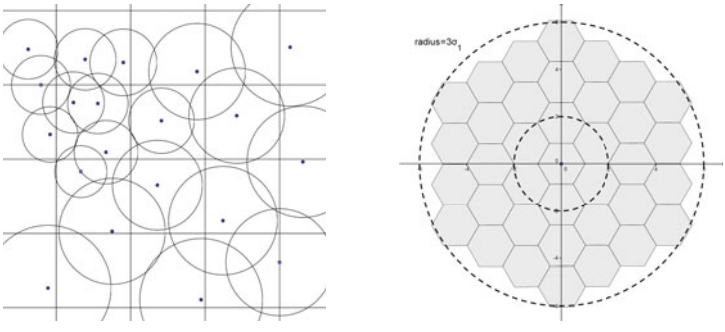
According to the data reported by Corner and Kaplan [7], the outer-inner standard deviation (σ_1/σ_0) in DoG with respect to parvocellular ganglion cells remains around 6–8. Ganglion cell's response curve flips quickly in the middle (red thick curve) part, which means a tiny difference of a segment across this positive region will yield to a significant output change, see figure 2. This phenomenon renders the output unreliable to determine the exact position of that segment. Therefore, we abandon this "unstable part" of the curve.

2.2 Receptive Field of Ganglion Cells

Some visual tasks may require various resolutions to represent a scene. For example, given the task that to recognize a person in a crowded street usually need high resolution on his/her face, however, background objects like vehicles, trees,

and other pedestrians can be represented with lower resolution in order to save computing resources. Thus, taking pixels as basic units seems to be incompetent. Inspired by the nonuniform sampling of retina, *pixel-independent* sampling methods like proposed in [8] are designed to overcome this problem. Such a method considers each pixel as continuous color patch in order to get rid of the restrict of pixels.

We, in this model, employ a pixel-independent sampling technique, see figure 3(a). Each ganglion cell's receptive field consists of hexagon-like packed photoreceptors. see figure 3(b).



(a) pixel-independent sampling (b) integration strategy

Fig. 3. (a) shows the pixel-independent sampling method. Each square denotes a single pixel, while the circles denote ganglion cells' receptive fields. Pixels are considered as continuous color patches (i.e. those squares with width and height) and ganglion cells with various densities and sizes are responsible for summing up sampled signals inside its receptive field. (b) describes the strategy for computing photoreceptors' weightings in DoG model for a single ganglion cell. We integrate the corresponding hexagon region in $\widehat{xoy}$ plane to get corresponding weightings. Outer dash circle indicates $3\sigma_1$ and inner circle indicates σ_1. σ_1 is the parameter in formula (11).

2.3 The Definition of Ganglion Cell "Neighbors"

In our model, two ganglion cells that are both connected to the same SC-like neuron may have interactions and stay close to each other. They both contribute to the receptive field of an SC-like neuron. Such two ganglion cells are termed as *neighbors*. Intuitively, if a ganglion cell has a neighbor, the geometrically nearest ganglion cell should be considered as that neighbor (Note that it may have more than one neigbor.). Also, we define the receptive field of an SC-like neuron to be composed of neighboring ganglion cells, i.e. any two ganglion cells within this field are neighbors. Apparently, at least three ganglion cells are needed to determine the position of a linear segments and according to the NMTN principle, SC-like neuron's receptive field is, therefore, composed of three ganglion cells. Each two of them are neighbors. If we take each ganglion cell as a node in a graph, the problem that to define neighbors for ganglion cells

can be related to 2-D triangulations. Edges in the triangulated graph denote neighboring relationships.

Definition 1. *Let the center of each ganglion cell's receptive field be a node in a graph, two cells are* neighbors *if and only if their centers are connected in the result graph of the Delaunay triangulation.*

According to the properties of *Delaunay Triangulation*, Nearest Neighbor Graph is a subgraph of the Delaunay triangulation which corresponds to our intuitive assumption.

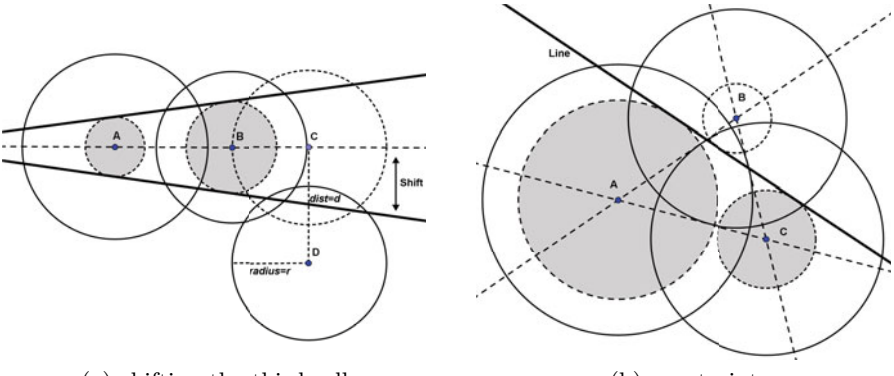
2.4 The Constraint on the Size-Position Relation of Ganglion Cells

The relation between the size and the position of a ganglion cell's receptive field has to meet some requirements that enable it correctly determine linear segments. It should guarantee the ability of discerning response patterns incurred by different linear segments (In other words, sufficient representation ability). Also, the entire system has to maintain a low energy consumption level in order to achieve the optimal trade-off following the NMTN principle.

So, we have two assumptions before discussing the constraint:

1. Smaller number of ganglion cells lead to lower energy consumption level.
2. Smaller receptive field leads to smaller scale of inter-cell-connections that make up the receptive field. In this case, it costs less computing resouces.

Definition 2. *For two or more ganglion cells that have non-zero responses, they are* consistent *with the responses if there is at least one linear segment that satisfies all their responses.*



(a) shifting the third cell (b) constraint

Fig. 4. Illustration of the constraint on size-position relation. Note that shaded response circle denote their responses are positive, while blank ones like cell B in (b) represent negative responses.

As we have discussed, the response of a single ganglion cell uniquely corresponds to the distance from a linear segment to its receptive field center. Nevertheless, we can not determine the exact position of that segment only by the response of one ganglion cell. For "two cells" case, although we can eliminate most of possible solutions, there still remain two candidates which can not be told apart, see figure 4(a).

Definition 3. *Let an SC-like neuron consists of two or more ganglion cells, it has a* weak discerning ability *if more than one possible linear segment satisfy all ganglion cells' responses.*

Admittedly, discerning ability of a neuron's receptive field with two ganglion cells is always weak. Moreover, as long as their centers stay on the same line, this "weakness" still remains, even if the number of ganglion cell increases. See ganglion cell A, B, C in figure 4(a). Therefore, the third ganglion cell C has to shift away from the line to some place like D in order to overcome the weakness.

Let us take a triple-cell set like shown in figure 4(b) as an example. Other than ganglion cells A and B, the third ganglion cell, as discussed above, shifts away from the dash line with distance $d(d \neq 0)$. For the sake of convenience, we introduce a function $G : \mathbb{R} \to \mathbb{R}$, which maps the response of a ganglion cell to the distance from a segment to its receptive field center. According to our previous assumption 2, the receptive field of the third ganglion cell D should be as small as possible. However, it is easy to show that this radius is at least d.

Proof. Suppose its radius $r < d$, say $r = d - \epsilon$, we can easily find a situation that make this triple-cell *weak*, for instance,

$$G(R_A) = G(R_B) \leq \epsilon$$

where R_A, R_B represent the responses of the first two ganglion cells.

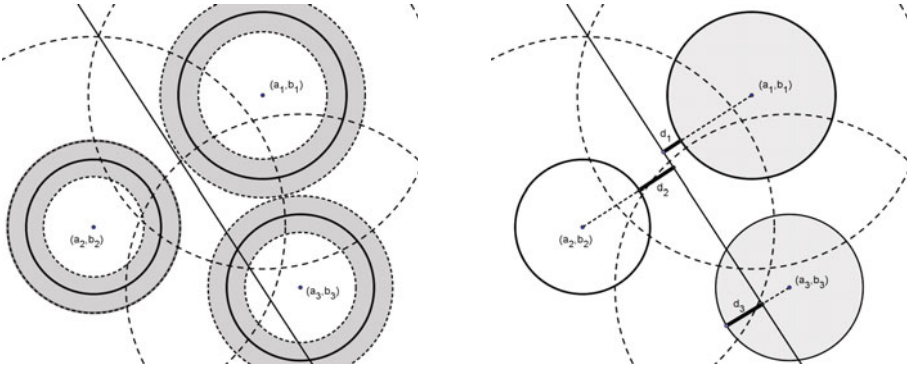
Hence, we get $r \geq d$. On the other hand, it should be small enough so as to achieve low level energy consumption which renders the optimal radius $r = d$.

For each ganglion cell in a triple-cell set should meet the requirement as described in figure 4(b). The radius of each receptive field must be equal to the distance from its center to the line connecting the other two neighbors' centers. More generally, given a ganglion cell and all its k neighbors (it must be involved in k triples around), let the minimum requirement of each triple-cell set be $r_1, r_2, \ldots, r_k$, then the radius of its receptive field should be $\max_{i=1}^{k} r_i$.

2.5 SC-Like Neurons

Hopefully, a linear segment is the common tangent of three response circles, see figure 4(b). Unfortunately, system bias usually lead to inconsistency of these cells. Thus, we need to release the rigid requirement by introducing *tolerance*, see the shaded ring in figure 5(a).

An SC-like neuron attempts to get the optimal linear segment which satisfies all responses. Given that the equation of a line: $u_1x_1 + u_2x_2 + 1 = 0$, and



(a) solution tolerance (b) optimize solution

Fig. 5. Getting the optimal solution of an detected segment

let $(a_1, b_1), (a_2, b_2), (a_3, b_3)$ be the centers of three receptive fields with radius r_1, r_2, r_3 respectively. The optimal linear segment minimizes the sum of square distances from that segment to each response circle along their radius direction. So we try to optimize the object function:

$$\min_{\boldsymbol{u}} f(\boldsymbol{u}) = \min_{\boldsymbol{u}} \sum_{i=1}^{3} \left(\frac{|\boldsymbol{u}^T \boldsymbol{c}_i + 1|}{\|\boldsymbol{u}\|_2} - r_i \right)^2 \tag{3}$$

where $\boldsymbol{u} = [u_1, u_2]^T, \boldsymbol{c}_i = [a_i, b_i]^T$.

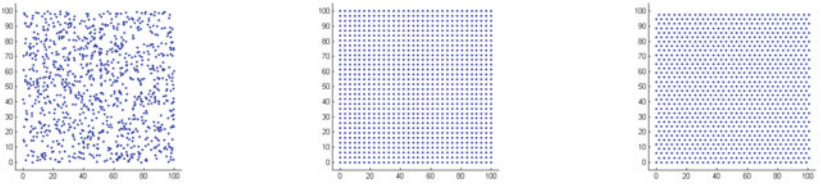
Under this framework, finally we can get an optimal linear segment $\hat{u_1} x_1 + \hat{u_2} x_2 + 1 = 0$, and $f_{\min}(\hat{\boldsymbol{u}}) = residual$. If this $residual$ is smaller than a certain threshold then report the solution, otherwise, report inconsistency. In our experiment in section 4, we use $\sum_{i=1}^{3} 0.25 r_i^2$ as the threshold.

3 Efficiency Analysis with Difference Cell Distribution

In order to compare the efficiency of the computational vision model in different ganglion cell distribution patterns, we generate three different distributions of ganglion cells: randomly packed, matrix-like packed and hexagon-like packed, all in a square region of $[0, 100] \times [0, 100]$, see figure 6.

The generated cells have to meet the requirement discussed in the above sections. Thus, different distribution cells lead to different mean radius of receptive fields. Based on hexagon-like packed cells, we also add a series random disturbance with $\epsilon_{offset} \sim \mathcal{N}(0, \boldsymbol{\sigma}^2)$, see table 1 for result.

The experiment shows hexagon-like packed ganglion cells yields to smallest mean radius. In other words, hexagon-like packed ganglion cells require lowest hardware complexity. This result also seems to be connected with the hexagonally packed photoreceptors (especially cones in retinal fovea).



(a) random packed (b) matrix-like packed (c) hexagon-like packed

Fig. 6. Three different cell distribution patterns within $[0, 100] \times [0, 100]$. Each point denotes a receptive field center.

Table 1. Mean radius of receptive fields with different distribution patterns

Distribution(no offset)	Mean radius
random	3.9245
matrix-like	3.0579
hexagon-like	2.9557

Offsets (σ)	Mean radius
0 (no offset)	2.9557
0.05	3.0282
0.1	3.0992
0.15	3.1671
0.2	3.2413
0.25	3.2910
0.5	3.6185

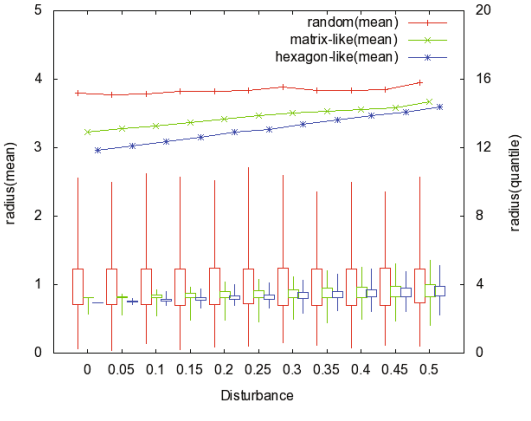


Fig. 7. Mean radius of receptive field and dispersion under different distribution patterns (including disturbance)

4 Representation of the Output of SC-Like Neurons

Based on the outputs of SC-like neurons, we rebuild some pictures using linear segments represented by these neurons, see figure 8.

For each valid output of SC-like neurons we use an elongated gaussian function to represent a linear segment. These gaussian functions can be viewed as building blocks to rebuild a picture. It is worth noting that although the result shown looks like those processed by edge detection filters, they are actually composed of many linear segments. In this case, these segments are much more steerable than pixels from at least two aspects: firstly, the segments number is much smaller than that of pixels in an ordinary picture; secondly, rebuilt picutres can be arbitrarily rescaled. From the results we can see the simplified computational vision model basically preserves the orientation information of input pictures.

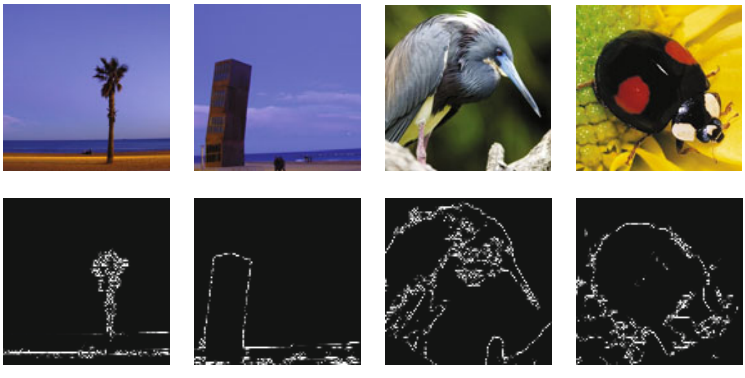


Fig. 8. Rebuilding the original pictures in an 100×100 square from SC-like neurons output. Each rebuilt picture is actually composed of many linear segments.

5 Discussion

Based on some biological facts, we design a three-layer computational vision model to represent orientation information, and try to reconstruct the original pictures on an SC-like neuron level. This kind of representation is fundamentally different to some local descriptors such as SIFT [9], SURF [10] or HoG [11] etc. which detect the gradient orientation information. Firstly, It is independent on resolution of input pictures due to the pixel-independent sampling strategy. Secondly, it provides an interface that is more steerable in further semantic processing.

Also, we hope to refine this model for further analyzing the optimal efficiency for the model. For example, introducing spacial-variant sampling like mentioned in [12] [13] [14] inspired by biological features and quantitatively analyzing its advantages in redundancy reduction. In addition, we would like to compare with the biological structures to get more valuable conclusions. For future works, we

need to consider more parameters to find out the optimal trade-off of computational vision model. These parameters also involve parallel color pathways and non-linear ganglion cell response functions, etc.

Acknowledgement

This work was supported by 973 Program (Project No. 2010CB327900) and NSFC major project (Project No. 30990263).

References

1. Kolb, H.: How the Retina Works. American Scienctist 91, 28–35 (2003)
2. Lee, B.B.: Receptive Field Structure in the Primate Retina. Vision Res. 36, 631–644 (1996)
3. Hubel, D.H., Wiesel, T.N.: Receptive Fields, Binocular interaction and functional architecutre in the cat's visual cortex. J. Physiology 160, 106–154 (1962)
4. Zhan, X., Shou, T.: Anatomical Evidence of Subcortical Contributions to the Orientation Selectivity and Columns of the Cat's Primary Visual Cortex. Neurosci. Lett. 324, 247–251 (2002)
5. Rodieck, R.W.: Quantitative analysis of cat retinal ganglion cell response to visual stimuli. J. Vision Res. 5, 583–601 (1965)
6. Kier, C.K., Buchsbaum, G., Sterling, P.: How retinal microcircuits scale for ganglion cells of different size. J. Neurosci. 15 (1995)
7. Croner, L.J., Kaplan, E.: Receptive Fields of P and M Ganglion Cells Across the Primate Retina. Vision Res. 35, 7–24 (1995)
8. Wei, H., Jiang, Y.: An Orientational Sensitive Vision Model Based on Biological Retina. In: 2nd International Conference on Cognitive Neurodynamics (2009)
9. Distinctive Image Features from Scale-invarient Keypoints. Internatinoal Journal of Computer Vision (2004)
10. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded Up Robust Features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
11. Dalal, N., Triggs, B.: Histograms of Oriented Gradiants for Human Detection. In: Computer Vision and Pattern Recognition (2005)
12. Wallace, R.S., Ong, P.W., Bederson, B.B., Schwartz, E.L.: Space-Variant Image-Processing. International Journal of Computer Vision 13, 71–90 (2004)
13. Shah, S., Levine, M.D.: Visual Information Processing in Primate Cone Pathways — Part I: A Model. IEEE Transaction on Systems, Man and Cybernetics, Part B 26, 259–274 (1996)
14. Balasuriya, S., Siebert, P.: A Biologically Inspired Computational Vision Frontend based on a Self-Organised Pseudo-Randomly Tessellated Artificial Retina. In: International Joint Conference on Neural Networks (2005)

The Application of Genetic Algorithm Based Support Vector Machine for Image Quality Evaluation

Li Cui* and SongYun Xie

Electrical engineering school,
Northwestern Polytechnical University, 710072, China
{l.cui,syxie}@nwpu.edu.cn

Abstract. In this paper, we have proposed a novel image quality evaluation algorithm based on the Visual Difference Predictor(VDP), a classical method of estimating the visual similarity between an image under test and its reference one. Compared with state-of-the-art image quality evaluation algorithms, this method have employed a genetic algorithm based support vector machine, instead of linear or nonlinear mathematical models, to describe the relationship between image similarity features and subjective image quality. Subsequent experiments shows that, the proposed method with the state-of-the-art image quality evaluation algorithms the Mean Square Error (MSE), the Structural SImilarity Metric (SSIM), the Multi-scale SSIM (MS-SSIM). Experiments show that VDQM performs much better than its counterparts on both the LIVE and the A57 image databases.

Keywords: image quality evaluation, automated feature selection, genetic algorithm, support vector machine.

1 Introduction

Image quality evaluation is an important topic of designing imaging systems and evaluating their performance, where various image operations (e.g., acquisition, transmission, processing and transmission) always attempts to maintain and even improve the quality of digital images.

Assuming humans to be the potential consumer of electrical imaging products, subjective methods dominated image quality evaluation for quite a long time. However, their complex and cumbersome experiment procedures hindered their application in real time cases. Therefore objective image quality evaluation using computer algorithms became more popular in recent years [3], where image quality estimated by image quality evaluation algorithms should be closely consistent with subjective perception to image quality. Image quality assessment algorithms usually indicate the perceptual quality of a distorted image by a figure of merit, which is also called the image quality metric.

During the development of image quality evaluation algorithms, psychologists laid on the foundation by reveal properties of the Human Vision System (HVS), while image processing experts attempted to simulate the whole functionality of the HVS using

* Corresponding author.

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 225–231, 2011.

computer algorithms, whose underlying mathematical models can be built either from conclusions obtained from psychological experiments or by treating the HVS as a black box.

Based on their requirement on reference images, image quality evaluation algorithms can be divided into: Full Reference (FR), Reduced Reference (RR) and No Reference (NR) algorithms, where full, partial, and no information about the reference image is needed respectively. This paper mainly discusses the development of FR image quality evaluation metrics.

The first approach to image quality evaluation is to estimate the image quality by computing the physical difference between distorted image signals and their reference ones, e.g., Mean Square Error (MSE), Signal Noise Ratio (SNR) and Peak Signal Noise Ratio (PSNR) metrics. It is reported that image quality estimated by these simple methods usually can not approximate well with subjective ratings.

One of the solutions to this problem is to consider of the anatomy and psychological properties of the HVS, based on which we can reconstruct the complete HVS. The metrics following this approach include the Visual Difference Predictor(VDP) [1] and Just Noticeable Difference (JND) model [2].

Another approach to reconstruct the HVS is replacing it with a visual information processing model, which could be in any form in mathematics. Based on the loss of image structural information, wang has proposed a series of image quality evaluation algorithms include Structural Similarity Index (SSIM) [4], the Multi-Scale SSIM (MS-SSIM) [5]. On the other hand, image quality evaluation algorithms Image Fidelity Criteria (IFC) [6] and Visual Image Fidelity (VIF) [7]are developed on the Natural Scenery Statistics (NSS) model.

Based on the VDP, the the early work of building the HVS model, we have proposed a novel image quality evaluation algorithm called Visual Difference based Quality Metric(VDQM), where a genetic algorithm based support vector machine, instead of traditionally linear or nonlinear mathematical models, is employed to describe the relationship between image similarity features and subjective image quality.

2 Related Works

This section briefly introduces the classical MSE metric , and a family of structural-similarity based image quality metrics, including UQI, SSIM and MS-SSIM. Among them, MSE and UQI simply measure the difference and similarity between image signals being compared. Based on the proper assumptions about the mechanism by which humans judge image quality, SSIM and MS-SSIM understand the biological HVS as a visual-information(signal)-processing system.

2.1 MSE

Despite the earliest known image quality metrics, MSE and PNSR (the Peak Signal Noise Ratio) are still widely used in many image processing application [3].

Give a distorted image $\hat{I}$ and its original form I in the size of $M \times N$, MSE is defined as follows:

$$\text{MSE} = \frac{1}{MN} \sum_{i=1,j=1}^{M,N} (I(i,j) - \hat{I}(i,j))^2 \tag{1}$$

$$\tag{2}$$

MSE is actually a special case of the Minkowski norm (distance): $E_p = \left(\sum_{i=1}^{N} |x_i - y_i|^p \right)^{1/p}, p \in [1, \infty]$, where $p = 1$ leads to the Mean Absolute Error (MAE) measure, $p = 2$ yields the square Root of MSE (RMSE), and $p = \infty$ gives the Maximum Absolute Difference (MAD) measure $E_\infty = \max_i |x_i - y_i|$.

Compared with other Minkowski norms, MSE is not only mathematically differentiable, but also has a clear physical meaning, i.e., the energy of the error signal. However MSE has long been criticized for its poor correlation with subjectively perceived image quality.

2.2 SSIM

The development of SSIM [4] follows a different philosophy than MSE. Based on an assumption that image quality deterioration is mainly cause by the loss of structural information, SSIM models the HVS using three parallel channels,which are in charge of processing luminance, contrast and structure information, respectively. Finally, SSIM is defined as follows:

$$\text{SSIM} = \sum_{i=1}^{N} [l(x_i, y_i)]^\alpha \cdot [c(x_i, y_i)]^\beta \cdot [s(x_i, y_i)]^\gamma, \tag{3}$$

where $l(x_i, y_i)$, $c(x_i, y_i)$ and $s(x_i, y_i)$ are result of comparing two image patches x_i and y_i in the luminance, contrast and structure channels.

2.3 MS-SSIM

Taking an input distorted image and its reference one, MS-SSIM [5] iteratively applies a low-pass filter and downsamples the filtered images by a factor of 2. The original image is indexed as 1 and the image at the highest scale M is obtained after $M - 1$ iterations.

At each scale (e.g., the jth scale), the luminance, contrast and structure comparison ($l_M(x_i, y_i)$, $c_j(x_i, y_i)$, $s_j(x_i, y_i)$) of two image patches x_i and y_i is computed in a similar way to SSIM. The overall MS-SSIM evaluation is obtained by combining comparison results at different scales together

$$\text{MS-SSIM} = \sum_{i=1}^{N} [l_M(x_i, y_i)]^{\alpha_M} \cdot \prod_{i=1}^{M} [c_j(x_i, y_i)]^{\beta_j} \cdot [s_j(x_i, y_i)]^{\gamma_j} \tag{4}$$

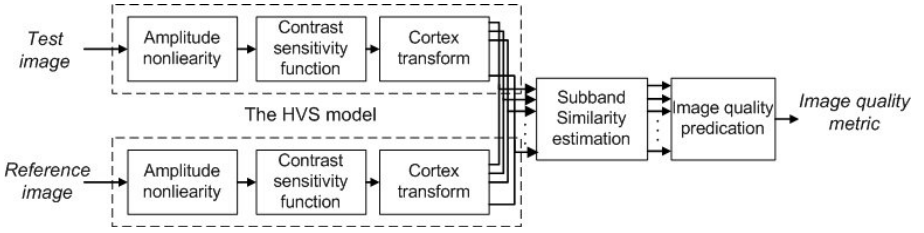


Fig. 1. The framework of our proposed method

3 The Proposed Method

In this section, a novel image quality evaluation algorithm VDQM is developed based on the VDP model for estimating the visual similarity between two images (as shown in Fig. 1). The whole algorithm is composed of three step: the HVS model, subband similarity estimation, image quality predication. Their functions are described in detail:

- The HVS model represents the low-order processing of our visual system. As shown in Fig.2, an input image pass through a series of processes : amplitude nonlinearity, contrast sensitivity function, cortex transform representing the response of the HVS to light level, spatial frequency and content of images, and are decomposed into 30 high-frequency subbands and 1 low-frequency.
- After these 31 subbands are qutiflyed into the range of [0 255], their similarities between reference and test images are computed using 12 image quality metrics MSE, PSNR, VSNR, SSIM, MS-SSIM, VIF, VIFP, UQI, IFC, NQM, WSNR and SNR. Therefore there are a set of 372(31x12) features extracted for each test image.
- Image quality is predicted by the support vector machine(SVM) from image similarity features, i.e, subband similarity measures, where the genetic algorithm (GA) is in charge of selecting the proper feature subset and parameters for SVM. Finally, the chosen features and parameters is presented in Table1 .

4 Experiments and Results

The performance test method is presented here: Instead of directly measuring the agreement between image quality measures and subjective ratings (Mean Opinion Score, MOS) , a nonlinear transformation process in form of $\mathrm{PMOS}(x) = \frac{\beta_1 e^{\beta_2(x-\beta_3)}}{2\left(1+e^{\beta_2(x-\beta_3)}\right)} + \beta_4 x + \beta_5$ is introduced to transform video quality metric values into Predicted MOS (PMOS), which are then compared with MOS using statistical measures, Correlation Coefficient (CC), Spearman Ranking Order Correlation Coefficient (SROCC), RMSE, Maximal Absolute Error (MAE). By providing a more intuitive impression of the performance improvement of one image quality metric over another, CC and RMSE are more widely used in the community of image (video) quality assessment.

Following the method described above: both our proposed method and the state-of-the art image quality evluation algorithm MSE, SSIM and MS-SSIM are testified

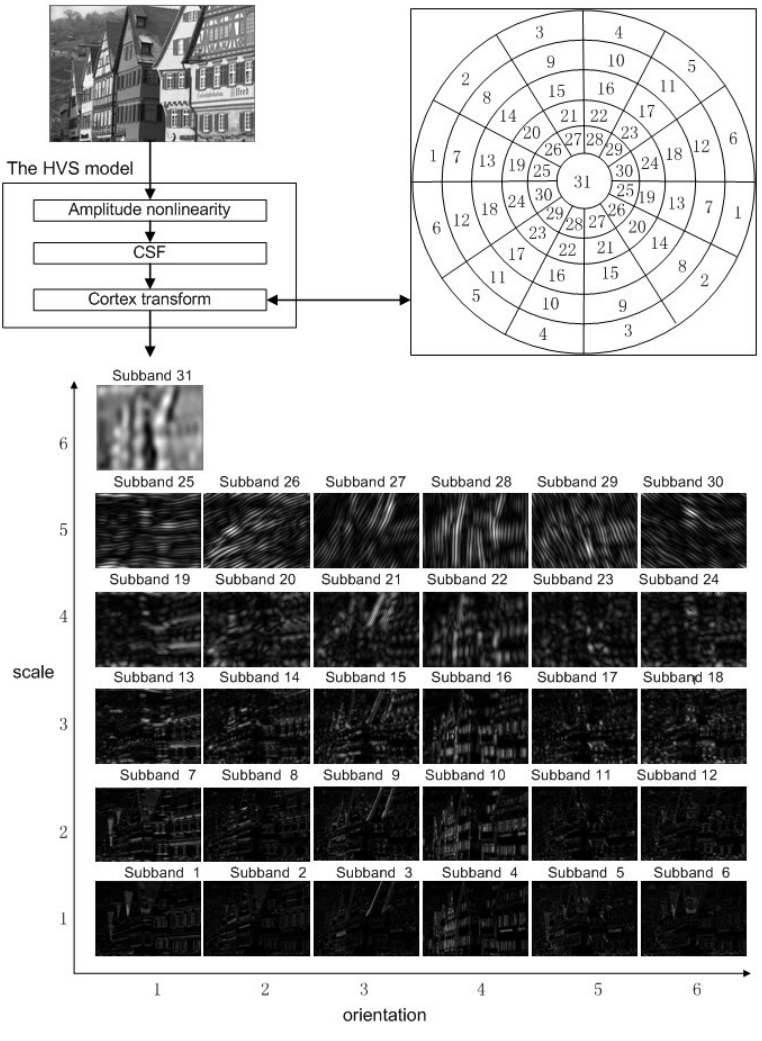


Fig. 2. The decomposition of "building" image into 31 subbands

on the LIVE image database [8] developed by Laboratory for Image and Video Engineering (LIVE) of Texas university and A57 image database developed [9] by Visual Communications Lab of Cornell University. All these image databases consist of three parts: original image with perfect quality, distorted image and subjective image quality ratings (Mean Opinion scores).

Table 2 presents the performance comparison of our proposed metric VDQM and the state-of-the-art image quality metrics. It is observed that VDQM performs much better than its counterparts (MSE, SSIM and IFC) with respect to each of the statistical measure: CC, SROCC, RMSE and MAE, on the LIVE and A57 image database. That

Table 1. The chosen features and parameters for image quality evaluation

Chosen Features					
index	subband number	Similarity computation	index	subband number	Similarity computation
1	1	PSNR	15	16	SSIM
2	2	SSIM	16	17	PSNR
3	3	PSNR	17	18	SSIM
4	4	MSE	18	19	PSNR
5	5	PSNR	19	20	MSE
6	6	SSIM	20	20	SSIM
7	7	PSNR	21	22	SSIM
8	11	PSNR	22	26	MSE
9	12	MSE	23	28	MSE
10	12	SSIM	24	29	PSNR
11	14	MSE	25	30	MSE
12	14	SSIM	SVM parameters		
13	15	PSNR	cost	gamma	epsilon
14	16	MSE	64	4.66×10^{-10}	2

Table 2. Statistical measures between PMOS obtained by image quality evaluation algorithms and real MOS

	LIVE				A57			
Measures	MSE	SSIM	MS-SSIM	VDQM	MSE	SSIM	MS-SSIM	VDQM
CC	82.86%	87.69 %	90.56%	96.03 %	77.96%	81.77%	89.12%	91.12%
SROCC	81.97%	87.63 %	89.76%	95.87 %	76.81%	80.51%	88.93%	90.96%
RMSE	9.06	7.74	6.83	4.36	0.5	0.3	0.21	0.12
MAE	7.18	5.87	5.19	3.72	0.42	0.25	0.17	0.10

proved our approach to image quality evaluation based on the incorporation of the HVS model and machine learning techniques is more successful than the traditional one.

5 Conclusion

In this paper, a novel image quality evaluation algorithm VDQM is proposed based on the Visual Difference Predictor(VDP), the the early work of building the HVS model. a genetic algorithm based support vector machine, instead of traditionally linear or non-linear mathematical models, is employed to describe the relationship between image similarity features and subjective image quality. Experiments show that VDQM performs much better than its counterparts (MSE, SSIM and IFC) on both the LIVE and the A57 image databases. In the future, we will consider of (1) extending the HVS model for gray-level images into a HVS model for full-color images and (2) employing more image similarity features, such as texture similarity, color difference equations and models.

References

1. Daly, S.:The visual difference predictor:an algorithm for the assessment of image fidelity. In: Digital Image and Human Vision, Cambridge, USA, pp 179–206 (1993)
2. The JND technology, Sarnoff Corporation, http://www.sarnoff.com
3. Wang, Z., Bovik, A.C.: Modern image quality assessment. Morgan & Claypool (2007)
4. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. IEEE Trans Image Processing 13(4), 600–612 (2004)
5. Wang, Z., Bovik, A.C., Simoncelli, E.P.: Multi-scale structural similarity for image quality assessment. In: IEEE Conference on Signals, Systems and Computers (2003)
6. Sheikh, H.R., Bovik, A.C., Veciana, G.D.: An information fidelity criterion for image quality assessment using natural scene statistics. IEEE Trans. on Image Processing 14(12), 2117–2128 (2005)
7. Sheikh, H.R., Bovik, A.C.: Visual image information and visual qulity. IEEE Trans. on Image Processing. 15(2), 430–444 (2006)
8. Cormack, L., Sheikh, H.R., Wang, Z.: LIVE image quality assessment database release 2, http://live.ece.utexas.edu/research/Quality/index.htm
9. Chandler, D.M., Hemami, S.S.: A57 image quality assessment database release, http://foulard.ece.cornell.edu/dmc27/vsnr/vsnr.html

A Gabor Wavelet Pyramid-Based Object Detection Algorithm

Yasuomi D. Sato[1,2], Jenia Jitsev[2,3], Joerg Bornschein[2], Daniela Pamplona[2], Christian Keck[2], and Christoph von der Malsburg[2]

[1] Department of Brain Science and Engineering, Graduate School of Life Science and Systems Engineering, Kyushu Institute of Technology,
2-4, Hibikino, Wakamatsu-ku, Kitakyushu, 808-0196, Japan
sato-y@brain.kyutech.ac.jp
[2]Frankfurt Institute for Advanced Studies (FIAS), Johann Wolfgang Goethe-University,
Ruth-Moufang-Str. 1, 60438 Frankfurt am Main, Germany
{sato,jitsev,bornschein,pamplona,keck,
malsburg}@fias.uni-frankfurt.de
[3] Max-Planck-Institute for Neurological Research, Gleueler Str. 50, 50931, Koeln, Germany
jenia.jitsev@nf.mpg.de

Abstract. We introduce visual object detection architecture, making full use of technical merits of so-called multi-scale feature correspondence in the neurally inspired Gabor pyramid. The remarkable property of the multi-scale Gabor feature correspondence is found with scale-space approaches, which an original image Gabor-filtered with the individual frequency levels is approximated to the correspondingly sub-sampled image smoothed with the low-pass filter. The multi-scale feature correspondence is used for effectively reducing computational costs in filtering. In particular, we show that the multi-scale Gabor feature correspondence play an effective role in matching between an input image and the model representation for object detection.

Keywords: Gabor Pyramid, Visual Object Detection, Multi-scale Feature Correspondence, Computer Vision.

1 Introduction

Object detection in real time is one of requisite processes in visual object recognition. In this work, we propose a so-called Gabor pyramid of the multi-scale feature correspondence finding for visual object detection as shown in Fig. 1. This is based on the modeling of receptive fields that are localized or decomposed in the Gabor spatial frequency domain (physiologically plausible Gabor decomposition). It has been hypothesized that the receptive fields in mammalian visual systems closely resemble Gabor kernels and has confirmed by a number of physiological studies on cats and primates [1].

In this Gabor pyramid object detection system, an input (I) image is down-sampled at an arbitrary scale, which may be related to the spatial frequency levels of the Gabor wavelets. Although the search region for a model (M) object

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 232–240, 2011.

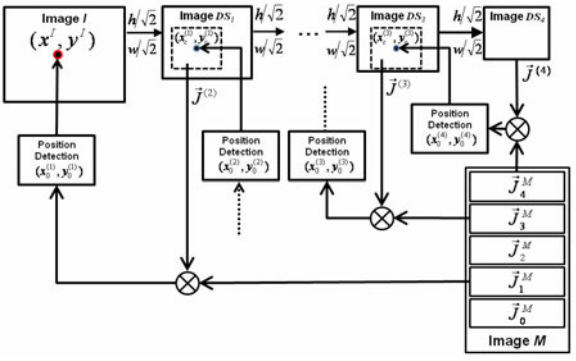


Fig. 1. Sketch of the whole Gabor pyramid algorithm for visual object detection. (x, y) represents the position of a single pixel within the image. $\boldsymbol{J}$ is the Gabor feature extracted from the image. $\otimes$ denotes the inner product between Gabor features for a model (M) image and the down-sampled (DS) image. Such mathematical symbols are thoroughly described in Sect. 2.

already stored in the system is localized in each down sampled image (broken squares in Fig. 1), the localization is carried out by finding the Gabor feature correspondence to the M feature. It is allowed to gradually specify the most likely position of the M object in each image, in which is analogous to flow from low to high resolution. Finally, an accurate position for the M object on the image I with highest resolution can be detected.

Correspondence finding between the Gabor filter for the image M and the low-pass Gabor filter for the down-sampled version of the I is the important aspect of the Gabor pyramid algorithm. The identification of feature correspondences enables to effectively find a search region to specify the M object even at a lower resolution than that of the M image. When the image resolution is increased, the search region gradually converges to instead detect the most likely position. This is analogous to a coarse-to-fine template matching method in pattern recognition studies, which is a potentially useful method that makes cost performance much lower [2]. However, no one can know the coarse-to-fine matching by finding the aforementioned Gabor feature correspondence.

In addition, physiological plausible Gabor decompositions present another significantly crucial advantage in the Gabor pyramid as it enables us to realize low computational cost with fewer Gabor filters on the limited image space without the loss of any physiological constraints. Conventionally in the correspondence-base visual object recognition model of dynamic link matching [3], 40 Gabor filters have to be used to computationally establish the recognition, putting a heavy burden on the performance of the software system. The same performance cost problem occurs even in the feature-based model of the Hubel-Wiesel type [4][5]. Thus, there is still currently a great deal of discussion regarding how to best deal with Gabor filters and the relevant performance cost problem.

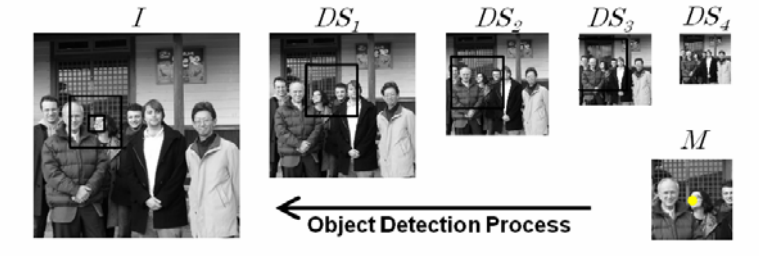


Fig. 2. Face detection process after down sampling an input (*I*) image. In each down sampled (*DS*) image, 64×64 pixel size windows (for example, solid squares in the DS_1, DS_2 and DS_3) are set up to search the most likely pixel position of the model (*M*) face. A filled circle is an extraction point for the *M* face. After the process, our Gabor pyramid system can detect the *M* face on the image *I* with a small frame of 20×20 pixel size.

In this work, with the full use of the physiologically plausible Gabor decomposition and scale correspondence finding between multi-resolution and Gabor feature, we attempt to develop the Gabor pyramid algorithm that model object images stored in the system can effectively and rapidly be detected on an input image. We also show that the Gabor pyramid technically supports the functionality of the coarse-to-fine template matching. This artificial vision has significant potential for practical applications, preserving the physiological nature of the Gabor filter. In Sect. 2, an object detection mechanism of the Gabor pyramid is explained in detail. In Sect. 3 and 4, numerical results of feature correspondence, multi-object as well as multi-face detection are given. In the final section, results will be discussed and conclusions given.

2 A Gabor Pyramid System

An outline of the Gabor pyramid system proposed here is shown in Fig. 1. We assume that a grey-scale natural input (*I*) image of multi people is first prepared with $w \times h$ pixel (*w* is the width while *h* is the height). The image *I* is down-sampled using the $[1/2]^l$ (l=0,…,4), and then is stored as an image DS_l in the system. Here the l=0 case represents the original size of the image *I*.

Let the image *M* with 100×100 pixel be cut out from the image *I*. There has to appear one single object centered in the image *M*. One single feature $\boldsymbol{J}_{l'}^{M}$ $=\{ J_{l',r}^{M} \}_{r=0,1\ldots,7}$ for l' th spatial frequency (where *r* represents orientation components, and l'=0,…,4) is extracted at a center of the image *M*, which is defined as the convolution of the image with a set of Gabor wavelet transformations. The Gabor filter responses *J* are usually given by:

$$\hat{J} = \int I(z - z')\psi(z - z')d^2z', \tag{1}$$

$$\psi(z) = \frac{k^2}{\sigma^2}\exp\left(-\frac{k^2 z^2}{2\sigma^2}\right)\left[\exp(ikz) - \exp\left(-\frac{\sigma^2}{2}\right)\right] \tag{2}$$

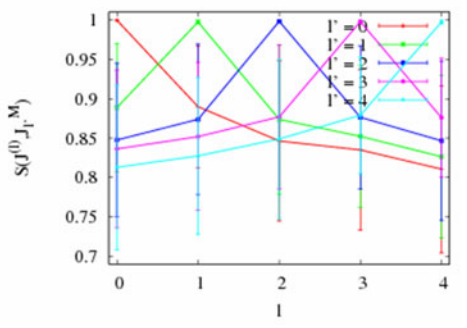


Fig. 3. Scale correspondence of the Model (*M*) feature to the one corresponding to the low-passed version features. In this figure, the feature similarity for *l'* th spatial frequency takes the function of a scaled down index *l*, plotting an average value and the SD of the similarity, calculated with 100 different sample image.

where σ=2π to approximate the shape of receptive fields observed in the primary visual cortex. The wave vector is parameterized as

$$\vec{k} = \begin{pmatrix} k_x \\ k_y \end{pmatrix} = \begin{pmatrix} k_l \cos\varphi_r \\ k_l \sin\varphi_r \end{pmatrix}, \quad k_l = 2^{-\frac{l+2}{2}}\pi, \quad \varphi = \frac{\pi}{8}r, \tag{3}$$

with the orientation parameter $r = 0, \ldots, 7$ and the scale parameter $l = 0, \ldots 4$. As feature values we use the magnitude

$$J = \left|\hat{J}(z)\right|. \tag{4}$$

In each image DS_l, the 64×64 pixel size of the Region-of-interest (ROI_l) is extracted, which $(x_c^{(l)}, y_c^{(l)})$ is located as a center of the ROI_l and is matched to the model Gabor feature $\boldsymbol{J}_l^M$. In the ROI_l, the Gabor features $\boldsymbol{J}^{(l)}(x^{(l)}, y^{(l)}) = \{ J^{(l)}{}_r(x^{(l)}, y^{(l)}) \}_{r=0,1,\ldots,7}$ are extracted for each $(x^{(l)}, y^{(l)})$ in order to calculate similarities to the relevant model feature, $S(\boldsymbol{J}^{(l)}(x^{(l)}, y^{(l)}), \boldsymbol{J}_l^M)$, which is given by

$$S(\vec{J}^{(l)}(x^{(l)}, y^{(l)}), \vec{J}_l^M) = \frac{\sum_r J_r^{(l)}(x^{(l)}, y^{(l)}) \cdot J_{l,r}^M}{\sqrt{\sum_r \left(J_r^{(l)}(x^{(l)}, y^{(l)})\right)^2 \sum_r \left(J_{l,r}^M\right)^2}} \tag{5}$$

We then choose the candidate point $(x_o^{(l)}, y_o^{(l)})$ by computing the highest value of the pixel location-specific similarity:

$$(x_0^{(l)}, y_0^{(l)}) = \underset{x^{(l)}, y^{(l)}}{\arg\max} \left\{ S(J^{(l)}(x^{(l)}, y^{(l)}), J_l^M) \right\}, \tag{6}$$

and span the new search region of the defined 64×64 pixel size around the normalized pixel location $(x_c^{(l-1)}, y_c^{(l-1)})$ on the next up-sampled level:

$$(x_c^{(l-1)}, y_c^{(l-1)}) = \sqrt{2}(x_0^{(l)}, y_0^{(l)}) . \tag{7}$$

Repetitively doing such position specific process for each down-sample image, an exact position (x_o^I, y_o^I) for specifying the model object is finally decided on an original image I with the highest resolution (see, a small square with 20×20 pixel size in the I image of Fig. 2). This object detection, carried out on a laptop computer (Intel Core(TM)2 Duo CPU 1.40GHz RAM 1.91GHz) in this case, is demonstrated as shown in Fig. 2. Here we note that the fixed search window seems to gradually converging to the desired model object from low resolution to high resolution as shown in Fig. 2. The runtime in the object detection process was less than 500 msec.

3 Scale Feature Correspondence to Image Resolution

In this work, a substantial reason why our Gabor pyramid algorithm can effectively detect not only faces, but also general objects, is to find feature correspondence between down-sample image resolutions and spatial frequency factors of the Gabor feature. We here confirm such feature correspondence findings, by using 100 different images of a single person i (i=0, …, 99). Each image is used both as the input and the model, which are respectively called M_i and I_i.

From the center of the image M_i, the one scale feature vector which consists of 8 orientation components is extracted for a scale factor l' (l'=0, …, 4). On the other hand, the image I_i is down-sampled with $(2)^{-l/2}h_i$ and $(2)^{-l/2}w_i$ (l=0, …, 4), which we will refer to as DS_l^i. From the center of each image, one feature vector with the same number of orientation components as the M_i case is obtained by filtering with a standard spatial frequency of the Gabor wavelets. Then, the Gabor feature for l' of the M_i takes inner-product with the Gabor feature of the DS_l^i to calculate their feature similarity:

$$S(J^{(l)}, J_{l'}^M) = \frac{\sum_r J_r^{(l)} \cdot J_{r,l'}^M}{\sqrt{\sum_r \left(J_r^{(l)}\right)^2 \sum_r \left(J_{r,l'}^M\right)^2}} . \tag{8}$$

The feature similarities for each spatial frequency l' are obtained as shown in Figure 3. In this figure, all values of the feature similarities are respectively averaged over the sampling number of the facial image, calculating the standard deviation (SD) of the feature similarity values. We have thus calculated tuning curves for each spatial frequency of the M feature. As shown in Fig. 3, the low-pass Gabor filter for the image DS_l best-matches to the model Gabor filter with the same spatial frequency factor, but it has obtained an incorrect scale correspondence to another DS image.

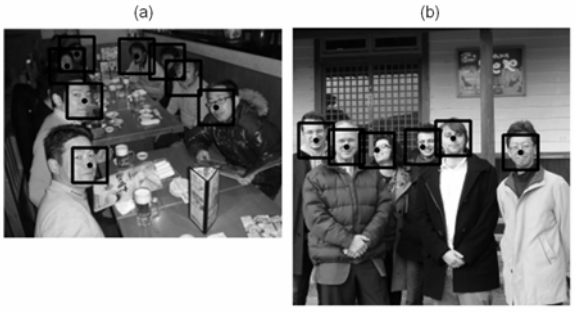


Fig. 4. Multi-face detection results. Solid squares on input images are detection windows. Filled circles are real positions that successfully achieved correct detection of features stored in memory.

Sato et al. [6][7] suggested the sue of scale-correspondence finding between feature components of the input and model. For example, the components of the Gabor feature vector are set up as the angular-eccentricity coordinate. If an input object is scaled down relative to the original size (in other words, having a lower resolution than the original), correspondingly, the components are radically shifted towards an outside of the coordinate. This addresses the scale correspondence of the down-sample image resolution to the Gabor kernel size filtered on the image. In order to support this address, we have shown the scale correspondence in Fig. 3.

4 Simulation Results

Finally, we demonstrate two detection experiments for a number of general objects and faces. In fact, as shown in Fig. 4(a), even though the face size is so small that its appearance has become blurred, and another face is partially occluded, the Gabor pyramid can easily detect these objects. This can be achieved due to single Gabor feature extraction from a fiducial point (that is, the tip of the nose).

At present, we notice that the detection window is fixed. Consequently, when the face size is big, the corresponding detection window is set up on a small portion of the face. When the face size is small, (i.e. smaller than 20×20 pixels), the face is positioned within the window. If this Gabor pyramid can be improved to specify the most likely size of the detected face, it may be able to automatically modify the detection window according to the specified size. In such improvement, we need other methods to specify the size, one of which scale and rotation transformation specific similarity computations proposed by [11] would be appropriate since it is a powerful method for scale and rotation invariant object recognition. When the scale and rotation transformation specific similarity computations are integrated within the framework of our Gabor pyramid with a functionality of translation invariance, we can say this improved Gabor pyramid system is recognition fully invariant to changes of scale, rotation and translation.

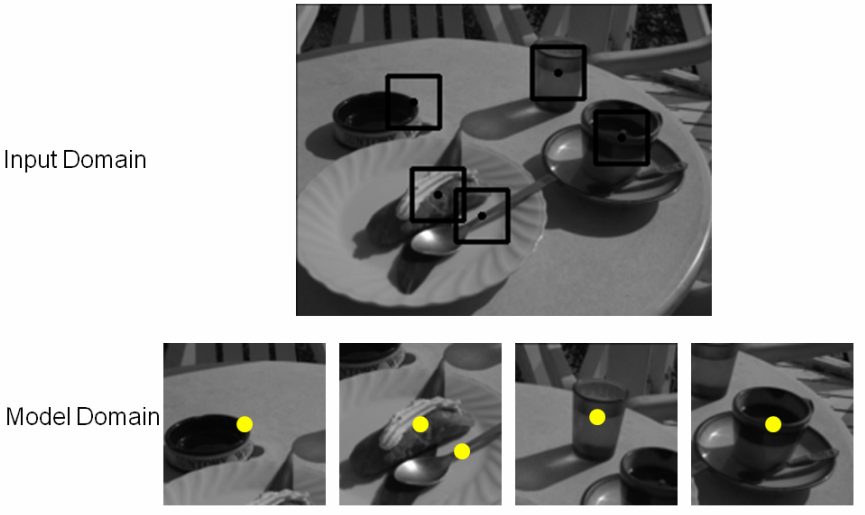


Fig. 5. Detection results for an image containing multiple objects (an ashtray, a cake, a spoon, a glass and a coffee cup). Solid squares centered on the filled black circle in the upper figure denote that a part of each general object, which is stored as one of the model feature representations (yellow circles), which can successfully be detected.

However, we have to be aware of some detection ability problems in matching by finding only single feature correspondence. This is shown in Fig. 4(b). The tip of the nose could not be specified for two of the five faces in the natural image even if the model Gabor feature is extracted from such a fiducial point on the face. In the cases, the face must be detected using another fiducial point such as the mouth. This result implies that the system should store not only a single but multiple features in memory to yield a better performance in the face detection.

Next, we test the detection ability for a number of general objects, (in Figure 5, an ashtray, a glass, a coffee cup, a cake and a spoon). In this test, the single Gabor feature is effectively extracted from the related object's contour. Then, a detection window of the Gabor pyramid system fills a segment of the object. As mentioned above, such detection may be interpreted as being achieved by selecting one of several stored features forming the whole object representation, which must correctly detect the position on an input image. Thus, by proposing visual object detection with the Gabor pyramid, we may suggest a possible model of visual object recognition.

5 Discussion and Conclusion

This work is an important and preliminary step toward practical applications of image processing and object recognition. As a next step, we attempt to establish a visual object recognition system that is fully invariant to scale, rotation, as well as translation, by integrating another detection algorithm for the most likely scale and rotation transformational states of the object into the Gabor pyramid algorithm in this work. This scale and rotation transformation detection was already proposed [7].

In general, the face detection system proposed by Viola and Jones [8] is often used in research fields of computer vision and image processing. It is well-known that this face detection does not work when a person turns their head to one side, or the facial resolution is too low[9]. However such a detection problem is expected to be solved in case of an improved version of the Gabor pyramid in which another invariant recognition mechanism is integrated. Consequently, the construction of such an integrated visual object recognition system is an urgent task.

We are planning on implementing the Gabor pyramid algorithm into FPGA. By this implementation, we will expect much faster speed of the Gabor pyramid performance with larger numbers of face or objects. Our Gabor pyramid could detect a coupled of faces/general objects in 1 [sec] without parallel processing. However, the implementation of our Gabor Pyramid into FPGA will be allowed to, in real time, process detection of further more faces or objects.

Several features extracted from fiducial points on the face or contours of the object, such as graphs, are often used a correspondence-based recognition model, and are necessary to achieve smooth visual object detection. This because simulation results in this work have indicated that there are still some difficulties associated with the detection process that uses only single feature, one of which is shown in Fig. 4(b). In order to overcome such difficulties, topological constraints such as facial graph consisting of several Gabor features is required.

In conclusion, there still great deal of work to be done in the construction of a neurally plausible object recognition model. However, we must also stress that the work described here is in the fundamental stage with regard to practical applications. In order for these applications to be successful, the method must demonstrate the flexibility and universality of the underlying concept of the Gabor pyramid processing. One of the most crucial mechanisms is correspondence finding between images of different resolutions and spatial frequencies of the Gabor filter. These are found due to the physiological plausible Gabor decomposition and have a great deal of potential to solve the computer vision problem. It also introduces the possibility of recycling view-dependent information about the initially unknown size of an object, which may have been regarded as the unnecessary. The similarity computation pursued here will contribution substantially to the further understanding of the highly integrated visual recognition mechanism behind the invariance.

Acknowledgments. This work was financially supported by the “Bernstein Focus: Neurotechnology through research grant 01GQ0840” funded by the German Federal Ministry of Education and Research (BMBF). Y.D.S was supported by the Grant-in-Aid for Young Scientist (B) No. 22700237.

References

1. Jones, J.P., Palmer, L.A.: An evaluation of the two-dimensional Gabor filter model of simple receptive fields in the cat striate cortex. Journal of Neurophysiology 58(6), 1233–1258 (1987)
2. Dufour, R.M., Miller, E.L.: Template Matching Based Object Recognition With Unknown Geometric Parameters. IEEE Transactions on Image Processing 11(12), 1385–1396 (2002)

3. Lades, M., Vorbrueggen, J.C., Buhmann, J., et al.: Distortion invariant object recognition in the dynamic link architecture. IEEE Transactions on Computers 42(3), 300–311 (1993)
4. Serre, T., Wolf, S., Bileschi, M., Riesenhuber, P.T.: Robust Object Recognition with Cortex-like Mechanisms. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(3), 411–426 (2007)
5. Hubel, D.H., Wiesel, T.N.: Receptive Fields, binocular interaction and functional architecture in the cat's visual cortex. Journal of Physiology 160(1), 106–154 (1962)
6. Sato, Y.D., Wolff, C., Wolfrum, P., von der Malsburg, C.: Dynamic Link Matching between Feature Columns for Different Scale and Orientation. In: Ishikawa, M., Doya, K., Miyamoto, H., Yamakawa, T. (eds.) ICONIP 2007, Part I. LNCS, vol. 4984, pp. 385–394. Springer, Heidelberg (2008)
7. Sato, Y.D., Jitsev, J., von der Malsburg, C.: A Visual Object Recognition System Invariant to Scale and Rotation (The ICANN 2008 Special Issue). Neural Network World 19(5), 529–544 (2009)
8. Viola, P., Jones, M.J.: Robust Real-Time Face Detection. International Journal of Computer Vision 57(2), 137–154 (2004)
9. Hayashi, S., Hasegawa, O.: Robust Face Detection for Low-Resolution Images. Journal of Advanced Computational Intelligence and Intelligent Informatics 10(1), 93–101 (2006)

Quality Evaluation of Digital Image Watermarking

Xinhong Zhang[1,2], Fan Zhang[2,3], and Yuli Xu[3]

[1] Computing Center, Henan University, Kaifeng 475001, China
zxh@henu.edu.cn
[2] Institute of Image Processing and Pattern Recognition, Henan University, Kaifeng 475001, China
[3] College of Computer and Information Engineering, Henan University, Kaifeng 475001, China
zhangfan@henu.edu.cn

Abstract. Many kinds of distortion take place during the digital image acquisition, processing, compression, storage, transmission and copy. All of these distortions would lead to a decline in visual quality of the image. Therefore, the image quality assessment is very important for us. Digital watermark is an important application of image processing. Different applications require different watermark techniques, and the different watermark techniques require different evaluation criteria. Currently, there have not a complete evaluation system about digital watermark. Because the uniform description of the performance, the test methods, the method of attack, the standard test procedure have not been established. How to judge the performance of a watermarking system is very important. In this paper, the evaluation of a watermarking system is associated to the watermark robustness, security, capacity and invisibility.

Keywords: Image quality evaluation, digital watermark, subjective evaluation, objective evaluation.

1 Introduction

Image quality evaluation method can be divided into the subjective evaluation and objective evaluation methods. The former is observed directly by setting the evaluation criteria and evaluation standards, human visual perception as well as, then give grade and evaluation. In accordance with the image of the statistical average grade to give a final assessment of the results by observer. There are two kinds subjective evaluation that is absolute and relative dimension, as shown in Table 1. This method of measurement reflects the quality of the visual images, but it can not be applied and described in mathematical models. From the view of engineering point, it is too time-consuming and laborious. In practice, the subjective evaluation method of image quality has been limited severely. Even it is suitable for certain applications, such as real-time video transmission, and other fields [1,2].

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 241–250, 2011.

Image objective evaluation method used mathematical model and computed similarity between the image distortion and the original image (or distortion) and quantized evaluation scores. Objective image quality evaluation method based on the dependence of the original image can be divided: full reference, reduced reference and no reference [3]. In all kinds of full reference objective image quality evaluation method, the MSE (mean squared error) and peak signal to noise ratio (PSNR) is widely used. The MSE is,

$$MSE = \frac{\sum_{0 \le i \le M} \sum_{0 \le j \le N} (f_{ij} - f'_{ij})^2}{M \times N}, \tag{1}$$

where f_{ij} and f'_{ij} denote the original image and restored image respectively, M, N are the high and the wide of images respectively. PSNR is the same as MSE in essence. Its expression as follows:

$$PSNR = 10 \log_{10} \frac{256 \times 256}{MSE}. \tag{2}$$

(1) and (2) looks intuitively and strictly, but the results they get are inconsistent with the visuals effect of human subjective. Because the mean square error and peak signal to noise ratio reflect the difference of the original image and restore the image in whole, not reflect in the local of images. As a greater point of difference between gray and more like a small point there are and other such a situation. It is clear that all pixels of image are on the same status, they can not reflect the human visual characteristics.

Table 1. The measure levels of subjective quality evaluation

level	Absolute measurement scale	Relative measurement scale
1	Best	The best in a crowd
2	Better	It is better than average in a crowd
3	Normal	It is average in a crowd
4	Worse	It is worse than average in a crowd
5	Worst	The worst in a crowd

There are a variety of evaluation methods based on the above principle, such as: assessing the quality of enhanced images based on human visual perception, image quality assessing model by using neural network and support vector machine, gradient information based image quality assessment, image quality assessment method based on contrast sensitivity, and so on. In next sections, we introduce some typical image quality evaluation methods.

2 Image Quality Evaluation Methods

Wang xiang-hui *et al.* [4] propose an approach that considers the background of the average brightness in local and space complexity in visual resolution of the

impact, which is used to judge the local gray-scale's leap whether is an important perception of parameter just noticeable difference (JND). And use this parameter calculates effective pixels perception of change respectively, which is given to the quantitative evaluation of the results of image enhancement. Firstly, the image is divided into two regions: detail area and smoothness area, then study the detail area of information enhancement and the smoothness area of noise reduction respectively, according to two indicators of image enhancement effect, then give an objective evaluation of the result.

When use only structure similarity (SSIM, Structure Similarity) method, the association diagram still exists a number of isolated points on the subjective and objective evaluation of the quality, (so-called "isolated point" which is a sample of subjective evaluation and objective evaluation of the value of the difference is larger in image quality evaluation model of the relationship between subjective and objective evaluation diagram) these isolated points reduce to quality evaluation of the accuracy. This method will adopt the value of SSM which regards as a parameter of the image quality is described. According to PSNR and the isolated point of issue is considered. Reference [5] adopts an objective evaluation of image quality of the method which based on neural network and support vector machines.

As the human eye has an extremely high sensitivity on the edge of the image texture information, gradient can better respond on the edge of the image texture information, so Yang chun-ling *et al.* [6] propose a method which improves the structure of the similarity evaluation method based on gradient information.

We can compute every image pixel of the gradient amplitude by Sobel operator and gradient amplitude of the image pixels, thus we can get the "gradient" image X' and Y' that relative to image X and Y, and x' and y' is defined as the corresponding sub-block of the image X' and Y'. So sub-block gradient contrast comparison can be defined as:

$$C_g(x,y) = \frac{2\sigma_{x'}\sigma_{y'} + C_2}{\sigma_{x'}^2 + \sigma_{y'}^2 + C_2}. \tag{3}$$

Sub-block gradient of the correlation coefficient can be defined as:

$$S_g(x,y) = \frac{(\sigma_{x'y'} + C_3)}{\sigma'_x\sigma'_y + C_3}, \tag{4}$$

where $\sigma_{x'}$ and $\sigma_{y'}$ represents x' and y' of the standard deviation respectively, thus $\sigma_{x'y'}$ indicates x' and y' of the covariance, adding constant C_2, C_3 to avoid the denominator is zero. Then, use the formula (5), (6) substitute the second part $c(x,y)$ and the third part $s(x,y)$ of SSIM model; therefore obtain improving the model method as follow:

$$GSSIM(x,y) = [l(x,y)]^{\alpha} \cdot [C_g(x,y)]^{\beta} \cdot [S_g(x,y)]^{\gamma}. \tag{5}$$

In accordance with the same method, according to the whole image of the similarity comparison, we can obtain the similarity score by averaging the various sub-blocks,

$$MGSSIM(X,Y) = \frac{1}{M}\sum_{j=1}^{M} GSSIM(x_j, y_j). \tag{6}$$

For each block of image, we can compute each pixel of gradient amplitude and direction of the gradient in sub-block, usually direction of the gradient can be quantified for 8 discrete values. Then information vector of the sub-block image is defined:

$$D = \{Amp_1, Amp_2, Amp_i, \cdots, Amp_8\}(i = 1, 2, \cdots, 8),$$

where Amp_i is sum which is all the pixels of gradient amplitude and the direction of gradient is i in the sub-block.

Finally, D_x and D_x represents the original image and degraded image corresponds to block information on the edge of Vector respectively. The similarity of the two sub-blocks is described:

$$s_e(x,y) = \frac{\sigma''_{xy} + C_3}{\sigma''_x \sigma''_y + C_3}, \tag{7}$$

where σ''_x and σ''_y represents D_x and D_y of the standard deviation respectively, and σ''_{xy} indicates D_x and D_y of the covariance, adding constant C_3 to avoid the denominator is zero. Then, use the formula (8) substitute the third part $s(x,y)$ of SSIM model; therefore obtain improving the model method as follow:

$$ESSIM(x,y) = [l(x,y)]^{\alpha} \cdot [c(x,y)]^{\beta} \cdot [s_e(x,y)]^{\gamma}. \tag{8}$$

In accordance with the same method, according to the whole image of the similarity comparison, we can obtain the similarity score by averaging the various sub-block,

$$MESSIM(X,Y) = \frac{1}{M}\sum_{j=1}^{M} ESSIM(x_j, y_j). \tag{9}$$

Wavelet-based contour-let transform (WBCT), a non-redundant sparse representation of image [3,11], can effectively reflect visual characteristics of the image, such that it is often used to capture the variation of visual perception to result from image distortion. The method is as follow: on the sending terminal, firstly, the reference image is decomposed by WBCT, and the reference image decomposes different scale and direction of the sub-band. Secondly, each sub-band carries out Contrast sensitivity (CSF) mask by theirs scale in order to make a different scale of the Coefficient in human perception has the same sense in all of the scale. Then the human visual perception characteristics determine a reasonable perception threshold, the separate statistics visual perception coefficient of the proportion which occupies in the various sub-bands. On receiving terminal, carries on similar processing to the distortion of images. Finally, we can obtain a comprehensive objective evaluation of image quality by comparing with reference images and distortion image characteristics information.

Wang zheng *et al.* [9] presents new methods which based on contrast sensitivity and integrate with HVS. According to the contrast sensitivity of HVS characteristics, namely spatial frequency-characteristic curve, 2D multi-level wavelet decomposition was applied to the test image for obtaining wavelet coefficients. From these coefficients, brightness, clarity and the relevant indicator in sub-bands were obtained. Then an arithmetical mean, which came from the geometric mean of these three measures multiplying sub-band weighted coefficients, was used as final comprehensive assessment indicator.

3 Quality Evaluation of Digital Watermark

Digital watermark is an important application of image processing. Different applications require different watermark techniques, and the different watermark techniques require different evaluation criteria. Currently, there have not a complete evaluation system about digital watermark. Because the uniform description of the performance, the test methods, the method of attack, the standard test procedure have not been established. How to judge the performance of a watermarking system is very important.

A user that utilisers a watermarking algorithm to embed an invisible watermark in his/her data (still image/video sequence) is concerned with two kinds of visual quality, namely the visual quality of the data due to the embedding of the watermark and the visual quality of the watermarked data due to attacks performed on it. These terms will be called VQ1 and VQ2, respectively. The following block diagram (Fig. 1) further explains what are the meanings of the terms.

Measuring the visual quality of a watermarking system concerns two questions. The first question is whether or not the watermarked data is perceptually different from the host data. The referenced host data can be data at studio quality level or data at consumer quality level. The second question is whether or not the received data, i.e. the data obtained after common processing or initial

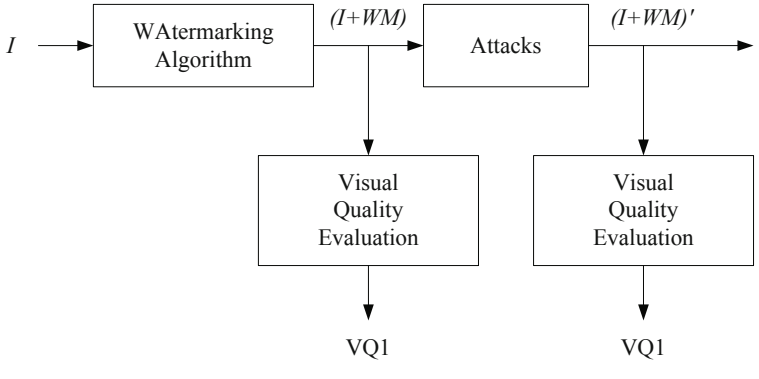


Fig. 1. The visual quality of the watermarked image

attacking, is perceptually different from the host data, in particular for received data from which the original message can no longer be retrieved.

The most well-known and widely used quality measure is the global mean square error. By normalizing this metric on the signals variance and taking the 10-logarithm of this ratio, the signal-to-noise (SNR) metric is obtained. If the normalization takes place on the signal (squared) peak value, the peak-SNR (PSNR) metric is obtained. Although it is known that this criterion may not correlate too well with the subjective ratings, we believe that the initial implementation of the benchmark should use the PSNR. Since the difference between host and watermarked data will be small in general, we expect that reasonable correlations between these initial ratings and the subjective quality of the watermarked data will be obtained.

In this paper, the evaluation of a watermarking system is associated to the watermark robustness, security, capacity and invisibility. The relationship between the watermark robustness, capacity and invisibility are shown as Fig. 2.

Watermark robustness and security. Watermark security refers to the ability for a watermarking system to survive in a hostile environment. Security for watermarking is naturally split into two kinds of security layers. Robustness against multiple watermarking is generally required especially against intentional attacks. Adding an additional watermark to a previously watermarked image must not remove the original watermark(s). If the original watermark(s) is (are) removed the image quality must degrade so the image will be unusable according to there requirements of the application.

Capacity. The term capacity stands for the amount of information that can be embedded within a given host data. More capacity implies more robustness since watermarks can be replicated.

Invisibility. In invisible watermarking, information is added as digital data to audio, picture or video, but it cannot be perceived as such (although it may be possible to detect that some amount of information is hidden). The watermark may be intended for widespread use and is thus made easy to retrieve or it may be a form of Steganography, where a party communicates a secret message embedded in the digital signal.

If the watermark information embedded in the larger capacity, the original image will be taken greater changes, then the watermark invisibility and robustness is likely to be affected; If we improve the invisibility of the embedded watermark, the watermark embedding capacity and robustness is likely to be

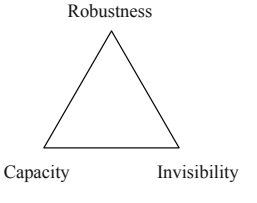


Fig. 2. The relationship between the watermark robustness, capacity and invisibility

the sacrifice; In order to improve the robustness of the watermark, it may affect the watermark embedding capacity and invisibility. For the different scenarios, we can strike a balance between the three points. Most of algorithms choose to lower the invisibility and capacity and to improve the robustness of the system.

3.1 Evaluation of Invisibility

Invisibility of digital watermark is also known as transparency. In the performance evaluation of digital watermark, the watermark transparency is a very important evaluation. Transparent assessment is divided into the subjective evaluation and objective evaluation.

(1) Subjective evaluation
The so-called subjective evaluation refers to the human visual effect as the evaluation criterion. That is given by the observer to judge the image quality. Subjective evaluation generally includes two steps. Firstly, images are ranked with the order from best to worst order. Secondly, observers assess each image based on the testing protocol to describe the processing of the object can be perceived. Table 2 lists the definition of image quality in the ITU-R Rec.500.

Table 2. The measure levels of subjective quality evaluation

Rating	Impairment	Qualilty
1	Not aware	Excellent
2	Perceived, but without prejudice to viewing	Good
3	Obstruct the view slightly	General
4	Prevent viewing	Poor
5	Hampered Watch	Very poor

In practice, the results of subjective evaluation varies from person to person. The mathematical models can not be used quantitatively to describe the image quality, and it is too time consuming. The application of subjective evaluation is very limited, so we want to use the objective, stable mathematical model to express the image quality.

(2) Objective evaluation
The commonly used objective evaluation methods include: the peak signal to noise ratio, mean square error, signal to noise ratio, the average absolute difference, Laplacian mean square errors etc. They are shown in the Table 3.

3.2 Evaluation of Robustness

Robust is the most important indicators of digital watermarking system. However, there is no mathematical proof about the robustness of digital watermark (whether positive or negative).The most common used evaluation of robustness

Table 3. The commonly used objective evaluation methods

Difference measurement	
Maximum differential equation	$MD = \max_{m,n} \lvert I_{m,n} - I'_{m,n} \rvert$
The mean absolute difference	$AD = \frac{1}{MN} \sum_{m,n} \left\lvert I_{m,n} - I'_{m,n} \right\rvert$
The average difference of the standard	$NAD = \sum_{m,n} \left\lvert I_{m,n} - I'_{m,n} \right\rvert / \sum_{m,n} \lvert I_{m,n} \rvert$
Mean squared error	$MSE = \frac{1}{MN} (I_{m,n} - \overline{I'_{m,n}})^2$
The standard squared error	$NMSE = \sum_{m,n} (I_{m,n} - \overline{I'_{m,n}})^2 / \sum_{m,n} I^2_{m,n}$
Differential P power mean value	$L^p = (\frac{1}{MN} \sum_{m,n} \left\lvert I_{m,n} - I'_{m,n} \right\rvert^p)^{1/p}$
SNR	$SNR = \sum_{m,n} I^2_{m,n} / \sum_{m,n} (I_{m,n} - \overline{I'_{m,n}})^2$
PSNR	$PSNR = MN \max_{m,n} I^2_{m,n} / \sum_{m,n} (I_{m,n} - \overline{I'_{m,n}})^2$
Image fidelity	$IF = 1 - \sum_{m,n} (I_{m,n} - \overline{I'_{m,n}})^2 / \sum_{m,n} I^2_{m,n}$
Relevant metrics	
Standard of mutual relations	$NC = \sum_{m,n} I_{m,n} I'_{m,n} / \sum_{m,n} I^2_{m,n}$
Related quality	$CQ = \sum_{m,n} I_{m,n} I'_{m,n} / \sum_{m,n} I_{m,n}$

is whether the watermarking algorithm is able to withstand the attack. Therefore, the robustness evaluation is based on the corresponding attacks.

The common attacks include: geometric attacks (such as translation, scaling, cropping, affine), simple attacks, simultaneous attacks, confusion attacks, collusion attacks. Robustness metrics include the correlation measure and bit error rate. The correlation measure usually use NC (normalized correlation) coefficient as the similarity measure of extracted watermark and the original watermark. Similarity is calculated as follows,

$$NC = \frac{\sum_{ij} w(i,j) * w'(i,j)}{\sqrt{\sum_{ij} w(i,j)^2} \sqrt{\sum_{ij} w'(i,j)^2}}. \tag{10}$$

Bit error rate is the ratio of the number of error bits to the number of all the embedded bits. Usually the correlation calculation is only used in determine whether a watermark exist or not, and in the other occasion, the bit error rate is used.

3.3 Evaluation of Security

Watermark security is a much broader concept than the robustness. Different watermark applications require different security evaluation. The application of the watermark in the military, must assume that the adversary's attack have the greatest power, then the security requirements of the watermark is very high. As to products watermark which is designed to avoid the use of children, the security requirements is only to resist the simplest attacks, and all the security watermark requirements relatively low. In each application of watermark, we need design the security according the actual requirements of system.

3.4 Other Evaluation

In addition to the above major evaluation methods, digital watermarking also have some parameter or properties associated to the evaluation methods, some of them are as follows,

Watermark capacity: the amount of information that can be embedded within a given host data.

The effectiveness of embedded watermark: the probability of successfully embedding watermark into a host data.

Blind detection: need not the information of original host image, the watermark can be detected successfully.

False alarm rate: the probability of detecting watermark in the image which has not any watermark in it.

Watermark key: by key or watermark encryption to control the watermark embedding and extraction.

Multiple watermarking: multiple non-interfering watermark can be embedded.

Calculation: the computing costs of watermark embedding and extraction algorithm.

4 Conclusions

This paper describes the traditional method of image quality evaluation, respectively, the subjective evaluation and objective evaluation. Digital watermark is an important application of image processing. Different applications require different watermark techniques, and the different watermark techniques require different evaluation criteria. Currently, there have not a complete evaluation system about digital watermark. Because the uniform description of the performance, the test methods, the method of attack, the standard test procedure have not been established. How to judge the performance of a watermarking system is very important. In this paper, the evaluation of a watermarking system is associated to the watermark robustness, security, capacity and invisibility.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 60873039), the Natural Science Foundation of Education Bureau

of Hunan Province (Grant No. 2008A520003), and the Science Foundation of Hunan University (Grant No. 07YBZR032).

References

1. Wang, K., Qiao, J., Kongas, A.: Quality Assessment of Digital Images. Measurement and Control Technology 19(5), 14–16 (2000)
2. Wang, Z., Xiao, W.: No-reference digital image quality evaluation based on perceptual masking. Computer Applications 26(12), 2838–2840 (2006)
3. Lu, W., Gao, X., Wang, T.: A Natural Image Quality Assessment Metric Based on Wavelet-based Contourlet Transform. ACTA Electronica Sinica 36(2), 303–308 (2008)
4. Wang, X., Zeng, M.: A new metric for objectively assessing the quality of enhanced images based on human visual perception. ACTA Electronica Sinica 19(2), 254–262 (2008)
5. Tong, Y., Zhang, Q., Chang, Q.: Image quality assessing model by using neural network and support vector machine. Journal of Beijing University of Aeronautics and Astronautics 32(9), 1031–1034 (2006)
6. Yang, C., Chen, G., Xie, S.: Gradient Information Based Image Quality Accessment. ACTA Electronica Sinica 35(7), 1313–1317 (2007)
7. Qi, Y., Ma, H., Tong, Y., Zhang, Q.: Image quality assessing model based on PSNR and SSIM. Computer Applications 27(2), 503–506 (2007)
8. Wang, T., Gao, X., Lu, W., Li, G.: A new method for reduced-reference image quality assessment. Journal of Xidian University 35(1), 101–109 (2008)
9. Wang, Z., Huang, L.: Image quality assessment method based on contrast sensitivity. Computer Applications 26(8), 22–33 (2006)
10. Xu, L., Ye, M., Zhang, Q.: New method to evaluate image quality. Computer Enginneering and Design 25(3), 418–420 (2004)
11. Wang, T., Gao, X., Zhang, D.: An Objective Content-based Image Quality Assessment Metric. Journal of Image and Graphics 12(6), 1002–1007 (2007)
12. Yang, W., Zhao, Y., Xu, D.: Method of image quality assessment based on human visual system and structural similarity. Journal of Beijing University of Aeronautics and Astronautics 34(1), 1–4 (2008)

Ensemble of Global and Local Features for Face Age Estimation

Wankou Yang[1,2], Cuixian Chen[1], Karl Ricanek[1], and Changyin Sun[2]

[1] Face Aging Group, Dept. Of Computer Science, UNCW, USA
[2] School of Automation, Southeast University, Nanjing 210096, China
wankou_yang@yahoo.com.cn, {chenc,ricanekk}@uncw.edu,
cysun@seu.edu.cn

Abstract. Automatic face age estimation is a challenging task due to its complexity owing to genetic difference, behavior and environmental factors, and also the dynamics of facial aging between different individuals. In this paper, we propose a feature fusion method to estimate the face age via SVR, which ensembles global feature from Active Appearance Model (AAM) and the local feature from Gabor wavelet transformation. Our experimental results on UIUC-PAL database show that our proposed method works well.

Keywords: AAM, Gabor, age estimation, feature fusion, ensemble.

1 Introduction

Human faces contain important information, such as gender, race, mood, and age [1,2]. Face age estimation has attracted great attentions recently in both research communities and industries, due to its significant role in human computer interaction (HCI), surveillance monitoring, and biometrics. However, there are many intrinsic and extrinsic factors which make it very difficult to predict the ages of human subjects from their face images accurately. The intrinsic factors include genetics, ethnicity, gender, and health conditions. The extrinsic factors include makeup, accessories, facial hair, and the variation of expression, pose and illumination. Furthermore, a face image of size $n_1 \times n_2$ is generally represented by a vector with dimensionality of or even more than $n_1 \times n_2$. It is still a challenging topic to significantly and effectively reduce the dimensionality from the original image space.

Recently, Yan et al. [3] proposed the patch-kernel regression (PKR) to study the human face age estimation and head pose estimation. Guo et al. [4] studied both manifold leanings to extract face aging features and local adjustment for age estimation. Luu et al. [5] proposed to conduct age estimation by a hierarchical model based on characteristics of human craniofacial development. Ricanek et al. [6] proposed a robust regression approach for automatic face age estimation, by employing Least Angle Regression (LAR) [7] for subset features selection. Chen et al. [7] studied an age estimation system tuned by model selection that outperforms all prior systems on the FG-NET face database. Most of the aforementioned publications on age estimation share the similar ideas: after facial features are extracted from the

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 251–259, 2011.

images, a dimension reduction method is applied to map the original vectors into a lower dimensional subspace. Then all or part of the components of the transformed vectors are used to construct a statistical model. Cootes et al. [8] proposed the Active Appearance Model (AAM) that described a statistical model of face shape and texture. It is a popular facial descriptor which makes use of the Principle Components Analysis (PCA) in a multi-factored way for dimension reduction while maintaining important structure (shape) and texture elements of face images. As pointed by Mark [9], shapes are accounted for the major changes during ones younger years, while wrinkles and other textural pattern variations are more prominent during ones older years . Since AAM extracts both shape and texture facial features, it is appropriate to use AAM in the age estimation system for feature acquisition. However, the adoption of PCA's in AAM can muddle important features because it attempts to maintain the greatest variance while creating orthogonal-projection vectors. Yan et al. [3] and Guo et al. [4] show that local features can be more robust against small misalignment, variation in pose and lightings.

On the other hand, Gabor wavelets have been applied successfully in image analysis and pattern recognition [10]. Therefore, applying Gabor wavelet transformation on the shape-normalized patch can take both advantages of shape model and local features. Each feature representation has its advantages and disadvantages. Fusing two feature representations via SVR could be a potential way to get an effective age estimation system.

2 Active Appearance Models

The active appearance model was first proposed by Cootes et al. [8]. AAM decouples and models shape and pixel intensities of an object. The latter is usually referred to as texture. The basic steps involved in building an AAM is as shown in Figure 1. A very important step in building an AAM model is identifying a set of landmarks and obtaining a training set of images with the corresponding annotation points either by hand, or by partial- to completely automated methods. As described in [8], the AAM model can be generated in three main steps: (1) A statistical shape model is constructed to model the shape variations of an object using a set of annotated training images. (2) A texture model is then built to model the texture variations, which is represented by intensities of the pixels. (3) A final appearance model is then built by combining the shape and the texture models.

2.1 Statistical Shape Model

A statistical shape model is built from a set of annotated training images. In a 2-D case, a shape is represented by concatenating n point vectors $\{(x_i, y_i)\}$

$$x = (x_1, x_2, \cdots, x_n, y_1, y_2, \cdots, y_n)^T \tag{1}$$

The shapes are then normalized by Procrustes analysis [10] and projected onto the shape subspace created by PCA

$$x = \bar{x} + P_s \bullet b_s \tag{2}$$

where $\bar{x}$ denotes the mean shape, $P_s = \{s_i\}$ is the matrix consisting of a set of orthonormal base vectors s_i and describing the modes of variations derived from training set, and b_s includes the shape parameters in the shape subspace. Subsequently, based on the corresponding points, images in the training set are warped to the mean shape to produce shape-free patches.

2.2 Statistical Texture Model

The texture model is generated very similar to the shape model. Based on the shape free patch, the texture can be raster scanned into a vector g. Then the texture is linearly normalized by the parameters $u = (\alpha, \beta)^T$ and g is given by

$$g = \frac{(gi - \beta \bullet 1)}{\alpha} \tag{3}$$

where α, β are the mean and the variance of the texture g, respectively, and $1=[1,1,\ldots,1]^T$ is the vector with the same length of g_i. The texture is ultimately projected onto the texture subspace based on PCA

$$g = \bar{g} + P_g \bullet b_g \tag{4}$$

where $\bar{g}$ is the mean texture, $P_g = \{g_i\}$ is the matrix consisting of a set of orthonormal base vectors g_i and describing the modes of variation derived from training set, and b_g includes the texture parameters in the texture subspace.

2.3 Combined Appearance Model

Finally, the coupled relationship between the shape and the texture is analyzed by PCA and the appearance subspace is created. At the end, the shape and the appearance can be described as follows:

$$x = \bar{x} + Q_s \bullet c \tag{5}$$

$$g = \bar{g} + Q_g \bullet c \tag{6}$$

where c is a vector of appearance parameters controlling both the shape and the texture, and Q_s and Q_g are matrices describing the modes of variation derived from the training set. Thus the final appearance model can be represented as $b = Qc$ where

$$b = \frac{W_s b_s}{b_g} = \frac{W_s (P_s)^T (x - \bar{x})}{(P_g)^T (g - \bar{g})} \tag{7}$$

and Q is the matrix of eigenvectors of b.

3 Gabor Feature Representation

Gabor wavelets were successfully used in face recognition. The Gabor wavelets, whose kernels are similar to the 2D receptive field profiles of the mammalian cortical simple cells, exhibit desirable characteristics of spatial locality and orientation selectivity, and are optimally localized in the space and frequency domains.

The Gabor wavelet can be defined as follows:

$$\psi_{u,v}(z) = \frac{\left\|k_{u,v}\right\|^2}{\sigma^2} e^{(-\left\|k_{u,v}\right\|^2 \left\|z\right\|^2 / 2\sigma^2)} \left[e^{ik_{u,v}z} - e^{-\sigma^2/2} \right] \tag{8}$$

where u and v define the orientation and scale of the Gabor kernels, $z = (x, y)$, $\|\bullet\|$ denotesthe norm operator, and the wave vector $k_{u,v}$ is defined as follows:

$$k_{u,v} = k_v e^{i\phi_u} \tag{9}$$

where $k_v = k_{\max} / 2^{v/2}$, and $\phi_u = u(\pi / 8)$. $k_{\max}$ is the maximum frequency, and f is the spacing factor between kernels in the frequency domain. In the most cases one would use Gabor wavelet of five different scales, $v = \{0, \cdots, 4\}$, and eight orientations, $u = \{0, \cdots, 7\}$.

The Gabor transformation of a given image $I(z)$ is defined as its convolution with the Gabor functions

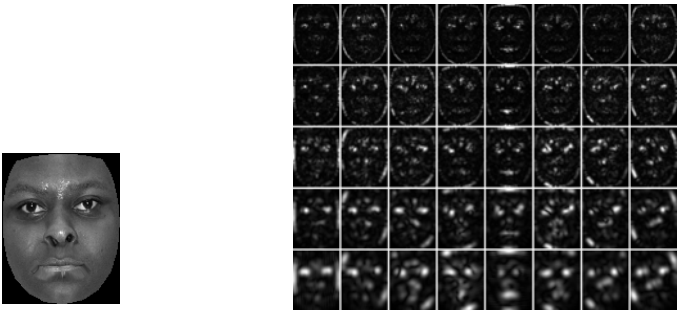
$$G_{u,v}(z) = I(z) * \psi_{u,v}(z) \tag{10}$$

where $z = (x, y)$ is the image position, $*$ is the convolution operator, and $G_{u,v}(z)$ is the convolution result corresponding to the Gabor kernel at scale v and orientation u. The Gabor wavelet coefficients are complex, which can be rewritten as:

$$G_{u,v}(z) = A_{u,v}(z) \bullet \exp(i\theta_{u,v}(z)) \tag{11}$$

with one magnitude item $A_{u,v}(z)$ and one phase item $\theta_{u,v}(z)$. We choose the magnitude as the feature representation of an image $I(z)$. Therefore, the set $S = \{A_{u,v}(z) : u \in \{0, \cdots, 7\}, v \in \{0, \cdots, 4\}\}$ forms the Gabor feature representation of the image $I(z)$. Fig.1 shows a face image and its corresponding Gabor magnitude feature images at five scale and eight orientation.

To encompass different spatial frequencies (scales), spatial localities, and orientation selectivities, we concatenate all these representation results and derive an augmented feature vector X. Before the concatenation, we first downsample each $A_{u,v}(z)$ by a factor ρ to reduce the space dimension, and normalize it to zero mean and unit variance. We then construct a vector out of the $A_{u,v}(z)$ by concatenating its row (or columns). Now, let $A_{u,v}^{(\rho)}$ denote the normalized vector construct from $A_{u,v}(z)$ (downsampled by ρ and normalized to zero mean and unit invariance), the augmented Gabor feature vector $A^{(\rho)}$ is then defined as follows:



(a) A sample image (b) The corresponding gabor feature representation

Fig. 1. Examples of Gabor Feature Representation

$$A^{(\rho)} = \left(A_{0,0}^{(\rho)t} A_{0,1}^{(\rho)t} \cdots A_{4,7}^{(\rho)t} \right) \tag{12}$$

where t is the transpose operation. The augmented Gabor feature vector thus encompasses all the elements (downsampled and normalized) of the Gabor feature representation set, $S = \left\{ A_{u,v}(z) : u \in \{0,\cdots,7\}, v \in \{0,\cdots,4\} \right\}$, as important discriminant information.

4 Our Proposed Face Age Estimation Method

4.1 The Algorithm

Our propose face age estimation method can be described as follows:

Step1. Do AAM on the image to get global feature *G* and the shape-free image.
Step2. Calculate the Gabor feature representation of the shape-free image.
Step3. Do PCA transformation on the Gabor feature representation to get a low dimensionality local feature *L*.
Step4. Do regress analysis to estimate the face age based on the Global feature G and local feature L via SVR.

Fig 2 shows the framework of our proposed face age estimation method.

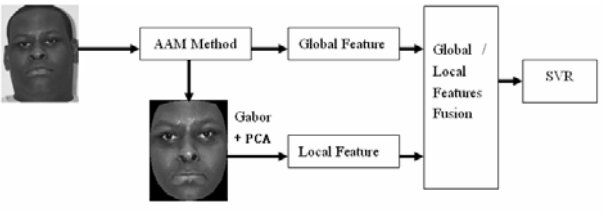


Fig. 2. Framework of our proposed face age estimation method

4.2 Performance Measure

The performance of age estimation is measured by the mean absolute error (MAE). The MAE is defined as the average of the absolute errors between the estimated ages and the observed ages, i.e. $MAE = \sum_{i=1}^{N} \left| \widehat{y}_i - y_i \right| / N$, where $\widehat{y}_i$ is the estimated age for the i^{th} test image, y_i is the corresponding observed age, and N is the total number of test images.

4.3 Face Aging Database

The UIUC Productivity Aging Laboratory (UIUC-PAL) face database [12] is selected for this experiment due to its quality of images and diversity of ancestry. Only the frontal images with neutral facial expression are selected for our age estimation algorithm. It contains 540 images with ages ranging from 18 to 93 years old. (See Figure 4 for sample images.) It is worth mentioning that UIUC-PAL is a multiethnicity adult database, which contains African-American, Asian, Caucasian, Hispanic and Indian. Fig 3 shows PAL sample images.

Fig. 3. Sample images of UIUC-PAL database

Histogram of age distribution of UIUC-PAL database is shown in Fig 4. The histogram of PAL database has two modes, one between age 18-20 and the other between 61 and 85.

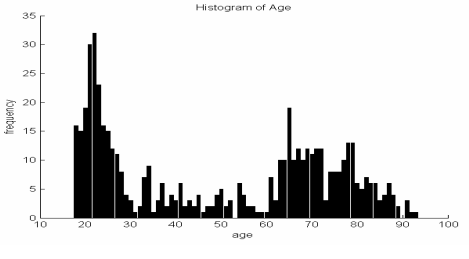


Fig. 4. Age Histogram of UIUC-PAL database

4.4 Experiment Setups

In UIUC-PAL database, each image is annotated with 161 landmarks as shown in [13]. The annotated faces with shape and texture information are presented to the AAM system to obtain the encoded appearance features, a set of transformed features

with dimension size 230. Here the AAM-Library tool [14] is utilized to implement the AAM system. Meanwhile the shape-free patch is also extracted from the annotated faces via the Active Shape Model provided by the AAM-Library tool. Next, Gabor wavelet transformation is applied on each shape-free image with 5 scales and 8 directions. Third, PCA transformation is performed on the Gabor wavelet feature representation to get final 400-dimensional feature.

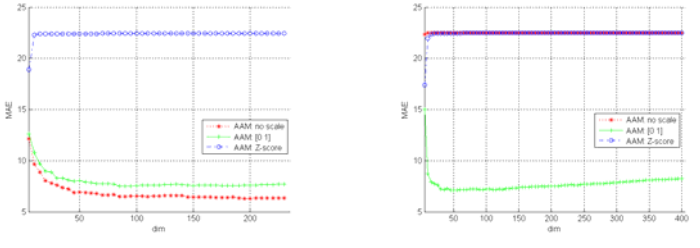
We use SVR as the age estimation regressor. We perform a standard 10-fold cross validation to evaluate the prediction error of the proposed method. We use the contributed package "Libsvm" [15] in Matlab for the computation of SVR (svm_type: EPSILON_SVR, kernel_type: RBF Kernel, gamma=0.25, C=64).

4.5 Experimental Results

In the first experiment, we compare three normalization methods with no-scaling on either AAM feature or the Gabor feature for age estimation. The experiment results are shown in Table 1. Fig. 5 shows the MAE curves with different feature dimensionality. For AAM features, no-scaling turns out to achieve the best MAE, comparing to the rest normalization methods. Here we compare two normalization methods: Min-Max and Z-score. Note that the Min-Max-[0, 1] with distinct hyper-parameters for SVR. On the other hand, for the Gabor features, Min-Max method gets the best results. In general sense, AAM features achieve better MAE consistently than Gabor features. It suggests that with single facial feature representation, AAM is one of the best facial feature representations. Based on the aforementioned results, hereafter, we only adopt the original AAM feature for further feature fusion studies. However, no-scaling method for Gabor is the worse case and we will not consider it any further in the feature fusion studies.

In the second experiment, we concatenate the AAM features with Gabor features with three different normalization methods. The results are shown in Table 2. In table 2, A denotes AAM, B denotes Gabor. From Table 2, we can find that it can improve the face age estimation result to ensemble the global and local features.

In Table 1 and Table 2, the superscript 1 means no scaling, superscript 2 denotes Min-Max-[0 1], superscript 3 denotes Z-score to normalize the features.



(a) MAE for AAM feature (b) MAE for Gabor feature

Fig. 5. MAE vs. feature dimensionality

Table 1. MAES of different normalization methods on global and local features, respectively

	AAM^1	AAM^2	AAM^3	$Gabor^1$	$Gabor^2$	$Gabor^3$
MAE	**6.29**	7.50	18.90	22.32	**7.10**	17.39
Std	**1.10**	0.91	1.21	1.23	**0.57**	1.60

Table 2. MAES of different normalization methods on global and local features

	AB^1	AB^2	AB^3
MAE	22.37	**5.88**	18.30
Std	1.22	**0.81**	1.52

5 Conclusions

In this paper, we propose a method to automatic estimate the face age via SVR, which fuse the global feature from AAM and the local feature from Gabor wavelet transformation. First, we perform AAM on the image to get the global feature and the shape-free image. Second, we do Gabor wavelet transformation on the shape-free image to get Gabor feature representation and do PCA to further reduce the dimensionality of Gabor feature representation. Third, we do regression analysis on the AAM feature and Gabor feature to estimate the face age via SVR. The experimental results on the UIUC database show that our proposed method has good performance. In the future work, we will research feature selection technologies to further improve the performance of the age estimation system.

Acknowledgments

This work is supported by the Intelligence Advanced Research Projects Activity, Federal Bureau of Investigation, and the Biometrics Task Force. The opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of our sponsors.

This work is also supported by NSF of China (90820009, 61005008, 60803049, 60875010), Program for New Century Excellent Talents in University Of China (NCET-08-0106), China Postdoctoral Science Foundation (20100471000) and the Fundamental Research Funds for the Central Universities (2010B10014).

References

1. Albert, A.M., Ricanek, K., Patterson, E.: A review of the literature on the aging adult skull and face: Implications for forensic science research and applications. Forensic Science International 172, 1–9 (2007)
2. Fu, Y., Guo, G., Huang, T.S.: Age synthesis and estimation via face: a survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(11), 1955–1976 (2010)

3. Yan, S., Zhou, X., Liu, M., Hasegawa-Johnson, M., Huang, T.S.: Regression from patch-kernel. In: ICPR 2008 (2008)
4. Guo, G.D., Fu, Y., Dyer, C., Huang, T.S.: Image-based human age estimation by manifold learning and locally adjusted robust regression. IEEE Transactions on Image Processing 17(7), 1178–1188 (2008)
5. Luu, K., Ricanek, K., Bui, T.D., Suen, C.Y.: Age estimation using active appearance models and support vector machine regression. In: IEEE Conf. on Biometrics: Theory, Applications and Systems (2009)
6. Ricanek, K., Wang, Y., Chen, C., Simmons, S.J.: Generalized multi-ethnic face age-estimation. In: IEEE Conf. on Biometrics: Theory, Applications and Systems (2009)
7. Chen, C., Chang, Y., Ricanek, K., Wang, Y.: Face age estimation using model selection. In: CVPRW 2010 (2010)
8. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. In: Burkhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1407, pp. 484–498. Springer, Heidelberg (1998)
9. Mark, L.S., Pittenger, J.B., Hines, H., Carello, C., Shaw, R.E., Todd, J.T.: Wrinkling and head shape as coordinated sources of age level information. Journal Perception and Psychophysics 27(2), 117–124 (1980)
10. Liu, C., Wechsle, H.: Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. IEEE Trans. IP 11(4), 467–476 (2002)
11. Goodall, C.R.: Procrustes methods in the statistical analysis of shape. J. Roy. Statist. Soc. B 53(2), 285–339 (1991)
12. Minear, M., Park, D.C.: A lifespan database of adult facial stimuli. Behavior Research Methods, Instruments, & Computers 36, 630–633 (2004)
13. Patterson, E., Sethuram, A., Albert, M., Ricanek, K.: Comparison of synthetic face aging to age progression by forensic sketch artist. In: IASTED International Conference on Visualization, Imaging, and Image Processing, Palma de Mallorca, Spain (2007)
14. Aam-library, `http://groups.google.com/group/asmlibrary?pli=1`
15. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), software `http://www.csie.ntu.edu.tw/cjlin/libsvm`

A Filter Based Feature Selection Approach Using Lempel Ziv Complexity

Sultan Uddin Ahmed[1], Md. Fazle Elahi Khan[2], and Md. Shahjahan[2]

[1] Dept. of Electronics and Communication Engineering
[2] Dept. of Electrical and Electronic Engineering
Khulna University of Engineering and Technology (KUET), Khulna – 9203, Bangladesh
{sultan_ahmed001,mdjahan8}@yahoo.com

Abstract. In this paper, a new filter based feature selection algorithm using Lempel-Ziv Complexity (LZC) measure, called 'Lempel Feature Selection' (LFS), is proposed. LZC finds the number of unique patterns in a time series. A time series is produced from the values of a feature and LZC of the feature is computed from the time series. The features are ranked according to their values of LZCs. Features with higher valued LZCs are selected and lower ones are deleted. LFS requires very less computation, since it just computes the LZCs of all features once. LFS is tested on several real world benchmark problems such as soybean, diabetes, ionosphere, card, thyroid, cancer, wine, and heart disease. The selected features are applied to a neural network (NN) learning model. NN produces better results with the selected features than that of randomly selected features.

Keywords: Feature selection, Filter based methods, Lempel-Ziv complexity, Neural network, Classification.

1 Introduction

The real world machine learning datasets consist of large number of relevant and redundant features. Selecting the relevant feature subset from the entire feature set has been a fertile field for the last several decades [1]. Feature selection (FS) increases the generalization ability, convergence, and comprehensibility of the learning model [1-3]. FS methods are computationally expensive, since only one feature subset is selected from 2^n feature subsets, where n is the number of features. However, appropriate feature selection techniques are essential for machine learning tasks when dealing with high dimensional or even low dimensional data.

There are huge numbers of feature selection approaches based on attenuation function and biomarkers [4, 5], correlation dimension [6], fuzzy and genetic algorithms [7, 8], ant colony optimization [7, 9] etc. They are broadly classified as wrapper and filter methods [2]. In the former one, a learning model is run on a particular subset of features and the resulting accuracy on the test set is then used to evaluate the feature subset. The subset which shows highest accuracy is finally selected. On the other hand, the later one filters out features which are irrelevant to the target concept [1]. Filter based methods rank the features according to some discrimination measure and select features having higher ranks. Filter methods have

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 260–269, 2011.

the advantage over wrapper-based methods in that the formers are computationally inexpensive and they do not need to consider all the subsets of features.

A number of filter based algorithms are found in the literature [10-12]. Features can be selected with measuring the correlations among the features together with classes [13, 14]. A feature having the higher correlation with its corresponding target is selected, while the lower ones are deleted. Some methods measure the correlation among the features [15, 16]. In this case, selecting the highly correlated features may be converged finally if less correlated features are deleted. That is why these approaches do not perform well in all the cases. Information theoretic measure has been used in ranking the features [17-21]. In this case, the mutual information among the features together with classes is computed and the redundant features are deleted. They have been successfully applied in many application areas such as microarray data, gene selection etc. Fractal correlation dimension (CFD) has been used as discrimination criterion to filter out the redundant features [6]. Although it has been succeeded in promoter recognition problem, it requires high computational cost and it is suitable to chaotic data.

Complexity means a large number of parts in intricate arrangement. Recently, there is great interest in measuring the complexity of system to solve real world problems [22]. Lempel-Ziv Complexity (LZC) is a good mathematical tool to compute the number of unique patterns in a time series [23]. Making use of LZC in feature selection is interesting. This paper deals with this issue for the first time.

In this paper, a filter based FS technique (LFS) that requires low computational effort is proposed. LFS ranks the features according to the values of LZC [23]. LZC finds the number of unique patterns in a time series. A time series is produced from the values of a feature and LZCs of the features are computed from the time series accordingly. The feature with higher value of LZC is selected and the lower one is deleted. In order to justify LFS, the selected features are applied to neural network (NN) learning model [24]. In the experiments, several benchmark classification problems such as soybean, diabetes, ionosphere, card, thyroid, cancer, wine, and heart disease are used. LFS is compared with one that selects features randomly. It is shown that the selected features with LFS produces good generalization ability in NN learning.

LFS has a number of advantages such as (i) it helps to produce important features which are indeed necessary for the training, (ii) it helps to produce robust classification, and (iii) it helps to remove the redundant information from the dataset. The rest of the paper is organized as follows. Section 2 presents a brief of Lempel-Ziv Complexity measure. The proposed algorithm LFS is described in section 3. Section 4 contains the experimental results. The paper is concluded in section 5.

2 Lempel-Ziv Complexity

A definition of complexity of a string is given by the number of the bits of the shortest computer program which can generate this string. There are many techniques to compute the complexity of a string. Among them algorithmic complexity [25-27], the Lempel-Ziv complexity [28], and a statistical measure of complexity [29] are most useful. The Lempel-Ziv algorithm objectively and quantitatively estimates the system

complexity through the change process of system structure, and has overcome the limitation of depicting the complexity through characteristic quantities of statistical complexity [30]. A short description of LZC measure is given below.

Lempel and Ziv proposed a useful complexity measure [23], [28], which finds the number of unique patterns in a time series. A standard algorithm how to compute LZC of a binary string *S* is realized from Fig. 1. The method uses merging and copying operations. Suppose, the *i-th* element of the string *S* is s_i and c(n) denotes the LZC of *S*. B1 and B2 are two buffers in computer memory. The method searches the unique patterns along each component of S and stores in a buffer consecutively. At first, c(n) is set to zero, the method increases c(n) by one if a new pattern is found. Final value of c(n) indicates the LZC of S.

1. Consider a binary string S with length n
2. Set: B1=NULL, B2=NULL, c(n)=0 and i=1
3. Repeat step 4 to step 7 while $i \leq n$
4. Store s_i *at the end of B1*
5. If i>1
Store s_{i-1} *into the B2*
Else
B2=NULL
6. If B2 does not contain B1
Set: c(n)=c(n)+1 and B1=NULL
7. Set: i=i+1
8. End

Fig. 1. Computation of Lempel-Ziv Complexity

According to the algorithm, a binary string consisting only of 0's (or 1's) has the complexity 2, since there are two unique patterns 0 and 000……. . A string consisting of sequence of 01's i.e. 01010101……01 has complexity 3, since there are three unique patterns 0, 1, 010101……. . For the binary string 0001101001000101, the unique patterns are 0, 001, 10, 100, 1000, 101 and so its LZC is 6.

3 LFS

The values of a feature are the floating values in the range of zero to unity. If a value is greater than 0.5, it is considered as one, otherwise zero. Hence a binary string is formed for a feature. The LZC of the string is computed and it is considered as the LZC of corresponding feature. In a similar fashion, LZCs of all the features are computed. The median value (MD) of LZCs is determined. If a feature has LZC less than MD, it is deleted. A feature with LZC of greater than MD is selected. In order to test the selected features, they are applied to a neural network classifier [24]. The consecutive steps of LFS are shown in Fig. 2.

a) Construct the binary strings of features
b) Calculate LZCs of all features
c) Calculate median value (MD) of LZCs
d) Repeat for k features
If(LZC_k<MD)
Delete k feature
End
End
e) Design a classifier for selected features
f) Train the classifier
g) Test the accuracy of classifier

Fig. 2. Algorithmic flow of LFS

4 Experimental Studies

4.1 Characteristics of Datasets

In order to demonstrate the performance of LFS, eight real world benchmark datasets such as soybean, diabetes, ionosphere, card, thyroid, cancer, wine, and heart disease are used. The datasets are collected from the University of California at Irvine (UCI) Repository of the machine learning database [31] and PROBEN1 [32]. The characteristics of datasets are listed in Table 1. For example, soybean is a 19 class problem with 82 features. It has total 683 examples. The total examples are divided into – training, validation, and testing sets. The training examples are used to train model, while the testing examples are used to test the generalization ability of learning.

Table 1. Characteristics of datasets

Datasets	Number of		
	examples	features	classes
soybean	683	82	19
diabetes	768	8	2
ionosphere	351	33	2
card	690	51	2
thyroid	7200	21	3
cancer	699	9	2
wine	178	13	3
heart	920	35	2

4.2 Experimental Results and Comparison

LZC may depend on the order of examples. Therefore, the examples are randomly arranged. LFS selects the feature subset and the selected features are applied to a

feed-forward neural network (NN) [24]. A NN consisting of single hidden layer is considered. The number of input nodes is equal to the number of selected features, the number of hidden nodes is arbitrary taken, and the number of output nodes is equal to the number of classes. The weights of NN are initially randomized in the range [-0.1, +0.1]. Backpropagation algorithm is used to train the NN to a desired accuracy [24]. In order to check the generalization ability, 'testing error rate' (TER) is defined as the ration of number of misclassified examples to the number of examples over testing set. The percentage average value (mean) of TERs over 20 independent trials are reported at several learning rate (η).

It is often very difficult to compare with other algorithms. This is because different algorithms have the different setups and focusing issues. It is quite difficult to develop an algorithm that coincides with all conditions of other algorithms. Therefore, an algorithm called 'random feature selection' (RFS) is developed in order to make a fair justification of the effectiveness of LFS. RFS is same as LFS except that it deletes features randomly without making any feature wise discrimination criteria. The same number of features that were deleted by LFS should be deleted by RFS. The results obtained with LFS and RFS are shown for several problems.

Soybean. LFS selects 45 features from 82 and RFS selects the same number of features randomly. The selected features are applied to a 45-30-19 (45 input nodes, 30 hidden nodes and 19 output nodes) NN. Soybean consists of total 683 examples. The first 342 examples are used as training and the last 170 examples as testing. Table 2 shows that LFS has always lower TER than that of RFS. For examples, at η=0.2, TERs of LFS and RFS are 8.00 and 9.35 respectively.

Table 2. Testing error rates (TERs) of soybean problem with LFS and RFS over 20 independent trials

η	TER	
	LFS	RFS
0.20	8.00	9.35
0.30	8.00	12.71
0.40	8.06	13.12

Diabetes. LFS selects 5 features. A NN size of 5-3-2 is considered. Diabetes problem consists of total 768 examples. The first 348 examples are used as training and the last 192 examples as testing. TERs obtained from NN training are reported in Table 3. The features selected with LFS always show lower TERs than those selected with RFS. For examples, at η=0.1, TERs of LFS and RFS are 21.51 and 26.88 respectively.

Table 3. TERs of diabetes problem with LFS and RFS over 20 independent trials

η	TER	
	LFS	RFS
0.10	21.51	26.88
0.20	21.66	26.51
0.30	22.18	28.33

Ionosphere. Both LFS and RFS select 18 features from 33. The size of NN is 18-7-2. The first 175 examples are used to train the NN and the last 88 examples are used to test the trained NN. The results are shown in Table 4. Although at η=0.3, LFS and RFS show same results, LFS marginally improves the results at other learning rates.

Table 4. TERs of ionosphere problem with LFS and RFS over 20 independent trials

η	TER	
	LFS	RFS
0.10	9.89	12.27
0.20	11.59	12.27
0.30	11.47	11.47

Card. 27 features are selected from 51. The first 345 examples are used to train a NN of 27-10-2 and the last 172 examples for testing. LFS obtains lower TERs than that of RFS as shown in Table 5. For examples, at η=0.1, TER of LFS is 14.07 while it is 20.93 with RFS.

Table 5. TERs of card problem with LFS and RFS over 20 independent trials

η	TER	
	LFS	RFS
0.10	14.07	20.93
0.20	14.19	20.47
0.30	14.07	19.65

Thyroid. Thyroid has total 21 features and the number of selected features is 11. A NN having 6 hidden nodes is trained with first 3600 examples. The last 1800 examples are used as testing. Table 6 contains the TERs with LFS and RFS. LFS always obtains lower TER than that of RFS. One can say that LFS selects the features that are relevant in the training.

Table 6. TERs of thyroid problem with LFS and RFS over 20 independent trials

η	TER	
	LFS	RFS
0.10	3.24	6.73
0.20	1.69	5.36
0.30	2.41	4.48

Cancer. A 5-3-2 NN is trained with first 350 examples of 5 selected features. The trained NN is tested with last 174 examples. Table 7 shows that the selected features with LFS can train the model with good generalization ability. RFS can select the relevant or redundant features, while LFS selects only the relevant features.

Table 7. TERs of cancer problem with LFS and RFS over 20 independent trials

η	TER	
	LFS	RFS
0.10	0.69	2.75
0.15	0.63	2.18
0.20	0.63	2.59

Wine. 7 features are selected from 13. A NN size of 7-4-3 is trained with first 89 examples and the last 44 examples are used for testing. LFS always achieves zero TER as shown in Table 8. It is understood that there is some redundancy in the entire features which are extracted out and removed by LFS, although wine is an easy problem for NN.

Table 8. TERs of wine problem with LFS and RFS over 20 independent trials

η	TER	
	LFS	RFS
0.10	00.0	1.36
0.20	00.0	0.68
0.30	00.0	0.91

Heart Disease. The number of selected features is 18. A NN of 18-5-2 is trained with first 460 examples and it is tested with last 230 examples. The experimental results are reported in Table 9 in terms of TER. The selected features with LFS shows good classification rate than of with RFS.

Table 9. TERs of heart problem with LFS and RFS over 20 independent trials

η	TER	
	LFS	RFS
0.1	19.74	21.65
0.15	20.26	21.65
0.2	20.57	21.83

4.3 Experiments by Varying Threshold Point – The Median (MD) Value

LFS considers the median value (MD) as a threshold to separate the relevant and redundant features. In this section, the effectiveness of LFS whether or not it can produce expected results if more features are allowed to delete followed by a modified threshold point is investigated. Several modified threshold points such as (MD+ 5%), (MD+10%) and (MD+15%) are made. In this case, LFS will remove those features which have LZC values smaller than the modified ones.

The results are listed in Table 10 for soybean, card, and heart disease problems. It is clear from the table that up to a threshold point (MD+5%); the results are approximately similar with those for the threshold MD. If the features are further removed by making a threshold such as (MD+10%) and (MD+15%), the average results greatly deteriorate for heart disease problem, while it slightly deteriorate for

soybean and card problems. One reason behind this may be removal of some important features by LFS with the new thresholds - (MD+10%) and (MD+15%). This indicates that the median is appropriate threshold point to decide how many features should be removed, although the threshold value has not been optimized.

Table 10. TERs of LFS by varying the threshold 'median' (MD) over 20 independent trials

Problem	η	Threshold	TER
Soybean	0.2	MD+5%	9.88
		MD+10%	10.29
		MD+15%	10.35
	0.3	MD+5%	9.18
		MD+10%	10.06
		MD+15%	9.47
	0.4	MD+5%	9.29
		MD+10%	9.53
		MD+15%	9.76
Card	0.1	MD+5%	15.41
		MD+10%	15.35
		MD+15%	15.47
	0.2	MD+5%	15.20
		MD+10%	15.35
		MD+15%	15.47
	0.3	MD+5%	15.47
		MD+10%	15.64
		MD+15%	17.70
Heart	0.1	MD+5%	20.30
		MD+10%	25.34
		MD+15%	25.53
	0.15	MD+5%	20.43
		MD+10%	25.17
		MD+15%	25.17
	0.2	MD+5%	20.30
		MD+10%	24.96
		MD+15%	33.30

5 Conclusions

A new filter based feature selection algorithm to find a feature subset from entire feature set using LZC measure, called LFS is presented in this paper. Features are ranked according to the higher values of LZC. Features with higher valued LZCs are chosen for training and lower valued ones are deleted according to a threshold level. In fact LZC is a good tool to determine the number of unique patterns in a feature and hence LFS selects the features which have higher number of unique patterns. The features obtained from the scheme produces promising results after applying to a NN learning model. The algorithm is tested on several real world benchmark classification problems such as soybean, diabetes, ionosphere, card, thyroid, cancer, wine, and heart disease. The effectiveness of LFS is justified throughout the various

experimental conditions such as different threshold points, different learning rates etc. The method produces good results when it is compared with one that selects features randomly.

References

1. Guyon, I., Elisseeff, A.: An Introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182 (2003)
2. Dash, M., Liu, H.: Feature selection for classifications. Intelligent Data Analysis: An International Journal 1, 131–156 (1997)
3. Kohavi, R., John, G.: Wrappers for feature subset selection. Artificial Intelligence 97, 273–324 (1997)
4. Pal, N.R., Chintalapudi, K.: A connectionist system for feature selection. International Journal of Neural, Parallel and Scientific Computation 5, 359–381 (1997)
5. Pal, N.R.: Computational intelligence for bioinformatics applications, Tutorial. In: ICONIP, Kyushu, Japan (2007)
6. Bhavani, S.D., Rani, T.S., Bapi, R.S.: Feature selection using correlation dimension: Issues and Applications in binary classification problems. Applied Soft Computing 8, 555–563 (2008)
7. Sivagaminathan, R.K., Ramakrishnan, S.: A hybrid approach for feature subset selection using neural networks and ant colony optimization. Expert Systems with Applications 33, 49–60 (2007)
8. Chakraborty, D., Pal, N.R.: A neuro-fuzzy scheme for simultaneous feature selection and fuzzy rule-based classification. IEEE Trans. on Neural Networks 15(1), 110–123 (2004)
9. Ani, A.: Feature subset selection using ant colony optimization. International Journal of Computational Intelligence 2(1), 53–58 (2005)
10. Watanabe, H., Yamaguchi, T., Katagiri, S.: Discriminative Metric Design for Robust Pattern Recognition. IEEE Trans. on Signal Processing 45(11), 2655–2662 (1997)
11. Liu, Y., Zheng, Y.F.: FS SFS: A Novel Feature Selection Method for Support Vector Machines. Pattern Recognition Society (2005)
12. John, G.H., Kohavi, R., Pfleger, K.: Irrelevant features and the subset selection problem. In: Proc. 11th Int. Conf. Machine Learning (ICML), San Francisco, Canada, pp. 121–129 (1994)
13. Hall, M.: Correlation based feature selection for machine learning. Doctoral dissertation, University of Waikato, Dept. of Computer Science (1999)
14. Mitra, P., Murthy, C.A., Pal, S.K.: Unsupervised feature selection using feature similarity. IEEE Transaction on Pattern Analysis and Machine Intelligence 24(3) (2002)
15. Yu, L., Liu, H.: Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution. In: Proceedings of the Twentieth International Conference on Machine Learning, Washington DC (2003)
16. Michalak, K., Kwasnicka, H.: Correlation-based Feature Selection Strategy in Neural Classification. In: Sixth International Conference on Intelligent Systems Design and Applications, vol. 1, pp. 741–746 (2006)
17. Meyer, P.E., Schretter, C., Bontempi, G.: Information-Theoretic Feature Selection in Microarray Data Using Variable Complementarity. IEEE Trans. Selected Topics in Signal Processing 2(3), 261–274 (2008)
18. Estevez, P.A., Tesmer, M., Perez, C.A., Zurada, J.M.: Normalized Mutual Information Feature Selection. IEEE Transactions on Neural Networks 20(2), 189–200 (2009)

19. Peng, H., Long, F., Ding, C.: Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(8), 1226–1238 (2005)
20. Eriksson, T., Kim, S., Kang, H.G., Lee, C.: An Information-Theoretic Perspective on Feature Selection in Speaker Recognition. IEEE Signal Processing Letters 12(7), 500–503 (2005)
21. Chow, T., Huang, D.: Estimating Optimal Feature Subsets Using Efficient Estimation of High-Dimensional Mutual Information. IEEE Transactions on Neural Networks 16(1), 213–224 (2005)
22. Jorgensen, T.D., Haynes, B.P.: Pruning artificial neural networks using neural complexity measures. International Journal of Neural Systems 18(5), 389–403 (2008)
23. Steeb, W.-H.: The Non linear Work Book, ch. 2, 3rd edn. World Scientific Publishing Co., Singapore
24. Song, Q., Soh, Y.C., Zhao, L.: A robust extended Elman backpropagation algorithm. In: The Proc. of International Joint Conference on neural network (IJCNN 2009), pp. 2971–2978 (2009)
25. Szczepanski, J., Amigo, J., Wajnryb, E., Vives, S.: Characterizing spike trains with Lempel-Ziv complexity. Neurocomputing 58–60, 79–84 (2004)
26. Chaitin, G.J.: Information, Randomness and Incompleteness. World Scientific, Singapore (1987)
27. Chaitin, G.J.: Algorithmic Information Theory. Cambridge University Press, Cambridge (1987)
28. Lempel, A., Ziv, J.: On the complexity of finite sequences. IEEE Transaction on Information Theory 22(1), 75–81 (1976)
29. Li, M., Vitanyi, P.: An Introduction to Kolmogorov Complexity and Its Applications. Springer, Heidelberg (1997)
30. Liu, F., Tang, Y.: Improved Lempel-Ziv Algorithm Based on Complexity Measurement of Short Time Series. In: Fourth International Conference on Fuzzy Systems and Knowledge Discovery (2007)
31. Asuncion, A., Newman, D.: UCI Machine Learning Repository, Schl. Inf. Comput. Sci., Univ. California, Irvine, CA (2007)
32. Prechelt, L.L.: PROBEN1-A set of neural network benchmark problems and benchmarking rules. Technical Report 21/94, Faculty of Informatics, University of Karlsruhe (1994)

Finger-Knuckle-Print Recognition Using LGBP

Ming Xiong, Wankou Yang, and Changyin Sun

School of Automation, Southeast University, Nanjing 210096, China
cysun@seu.edu.cn

Abstract. Recently, a new biometrics, finger-knuckle-print recognition, has attractive interests of researchers. The popular techniques used in face recognition are not applied in finger-knuckle-print recognition. Inspired by the success of Local Gabor Binary Patterns (LGBP) in face recognition, we present a method that uses LGBP to identify finger-knuckle-print images. The experimental results show that our proposed method works well.

Keywords: finger-knuckle-print, Gabor feature representation, LBP.

1 Introduction

With the rapid development of computer techniques, in the past three decades researchers have exhaustively investigated the use of a number of biometric characteristics [1,2], including fingerprint, face, iris, retina, palm-print, hand geometry, finger surface shape, voice, ear, gait and signature, etc. Although many biometric techniques are still under the stage of research and development, some biometric systems have been developed and used in a large scale; for example, the Hong Kong government has been using the fingerprint recognition system as the automated passenger clearance system (e-channel) since 2004 [3].

Recently, researchers noticed that the textures in the outer finger surface, especially in the area around finger joint, has the potential to do personal authentication. D. L. Woodward et al. [4,5] recently used the 3D range image of the hand to calculate the curvature surface representation of the index, middle, and ring fingers for similarity comparison. In [6], C. Ravikanth et al. applied the subspace analysis methods, which are widely used in appearance based face recognition, to the finger-back surface images for feature extraction and person classification. The above works made a good effort to validate the uniqueness of biometric characteristic in the outer finger surface; however, they did not provide a practical solution to establishing an efficient system using the outer finger surface features. In addition, the method [4,5] mainly exploits the 3D shape information of finger back surface but does not fully use the texture information; while the subspace analysis methods used in [6] may not be able to effectively extract the distinctive line and junction features in finger back surface. Since the main features in an FJP image are lines, L. Zhang et al. designed a hardware captured device and used multiple 2-D Gabor filters along different directions to filter the finger-knuckle-print image and extract the local orientation information. The local orientation was then coded using a competitive coding scheme and then two FJP images can be matched by calculating their angular distance with the code maps [7].

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 270–277, 2011.

W. Zhang et al. [8,9] proposed a method for face recognition, named Local Gabor Binary Patterns (LGBP), which combined Gabor wavelet and LBP and has achieved impressive performance. Inspired by LGBP, we propose that use Gabor feature representation and locale binary pattern (LBP) to identify finger-knuckle-print images in this paper. The kernels of Gabor wavelets are similar to two-dimensional receptive field profiles of the mammalian cortical simple cells, exhibit desirable characteristics of spatial locality and orientation selectivity. The LBP operator has strong gray-scale and rotation invariance, can greatly conquer rotating shift and uneven illumination problem which is hardly to be avid in pretreatment process, so as to exact Finger-Knuckle-Print feature more effectively.

The rest of this paper is organized as follows. Section 2 introduces Gabor feature representation and LBP. Section 3 reports the experimental results. Finally, conclusions are presented in Section 4.

2 Gabor and LBP

2.1 Gabor Feature Representation

Gabor wavelets were successfully used in face recognition. The Gabor wavelets, whose kernels are similar to the 2D receptive field profiles of the mammalian cortical simple cells, exhibit desirable characteristics of spatial locality and orientation selectivity, and are optimally localized in the space and frequency domains.

The Gabor wavelet can be defined as follows:

$$\psi_{u,v}(z) = \frac{\left\| k_{u,v} \right\|^2}{\sigma^2} e^{(-\left\| k_{u,v} \right\|^2 \left\| z \right\|^2 / 2\sigma^2)} \left[e^{ik_{u,v}z} - e^{-\sigma^2/2} \right] \tag{1}$$

where u and v define the orientation and scale of the Gabor kernels, $z = (x, y)$, $\|*\|$ denotesthe norm operator, and the wave vector $k_{u,v}$ is defined as follows:

$$k_{u,v} = k_v e^{i\phi_u} \tag{2}$$

where $k_v = k_{\max} / 2^{v/2}$, and $\phi_u = u(\pi/8)$. $k_{\max}$ is the maximum frequency, and f if the spacing factor between kernels in the frequency domain. In the most cases one would use Gabor wavelet of five different scales, $v = \{0, \cdots, 4\}$, and eight orientations, $u = \{0, \cdots, 7\}$.

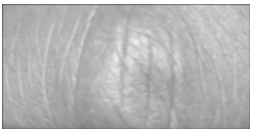
The Gabor transformation of a given image $I(z)$ is defined as its convolution with the Gabor functions

$$G_{u,v}(z) = I(z) * \psi_{u,v}(z) \tag{3}$$

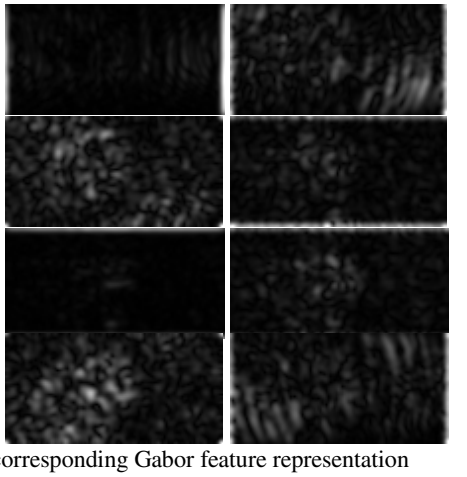
where $z = (x, y)$ denotes the image position, $*$ denotes the convolution operator, and $G_{u,v}(z)$ is the convolution result corresponding to the Gabor kernel at scale v and orientation u . The Gabor wavelet coefficients is a complex, which can be rewritten as:

$$G_{u,v}(z) = A_{u,v}(z) * \exp(i\theta_{u,v}(z)) \tag{4}$$

with one magnitude item $A_{u,v}(z)$ and one phase item $\theta_{u,v}(z)$. We choose the magnitude as the feature representation of an image $I(z)$. Therefore, the set $S = \{A_{u,v}(z) : u \in \{0,\cdots,7\}, v \in \{0,\cdots,4\}\}$ forms the Gabor feature representation of the image $I(z)$. Fig. 1 shows Gabor feature representation of one scale and eight orientations of one Finger-Knuckle-Print image, with the following parameters: $\sigma = 2\pi$, $k_{\max} = 2.5 * \pi / 2$.



(a) One FKP image



(b) The corresponding Gabor feature representation

Fig. 1. Gabor feature representation of one scale and eight orientation

To encompass different spatial frequencies (scales), spatial localities, and orientation selectivities, we concatenate all these representation results and derive an augmented feature vector X. Before the concatenation, we first downsample each $A_{u,v}(z)$ by a factor ρ to reduce the space dimension, and normalize it to zero mean and unit variance. We then construct a vector out of the $A_{u,v}(z)$ by concatenating its row (or columns). Now, let $A_{u,v}^{(\rho)}$ denote the normalized vector construct from $A_{u,v}(z)$ (downsampled by ρ and normalized to zero mean and unit invariance), the augmented Gabor feature vector $A^{(\rho)}$ is then defined as follows:

$$A^{(\rho)} = \left(A_{0,0}^{(p)t} A_{0,1}^{(p)t} \cdots A_{4,7}^{(p)t}\right) \tag{5}$$

where t denotes the transpose operation. The augmented Gabor feature vector thus encompasses all the elements (downsampled and normalized) of the Gabor wavelet representation set, $S = \{A_{u,v}(z) : u \in \{0,\cdots,7\}, v \in \{0,\cdots,4\}\}$, as important discriminant information.

2.2 Locale Binary Pattern (LBP)

The original LBP method is introduced by Ojala to be used in texture description [10][11]. It is based on thresholding neighborhood pixel values against the center pixel in a circular order to form a binary pattern. Then these patterns of different pixels are assorted and concatenated into a histogram so that each pattern corresponds to one bin. This histogram is used to represent the original image for later classification purpose. Fig. 2 gives an illustration of the basic LBP operator. Fig 3 shows a LBP encode image.

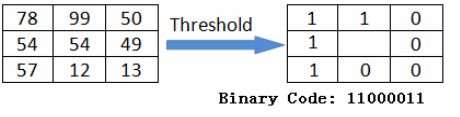


Fig. 2. The basic LBP operator

Fig. 3. LBP transform image of one Finger-Knuckle-Print image

We can use $LBP_{P,R}$ to denote LBP operators with different sizes, in which (P,R) means P sampling points on a circle of radius R. It allows for any value of P and R, for the gray values of neighbors which do not fall exactly in the center of pixels is estimated by bilinear interpolation. The total 2^P different patterns are concatenated into a histogram by their number of occurrences.

Allowing for that experimental results show certain local binary patterns are fundamental properties of texture images, Ojala proposed an improved LBP operator called *uniform patterns*, which contain at most two 0/1 or 1/0 transitions when the binary string is considered circular. We denote it by $LBP^{u2}_{P,R}$, in which $u2$ reflects the use of rotation invariant uniform patterns with bit transitions at most two. For example, the $LBPLBP^{u2}_{8,1}$ operator quantifies the total 256 LBP values into 59 bins according to uniform strategy (58 uniform patterns and the other patterns are assorted to the 59th pattern). Ojala reported in their experiments with texture images, uniform patterns account for a bit less than 90% of all patterns when using the (8,1) neighborhood and for around 70% in the (16,2) neighborhood[10].

The histogram of image is defined as

$$H_i = \sum_{x,y} I\{f_l(x,y) = i\} \quad i = 0,1,\cdots,n-1 \tag{6}$$

in which n is the number of different labels produced by LBP operator.

$$I(A) = \begin{cases} 1 & \text{A is true} \\ 0 & \text{A is false} \end{cases} \tag{7}$$

A large number of documents and experimental results showed that extract the LBP histogram of the whole image in the high-capacity database is far from enough. Take Finger-Knuckle-Print as example, if we only use the LBP histogram of the whole image as the discriminate feature, the recognition rate is quite low. To solve the

problem, we usually divide the original image to several blocks, e.g. 3*3 or 7*7, calculates the LBP histogram of each sub-image. Then, we concatenate all these histogram results and derive a new one, as the feature representation of the whole image. In this way, we merge the locale feature and the whole information of the image together effectively.

Algorithms such as Histogram Intersection, Log-likelihood Statistic and Chi Square Statistic can be used to discriminate histogram features [12]. In proposed algorithm, we use Chi Distance Statistic to calculate the distance between histogram features:

$$\chi^2(S,M) = \sum_i \frac{(S_i - M_i)^2}{S_i + M_i} \tag{8}$$

in which S denotes the test image to be recognized, M denotes marked images in library, S_i means probability of test image in the ith area of histogram, M_i means probability of marked image in the ith area of histogram.

When we calculate the $LBP^{u2}_{P,R}$ operator histogram of the image in blocks, the Chi Distance Statistic could be extended as follows:

$$\chi^2(S,M) = \sum_{i,j} \frac{(S_{i,j} - M_{i,j})^2}{S_{i,j} + M_{i,j}} \tag{9}$$

2.3 The Algorithm of the Proposed Method

The aforesaid algorithm can be described as follows:

Step1. Calculate the Gabor feature representation of finger-knuckle-print image;
Step2. For every Gabor feature representation of image, extract the histogram feature by $LBP^{u2}_{8,1}$ in blocks.
Step3. Classify test samples by Chi Distance Statistic, get the recognition rate.

3 Experiments

We use the PolyU FKP database [7,13,14] to evaluate the performances of PCA, LDA, Gabor+PCA, Gabor+LDA, LBP and the propose method. The PolyU FKP database was collected from 165 volunteers, including 125 males and 40 females. Among them, 143 subjects are 20-30 years old and the others are 30-50 years old. The images were collected in two separate sessions. In each session, the subject was asked to provide 6 images for each of the left index finger, the left middle finger, the right index finger and the right middle finger. In total, the database contains 7,920 images from 660 different fingers. The original image size is 220*110. Some finger-knuckle-print images are shown in Fig. 4. In all the experiment, we do experiments on left index finger, left middle finger, right index finger and right middle finger respectively.

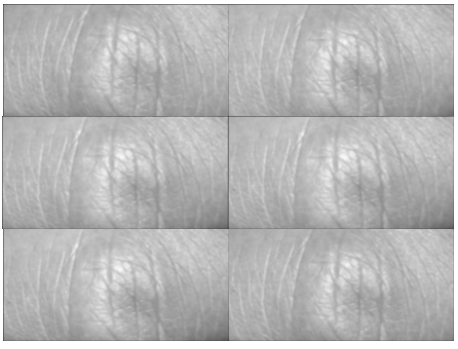


Fig. 4. Part of Finger-Knuckle-Print images in Ploy FKP

In the experiment, we resize the image size to 110*55 and used the first l (l=*5,6,7,8*) images per class for training and the remaining images for testing. In the PCA stage of PCA, LDA, we preserved nearly 90 percent image energy to select the number of principal components. Finally, a nearest neighbor classifier with cosine distance is employed. In LBP, we use Chi Distance Statistic for classification. The final recognition rates are shown in Table 1, Table 2, Table 3 and Table 4.

Table 1. Recognition Rates of Left index finger

	$l=5$	$l=6$	$l=7$	$l=8$
PCA	0.5974	0.5638	0.7459	0.8561
LDA	0.7255	0.7283	0.8691	0.9364
Gabor+PCA	0.9342	0.9253	0.9806	0.9879
Gabor+LDA	0.9437	0.9485	0.9891	0.9909
LBP	0.9100	0.9010	0.9709	0.9848
Proposed	**0.9394**	**0.9414**	**0.9903**	**0.9939**

Table 2. Recognition Rates of Left middle finger

	$l=5$	$l=6$	$l=7$	$l=8$
PCA	0.5827	0.5364	0.7467	0.8273
LDA	0.7143	0.7030	0.8655	0.9167
Gabor+PCA	0.9117	0.9101	0.9842	0.9924
Gabor+LDA	0.9238	0.9263	0.9903	0.9955
LBP	0.8952	0.8909	0.9770	0.9864
Proposed	**0.9411**	**0.9424**	**0.9976**	**0.9985**

From Table 1, Table 2, Table 3, Table 4, we can find that: (1) The proposed method has the top recognition rate; (2) LDA has better performance than PCA; (3) Gabor plus former methods has better performance than the corresponding methods; (4) The recognition rate of the middle finger is higher than that of the index finger.

Table 3. Recognition Rates of Right index finger

	$l=5$	$l=6$	$l=7$	$l=8$
PCA	0.6355	0.6051	0.7782	0.8500
LDA	0.7697	0.7606	0.8788	0.9303
Gabor+PCA	0.9584	0.9586	0.9927	0.9955
Gabor+LDA	0.9680	0.9626	0.9927	0.9985
LBP	0.9550	0.9556	0.9952	0.9955
Proposed	**0.9784**	**0.9727**	**0.9988**	**1.0000**

Table 4. Recognition Rates of Right middle finger

	$l=5$	$l=6$	$l=7$	$l=8$
PCA	0.6251	0.6010	0.7733	0.8621
LDA	0.7758	0.7525	0.8885	0.9242
Gabor+PCA	0.9299	0.9293	0.9867	0.9939
Gabor+LDA	0.9446	0.9323	0.9867	0.9985
LBP	0.9091	0.9121	0.9758	0.9939
Proposed	**0.9515**	**0.9475**	**0.9927**	**0.9970**

4 Conclusions

In this paper, we chose LGBP to identify finger-knuckle-print. First, we calculate the Gabor feature representation of the image; Second, for every Gabor feature representation, calculate LBP histogram feature corresponding. Third, use Chi Distance Statistic to get the classification rate. The experimental results show that our proposed method has a good performance.

Acknowledgments

This project is supported by NSF of China (90820009, 61005008), China Postdoctoral Science Foundation (20100471000).

References

1. Zhang, D.: Automated Biometrics: Technologies and Systems. Kluwer Academic, Dordrecht (2000)
2. Jain, A.K., Flynn, P., Ross, A.: Handbook of Biometrics. Springer, Heidelberg (2007)
3. E-channel System of the Hong Kong government, `http://www.immd.gov.hk/ehtml/20041216.htm`
4. Woodard, D.L., Flynn, P.J.: Finger surface as a biometric identifier, Computer Vision and Image Understanding, vol. 100, pp. 357–384 (2005)
5. Woodard, D.L., Flynn, P.J.: Personal identification utilizing finger surface features. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2005, vol. 2, pp. 1030–1036 (2005)

6. Ravikanth, C., Kumar, A.: Biometric Authentication using Finger-Back Surface. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2007, pp. 1–6 (2007)
7. Zhang, L., Zhang, L., Zhang, D., Zhu, H.: Online Finger-Knuckle-Print Verification for Personal Authentication, Pattern Recognition. Pattern Recognition 43(7), 2560–2571 (2010)
8. Zhang, W., Shan, S., Gao, W., Chen, X., Zhang, H.: Local Gabor Binary Pattern Histogram Sequence (LGBPHS): A Novel Non-Statistical Model for Face Representation and Recognition. In: ICCV (2005)
9. Zhang, B., Shan, S., Chen, X., Gao, W.: Histogram of gabor phase patterns (HGPP): a novel object representation approach for face recognition. IEEE Trans. Image Processing 16(1), 57–68 (2007)
10. Ojala, T., Pietikainen, M., Maeopaa, T.: Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. IEEE Trans. Pattern Analysis and Machine Intelligence 24, 971–987 (2002)
11. Ojala, T., Pietikainen, M., Harwood, D.: A Comparative Study of Texture Measures with Classification Based on Feature Distributions. Pattern Recognition 29, 51–59 (1996)
12. Ahonen, T., Hadid, A., Pietikainen, M.: Face Description with Local Binary Patterns: Application to Face Recognition. IEEE Trans. Pattern Analysis and Machine Intelligence 28, 2037–2041 (2006)
13. Zhang, L., Zhang, L., Zhang, D.: Finger-knuckle-print: a new biometric identifier. In: Proceedings of the IEEE International Conference on Image Processing (2009)
14. The Hong Kong PolyU FKP database, http://www4.comp.polyu.edu.hk/~biometrics/FKP.htm

Applying ICA and SVM to Mixture Control Chart Patterns Recognition in a Process

Chi-Jie Lu[1], Yuehjen E. Shao[2,*], and Chao-Liang Chang[2]

[1] Department of Industrial Engineering and Management, Ching Yun University, Jung-Li 320, Taoyuan, Taiwan, R.O.C.
jerrylu@cyu.edu.tw
[2] Department of Statistics and Information Science, Fu Jen Catholic University, Hsinchuang, Taipei County 242, Taiwan, R.O.C.
{stat1003,498726191}@mail.fju.edu.tw

Abstract. Mixture control chart patterns (CCPs) mixed by two types of basic CCPs together usually exist in the real manufacture process. However, most existing studies are considered to recognize the single abnormal CCPs. This study utilizes independent component analysis (ICA) and support vector machine (SVM) for recognizing mixture CCPs recognition in a process. The proposed scheme, firstly, uses ICA to the monitoring process data containing mixture patterns for generating independent components (ICs). The undetectable basic patterns of the mixture patterns can be revealed in the estimated ICs. The ICs are then used as the input variables of the SVM for building CCP recognition model. Experimental results revealed that the proposed scheme is promising for recognizing mixture control chart patterns in a process.

Keywords: Control chart pattern recognition, out-of-control process, independent component analysis, support vector machine.

1 Introduction

Control charts are one of the most popular tools used in statistical process control (SPC) and have been intensively used to monitor and improve the quality of manufacturing processes. A process is out-of-control when a data point falls outside the control limits or a series of data points exhibit unnatural/abnormal patterns [1]. Recognizing unnatural control chart patterns (CCPs) is an important issue in SPC since they can be associated with specific assignable causes adversely affecting the process. Six basic CCPs are commonly exhibited in control charts including normal (NOR), stratification (STA), systematic (SYS), cyclic (CYC), trend (TRE) and shift (SHI) [2][3]. Note that the stratification, systematic, cyclic, trend and shift patterns are called abnormal CCPs. The basic CCPs can happen in a process. Figure 1 shows these six basic control chart patterns.

* Corresponding author.

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 278–287, 2011.

(a) Normal (b) stratification (c) Systematic

(d) Cyclic (e) Trend (f) shift

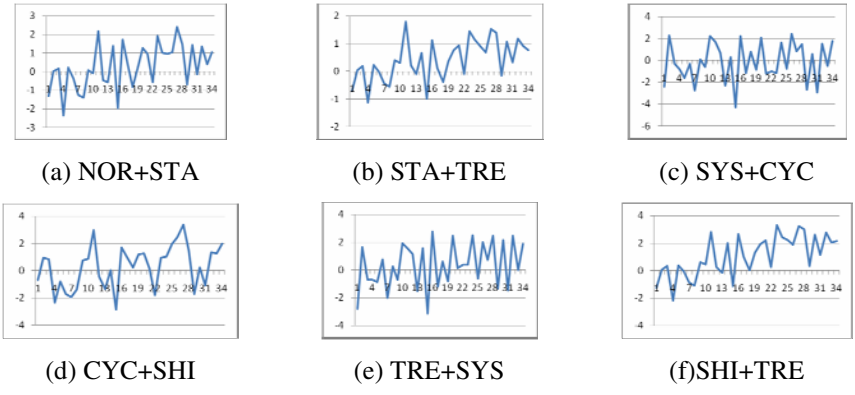
Fig. 1. Six basic control chart patterns

There have been many studies have been conducted on control chart pattern recognition [3][4][5]. Most of the existing studies were concerned with the recognition of the single abnormal control chart patterns (as shown in Figure 1) in a univariate process. Few researches have been reported on identifying CCPs in a process [6]. However, in most real control chart applications, the observed process data may be mixture patterns where two patterns may exist together. Without loss of generality, Figure 2 shows six mixture CCPs which are respectively mixed by two basic patterns. Compared to the patterns illustrated in Figure 1, it can be observed from Figure 2 that the mixture CCPs are more difficult to be recognized than the basic CCPs.

Only few literatures have been reported on the recognition of mixture process patterns. The work of [7] used the back-propagation neural network (BPN) to recognize mixture CCPs. The work of [8] integrated wavelet method and back-propagation neural network for on-line recognition of mixture CCPs. An efficient statistical correlation coefficient method for the recognition of mixture CCPs was proposed by [9]. However, the existing studies were proposed for recognizing mixture CCPs in a univariate process. Since, in modern manufacturing process, there are usually a number of quality characteristics that need to be simultaneously controlled, how to effective identify mixture CCPs in a process is an important and challenging task.

In this study, a control chart pattern recognition scheme by combining independent component analysis (ICA) and support vector machine (SVM) is proposed (called ICA-SVM scheme) for identifying mixture CCPs in a process. ICA is a novel feature extraction technique feature extraction technique to find independent sources given only observed data that are mixtures of the unknown sources, without any prior knowledge of the mixing mechanisms [10]. The independent sources, called independent components (ICs), are hidden information of the observable data. ICA has been employed successfully in various fields of multivariate data processing, from signal processing to time series prediction [10]. However, there are still few applications of using ICA in control chart pattern recognition. Lu et al. [11] integrated ICA, engineering process control and BPN to recognize shift and trend patterns in

correlated process. An ICA-based monitoring scheme to identify shift pattern in an autocorrelated process was studied by [12]. The combination of ICA and SVM for diagnosing mixture CCPs which are mixed by the normal and other abnormal basic patterns was investigated by [13]. However, their work did not consider the recognition of CCPs mixtured by any two basic patterns.



(a) NOR+STA (b) STA+TRE (c) SYS+CYC

(d) CYC+SHI (e) TRE+SYS (f)SHI+TRE

Fig. 2. Mixture CCPs: (a) Normal+Stratification (b) Systematic+Trend, (c) Systematic +Cyclic, (d) Cyclic+Shift, (e) Trend+ Systematic, (f) Shift+Trend

Support vector machine (SVM), based on statistical learning theory, is a novel neural network algorithm [14]. It can lead to great potential and superior performance in practical applications. This is largely due to the structure risk minimization principles in SVM, which has greater generalization ability and is superior to the empirical risk minimization principle as adopted in neural networks. The SVM has attracted the interest of researchers and has been applied many applications such as texture classification and data mining [14]. However, few studies have been conducted using SVM for CCP recognition [13].

The proposed ICA-SVM scheme first uses ICA to the observed process data contained mixture patterns for generating independent components. The estimated ICs are then served as the independent sources of the mixture patterns. The hidden basic patterns of the mixture patterns could be discovered in these ICs. The ICs are then used as the input variables of the SVM for building CCP recognition model. The rest of this paper is organized as follows. Section 2 gives brief overviews of ICA and SVM. The proposed model is described in Section 3. Section 4 presents the experimental results and this study is concluded in Section 5.

2 Methodology

2.1 Independent Component Analysis

In the basic conceptual framework of ICA algorithm [13], it is assumed that m measured variables, $\mathbf{x} = [x_1, x_2, \cdots, x_m]^T$ can be expressed as linear combinations of n unknown latent source components $\mathbf{s} = [s_1, s_2, \cdots, s_n]^T$:

$$\mathbf{x} = \sum_{j=1}^{n} \mathbf{a}_j s_j = \mathbf{A}\mathbf{s} \tag{1}$$

where $\mathbf{a}_j$ is the *j-th* row of unknown mixing matrix **A**. Here, we assume $m \geq n$ for A to be full rank matrix. The vector **s** is the latent source data that cannot be directly observed from the observed mixture data **x**. The ICA aims to estimate the latent source components **s** and unknown mixing matrix **A** from **x** with appropriate assumptions on the statistical properties of the source distribution. Thus, ICA model intents to find a de-mixing matrix **W** such that

$$\mathbf{y} = \sum_{j=1}^{n} \mathbf{w}_j x_j = \mathbf{W}\mathbf{x}, \tag{2}$$

where $\mathbf{y} = [y_1, y_2, \cdots, y_n]^T$ is the independent component vector. The elements of **y** must be statistically independent, and are called independent components (ICs). The ICs are used to estimate the source components s_j. The vector $\mathbf{w}_j$ in equation (2) is the *j-th* row of the de-mixing matrix **W**.

The ICA modeling is formulated as an optimization problem by setting up the measure of the independence of ICs as an objective function followed by using some optimization techniques for solving the de-mixing matrix W. Several existing algorithms can be used for performing ICA modeling [13]. In general, the ICs are obtained by using the de-mixing matrix **W** to multiply the observed data x, i.e. $\mathbf{y} = \mathbf{W}\mathbf{x}$. The de-mixing matrix **W** can be determined using an unsupervised learning algorithm with the objective of maximizing the statistical independence of ICs. The ICs with non-Gaussian distributions imply the statistical independence [13].

The ICA modeling is formulated as an optimization problem by setting up the measure of the independence of ICs as an objective function followed by using some optimization techniques for solving the de-mixing matrix **W**. Several existing algorithms can be used for performing ICA modeling [10]. In this study, the *FastICA* algorithm proposed by [10] is adopted in this paper.

2.2 Support Vector Machine

The basic idea of applying SVM to pattern recognition can be stated briefly as follows. We can initially map the input vectors into one feature space (possible with a higher dimension), either linearly or non-linearly, which is relevant with the selection of the kernel function. Then, within the feature space from the first, we seek an optimized linear division, that is, construct a hyperplane which separates two classes (this can be extended to multi-class).

A description of SVM algorithm is follows. Let $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, $\mathbf{x}_i \in R^d$, $y_i \in \{-1, 1\}$ be the training set with input vectors and labels. Here, *N* is the number of sample observations and *d* is the dimension of each observation, y_i is known target. The algorithm is to seek the hyperplane $\mathbf{w} \cdot \mathbf{x}_i + b = 0$, where **w** is the vector of hyperplane and *b* is a bias term, to separate the data from two classes with maximal

margin width $2/\|\mathbf{w}\|^2$, and the all points under the boundary is named support vector. In order to optimal the hyperplane that SVM was to solve the optimization problem was following [14].

$$\text{Min } \Phi(\mathbf{x}) = \frac{1}{2}\|\mathbf{w}\|^2 \tag{3}$$

$$\text{S.t. } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, i = 1, 2, ..., N$$

It is difficult to solve (3), and it can transform the optimization problem to be dual problem by Lagrange method. The value of α in the Lagrange method must be non-negative real coefficients. The (3) can be transformed into the following constrained form,

$$\text{Max } \Phi(\mathbf{w}, b, \xi, \alpha, \beta) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1, j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i{}^T \mathbf{x}_j \tag{4}$$

$$\text{S.t. } \sum_{j=1}^{N} \alpha_j y_j = 0$$

$$0 \leq \alpha_i \leq C, i = 1, 2, ..., N$$

In (4), C is the penalty factor and determines the degree of penalty assigned to an error. It can be viewed as a tuning parameter which can be used to control the trade-off between maximizing the margin and the classification error.

In general, it could not find the linear separate hyperplane in all application data. In the non-linear data, it must transform the original data to higher dimension of linear separate is the best solution. The higher dimension is called feature space, it improve the data separated by classification. The common kernel function are linear, polynomial, radial basis function (RBF) and sigmoid. Although several choices for the kernel function are available, the most widely used kernel unction is the RBF kernel defined as $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2), \gamma \geq 0$ [14], where γ denotes the width of the RBF. Thus, the RBF is applied in this study. The original SVM was designed for binary classifications. Constructing multi-class SVM is still an ongoing research issue. In this study, we used multi-class SVM method proposed by [15]. For more details, please refer to [15].

3 The Proposed ICA and SVM Model

Figure 3 shows the research scheme of the proposed ICA-SVM model. As shown in Figure3, the proposed scheme consists of two stages. In the training stage, the aim is to find the best parameter setting to train SVM model for CCP recognition. The first step of the training stage is to generate six basic CCPs as shown in Figure 1. Then, they are used as training sample to build SVM model. Since the RBF kernel function is adopted in this study, the performance of SVM is mainly affected by the setting of parameters of two parameters (C and γ). There are no general rules for the choice of

the parameters. In this study, the grid search proposed by [16] is used for parameters setting. The trained SVM model with best parameter set is preserved and used in the monitoring stage for CCP recognition.

In the monitoring stage, the first step is to collect two observed data from monitoring a process. Then, the ICA model is used to the observed data to estimate two ICs. Finally, the trained SVM is utilized to each IC to produce CCP recognition results.

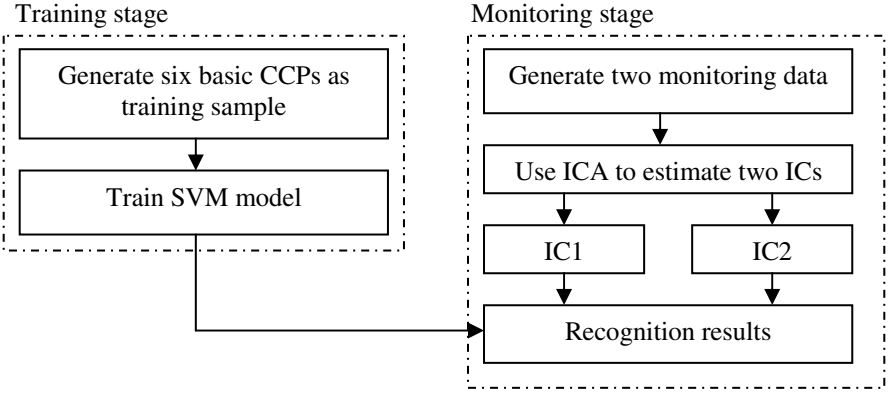


Fig. 3. The research scheme of the proposed ICA-SVM model

As an example, Figures 4(a1) and (a2) show two observed data collected from the monitoring a process. It is assumed that the data are mixed by systematic and cyclic patterns. Then, the ICA model is used to the data to generate two ICs which are illustrated in Figures 4(b1) and (b2). It can be found that Figures 4(b1) and (b2) can be used to represent systematic and cyclic patterns, respectively. For each IC, the trained SVM model is used to recognize the pattern exhibited in the IC. According to the SVM results, the process monitoring task is conducted to identify which basic patterns are exhibited in the process.

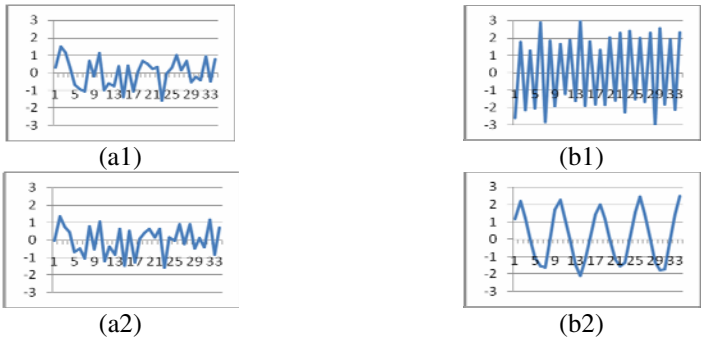


Fig. 4. (a1) and (a2) the observed data mixed by systematic and cyclic patterns; (b1) the IC represents systematic pattern; (b1) the IC represents cyclic pattern

4 Experimental Results

In this study, eight basic CCPs (as shown in Figure 1) and 21 mixture CCPs are used for training and testing the proposed ICA-SVM scheme, respectively. The eight basic patterns are generated using the same equations and values of different pattern parameters, as used by [3]. The parameters along with the equations used for simulating the CCPs are given in Table 1. The values of different parameters for abnormal patterns are randomly varied in a uniform manner between the limits. It is assumed that, in the current approach for pattern generation, all the patterns in an observation window are complete. The observation window used in this study is 34 data points.

Table 1. Parameters for simulating control chart patterns

Control chart patterns	Pattern equations	Pattern parameters
NOR	$x_i = u + r_i\sigma$	Mean(u)=0 Standard deviation(σ)=1
STA	$x_i = u + r_i\sigma'$	Random noise(σ')=(0.2(σ) to 0.4(σ))
SYS	$x_i = u + r_i\sigma + d(-1)^i$	Systematic departure(d)=(1(σ) to 3(σ))
CYC	$x_i = u + r_i\sigma + a\cdot\sin(2\pi i/t)$	Amplitude(a)=(1.5(σ) to 2.5(σ)) Period(t)=(8 and 16)
TRE	$x_i = u + r_i\sigma \pm ig$	Gradient(g)=(0.05(σ) to 0.1(σ))
SHI	$x_i = u + r_i\sigma \pm ks$ k=1 if i>P, else k=0	Shift magnitude(s)=(1.5(σ) to 2.5(σ)) Shift position(P)=(7, 13, 19)

Note: i = discrete time point at which the pattern is sampled (i= 1, ..., 34), r_i = random value of a standard normal variate at *i-th* time point, and x_i =sample value at *i-th* time point.

For generating 21 mixture patterns, the equation is defended as $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} \\ w_{13} & w_{14} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}$, where s_1 and s_2 represents two source patterns composed of basic CCPs, w_{ij} are mixing coefficients which are randomly generated between 0 and 1. In this study, without loss of generality, it is assumed that $\sum_j^2 w_{1j} = 1$ and $\sum_j^2 w_{2j} = 1$.

Features of observation window, such as mean, standard deviation, kurtosis and slop characteristics are adopted in literatures to improve the performance of CCP recognition [3]. However, features may ignore some useful information contained in the original data points of observation window since they are abstract information of the data points. Therefore, the proposed ICA-SVM model directly uses the 34 data pints of observation window as inputs of the SVM model. That is, there are 34 input variables are used in the proposed model for CCP recognition. In order to demonstrate the performance the proposed ICA-SVM scheme, the single SVM model without using ICA as preprocessing is constructed. It also directly uses the data pints of observation window as inputs.

After using the Grid search method to the two models, the best parameter sets for the ICA-SVM and single SVM models are ($C=2^3$, $\gamma=2^3$) and ($C=2^5$, $\gamma=2^1$), respectively. Note that the model selection details of the three models are omitted for saving space. The recognition results of the ICA-SVM and single SVM models are respectively illustrated in Tables 2 and 3.

From Tables 2-3, it can be found that the average correct classification rates of the proposed ICA-SVM model and the single SVM model are 86.09% and 78.77%, respectively. The proposed model outperforms the single SVM model in most testing mixture CCPs. Therefore, the proposed ICA-SVM scheme can effectively recognize control chart patterns mixtured by any two basic patterns in a process.

Examination of the recognition results in Table 2 reveals that the correct classification rate of SHI+SHI pattern is significantly lower than that of other patterns. It may because that the profile of the SHI pattern is difficult to be recognized from the original data points. There results suggest that inclusion of an additional feature that can distinguish between SHI and other CCPs can be very useful. Improving the recognition capabilities of the proposed ICA-SVM scheme are needed to be investigated in the future.

Table 2. Recognition results using the proposed ICA-SVM model

Mixture patterns	Correct Rate	Incorrect Rate	Mixture patterns	Correct Rate	Incorrect Rate
NOR+NOR	96.05%	3.95%	SYS+SYS	97.35%	2.65%
NOR+STA	99.35%	0.65%	SYS+CYC	81.83%	18.18%
NOR+SYS	98.20%	1.80%	SYS+ TRE	98.63%	1.38%
NOR+CYC	92.75%	7.25%	SYS+ SHI	56.63%	43.38%
NOR+TRE	98.10%	1.90%	CYC+CYC	65.25%	34.75%
NOR+SHI	100.00%	0.00%	CYC+TRE	81.80%	18.20%
STA+STA	98.85%	1.15%	CYC+SHI	57.00%	43.00%
STA+SYS	99.20%	0.80%	TRE+TRE	99.60%	0.40%
STA+CYC	99.28%	0.72%	TRE+SHI	56.68%	43.33%
STA+TRE	99.18%	0.82%	SHI+SHI	4.80%	95.20%
STA+ SHI	100.00%	0.00%	**Average**	**84.79%**	**15.21%**

Table 3. Recognition results using SVM model alone

Mixture patterns	Correct Rate	Incorrect Rate	Mixture patterns	Correct Rate	Incorrect Rate
NOR+NOR	97.95%	2.05%	SYS+SYS	97.60%	2.40%
NOR+STA	97.83%	2.18%	SYS+CYC	0.78%	99.23%
NOR+SYS	83.88%	16.13%	SYS+ TRE	100.00%	0.00%
NOR+CYC	88.20%	11.80%	SYS+ SHI	27.13%	72.88%
NOR+TRE	100.00%	0.00%	CYC+CYC	66.15%	33.85%
NOR+SHI	100.00%	0.00%	CYC+TRE	100.00%	0.00%
STA+STA	94.65%	5.35%	CYC+SHI	8.78%	91.23%
STA+SYS	94.58%	5.43%	TRE+TRE	100.00%	0.00%
STA+CYC	87.35%	12.65%	TRE+SHI	100.00%	0.00%
STA+TRE	100.00%	0.00%	SHI+SHI	9.30%	90.70%
STA+SHI	100.00%	0.00%	**Average**	**78.77%**	**21.23%**

5 Conclusion

Effective recognition of mixture CCPs in a process is an important and challenging task. In this study, a CCPs recognition scheme by integrating ICA and SVM is proposed. The proposed scheme, firstly, uses ICA to the mixture patterns to generate ICs. Then, the SVM model is used to each IC for pattern recognition. Twenty-one mixture CCPs are used in this study for evaluating the performance of the proposed method. Experimental results showed that the proposed ICA-SVM scheme outperforms the single SVM model without using ICA as preprocessing. According to the experimental results, it can be concluded that the proposed scheme can effectively recognize mixture control chart patterns in a process.

Acknowledgment. This work is partially supported by the National Science Council of the Republic of China, Grant No. NSC 99-2221-E-030-014-MY3. The author also gratefully acknowledges the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

1. Montgomery, D.C.: Introduction to statistical quality control. John Wiley & Sons, New York (2001)
2. Western Electric.: Statistical quality control handbook. Western Electric Company, Indianapolis (1958)
3. Gauri, S.K., Charkaborty, S.: Recognition of control chart patterns using improved selection of features. Computer & Industrial Engineering 56, 1577–1588 (2009)
4. Assaleh, K., Al-assaf, Y.: Feature extraction and analysis for classifying causable patterns in control charts. Computer & Industrial Engineering 49, 168–181 (2005)
5. Guh, R.S.: A hybrid learning-based model for on-line detection and analysis of control chart patterns. Computer & Industrial Engineering 49, 35–62 (2005)

6. El-Midany, T.T., El-Baz, M.A., Abd-Elwahed, M.S.: A proposed framework for control chart pattern recognition in multivariate process using artificial neural networks. Expert Systems with Application, 1035–1042 (2010)
7. Guh, R.S., Tannock, J.D.T.: Recognition of control chart concurrent patterns using a neural network approach. International Journal of Production Research 37(8), 1743–1765 (1999)
8. Chen, Z., Lu, S., Lam, S.: A hybrid system for SPC concurrent pattern recognition. Advanced Engineering Informatics 21, 303–310 (2007)
9. Yang, J.H., Yang, M.S.: A control chart pattern recognition scheme using a statistical correlation coefficient method. Computers & Industrial Engineering 48, 205–221 (2005)
10. Hyvärinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. John Wiley & Sons, New York (2001)
11. Lu, C.J., Wu, C.M., Keng, C.J., Chiu, C.C.: Integrated application of SPC/EPC/ICA and neural networks. International Journal of Production Research 46(4), 873–893 (2008)
12. Lu, C.J.: An independent component analysis-based disturbance separation scheme for statistical process monitoring. Journal of Intelligent Manufacturing (2010), doi: 10.1007/s10845-010-0394-3
13. Lu, C.J., Shao, Y.E., Li, P.H., Wang, Y.C.: Recognizing mixture control chart patterns with independent component analysis and support vector machine. In: Zhang, L., Lu, B.-L., Kwok, J. (eds.) ISNN 2010. LNCS, vol. 6064, pp. 426–431. Springer, Heidelberg (2010)
14. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, Berlin (2000)
15. Hsu, C.W., Lin, C.J.: A comparison of methods for multiclass support vector machines. IEEE Transactions on Neural Network 13, 415–425 (2002)
16. Hsu, C.W., Chang, C.C., Lin, C.J.: A practical guide to support vector classification. Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan (2003)

Gender Classification Using the Profile

Wankou Yang[1,2], Amrutha Sethuram[1], Eric Patternson[1],
Karl Ricanek[1], and Changyin Sun[2]

[1] Face Aging Group, Dept. Of Computer Science, UNCW, USA
[2] School of Automation, Southeast University, Nanjing 210096, China
Wankou_yang@yahoo.com.cn, {sethurama,patternsone,
ricannekk}@uncw.edu, cysun@seu.edu.cn

Abstract. Gender classification has attracted a lot of attention in computer vision and pattern recognition. In this paper, we propose a gender classification method. First, we present a robust profile extraction algorithm; Second, we implement Principal Components Analysis (PCA) and Independent Components Analysis (ICA) to extract discriminative features from profile to estimate the face gender via SVM. Our experimental results on Bosphorus 3D face database show that our proposed method works well.

Keywords: PCA, ICA, gender classification, feature extraction.

1 Introduction

Human faces contain important information, such as gender, race, mood, and age [1, 2]. In the area of human-computer interaction, there are both commercial and security interests to develop a reliable gender classification system from a good or low quality images. Gender perception and classification have been studied extensively from psychological prospect [3,4], which show that gender has close relationships with both 2D information and 3D shape [5,6].

Wild et al. showed that gender classification achieved much lower recognition rate for children's face than the ones of adults [4]. Moghaddam and Yang [7] developed a robust gender classification system based on RBF-kernel SVM on a set of FERET raw images, and concluded that the nonlinear SVM outperformed the traditional pattern classifiers on gender classification problems. There are many publications on gender classification by using FERET database, such as [2,8] etc. Recently, Yang et al. investigated three gender classification algorithms (SVM, FLD and Real Adaboost) with three different preprocessing methods on a large Chinese database and achieved a good accuracy [9]. Baluja and Rowley [3] studied a method based on an Adaboost for classification from low resolution grayscale face images. Gao and Ai [] proposed using Active Shape Model (ASM) for face representation and using probabilistic boosting trees approach for gender classification on a set of multiethnic faces. Guo et al. [10] studied the aging effect on gender classification and showed that the gender classification accuracy on young and senior faces can be much lower than the one on adults'

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 288–295, 2011.

faces, and hence concluded that the age of a person affected the gender recognition significantly. Wang et al. [11] gave a robust gender classification system via model selection. Erno et al. [2] gave a systematic study on gender classification with face diction and face alignment.

Currently most of research works about gender are based on 2D images and only a few studies have research gender based on 3D shapes information, which is due to the expensive price of 3D sensors and the high calculation complexity of 3D data. Alice et al. studied the roles of shape and texture information in gender classification [12]. X. Lu et al. presented a method by combing the registered range and intensity images for gender classification via SVM [13]. J. Wu et al. proposed a weighted PGA and supervised PGA to parameterize the facial needle-maps and compared their performances with PGA for gender classification [14]. Y. Hu et al. gave a 3D facial gender classification by fuse the classification results of the facial regions [15].

Inspired by the success of the facial profile in 3D face recognition, we propose a method for gender classification, named ICProfile, which is based on 3D facial profile, Independent Components Analysis (ICA) and SVM.

2 Related Knowledge

2.1 Independent Components Analysis

Bartlett et al [16] proposed two architectures for ICA. Here we use the architecture I. Denote by x a p-dimensional vector, the ICA of x seeks for a sequence of projection vectors $w_1, w_2, \cdots, w_q (q < p)$ to maximize the statistical independence of the projected data. It can be expressed as follows: $s = W^T x$ (1) where s denotes the ICs of x and $W = [w_1, \cdots, w_q]$ is called the unmixing matrix. Various criteria, such as those based on mutual information, negentropy and higher-order cumulants, have been proposed for computing W [17]. Among them the FastICA algorithm has been widely used in pattern recognition [17,18]. Usually, principal components analysis (PCA) is implemented to whiten the data and reduce the dimensionality before applying ICA.

2.2 SVM

SVM [19] is a supervised learning technique from the field of machine learning and is applicable to both classification and regression. The basic training principal behind SVM is finding the optimal separating hyperplane that separates the positive and negative samples with maximal margin. Based on this principal, a linear SVM use a systematic approach to find a linear function with the lowest VC dimension. For linearly non-separable data, SVM can map the input to a high dimensional feature space where a linear hyperplane can be found. SVM has been successfully used in gender classification and age estimation [7].

3 Our Proposed Method for Gender Classification

Facial profile has been successfully applied in 3D face recognition [20]. So it is nature to classify gender based on the 3D facial profile. In the proposed method, first, we extract the profile; second, we normalize the profile; third, we extract discriminative feature from the normalized profile and classify gender via SVM.

3.1 Pre-process and Profile Extraction Algorithm

In this paper, we evaluate the performance of the proposed method on Bosphorus 3D face database. The presented 3D face point in Bosphorus 3D face database [22] is a raw 3D point cloud. Some 3D point clouds and the corresponding 2D images in bosphorus face database are shown in Fig 1. Here we do some pre-processing work to extract the profile. The pre-processing and profile extraction framework is shown in Fig 2. In this paper, we present a 3D face as a n*3 matrix of x, y, z-coordinates of the point cloud (i=1,2,…,n, where n is the point number).

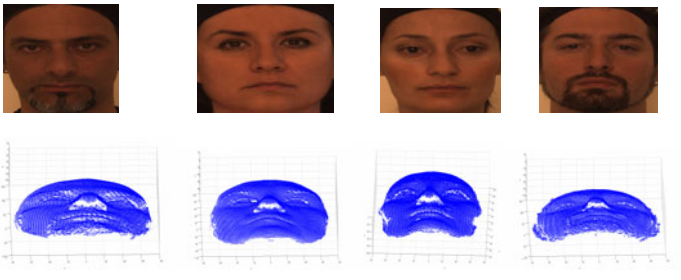


Fig. 1. Some images and point clouds in Bosphorus 3D face database

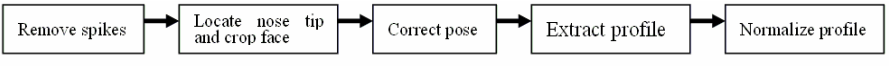
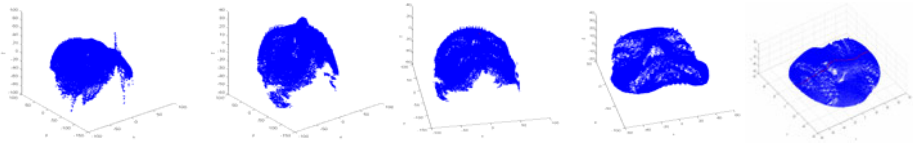


Fig. 2. Framework of pre-processing and profile extraction

First, we remove the spike (outliers) since the spikes influence the calculation the nose tip location. We stat the neighbors in 5mm distance in x-y plane and 1mm distance in z-direction. And we remove the 5 percent points with the fewest neighbor number, which are statistically judged as spikes. Fig 3(a) shows a point cloud with spikes, Fig 3(b) shows the point cloud after spike removing.

Second, we choose the point with largest z-coordinate as the nose tip since the faces in Bosphorus are almost front faces. Fig 3(c) shows a nose tip of one point cloud.

Third, we use a sphere of radius r centered at the nose tip to crop the face. In our experiments we set $r=80$. Fig 3(d) shows a cropped face.



a) b) c) d) e)

Fig. 3. An example of pre-processing and the extraction of the profile

Forth, we use PCA transformation to correct the pose of the cropped face to get the rotation matrix R and the translation matrix T [21]. We use matrix R and T to correct the pose of P. Let P is $3*n$ matrix of x, y, z coordinates of the point cloud of a face.

$$P = \begin{bmatrix} x_1 & & x_n \\ y_1 & \ddots & y_n \\ z_1 & & z_n \end{bmatrix} \tag{1}$$

The total scatter matrix R and the eigenvectors and eigenvalues of R are described as follows:

$$R = \frac{1}{n}\sum_{k=1}^{n}(P_k - m)(P_k - m)^T,\ m = \frac{1}{n}\sum_{k=1}^{n} P_k \tag{2}$$

$$R \bullet w_i = \lambda_i \bullet w_i \quad (i = 1,2,3,\ \lambda_1 > \lambda_2 > \lambda_3)$$

where P_k is the kth column of P. w_1 denotes the direction of the largest deviation of P and w_3 denotes the direction of the smallest deviation of P. So, w_1 denotes the profile direction (y-coordinate direction) and w_3 denotes the z-coordinate direction, w_2 denotes the x-coordinate direction. So far, we get the rotation matrix R and the translation matrix T.

Fifth, we correct the pose of the point cloud P according to R and T. Fig 3(e) shows a correct point cloud.

Sixth, we choose the intersection between P and the y-z plane as the profile. Fig 3(e) shows the extracted corresponding profile. Finally, we only choose z-coordinate vectors to denote the final extracted profile.

So far, we give the complete algorithm to extract the profile. When we obtain the profile, the first step of the normalization process is to resizes the vectors to a fixed length vectors, and the second step is to normalize the vectors to have a minimum 0 and a maximum 255. Fig 4(a) shows some normalized results of the whole profiles. Fig 4(b) shows some normalized results of the upper profiles. From Fig 3(a), we can find that mouth parts of a profile are easy to be influenced by the expression. We use $\{a_t^k\}$ to denote the normalized profile.

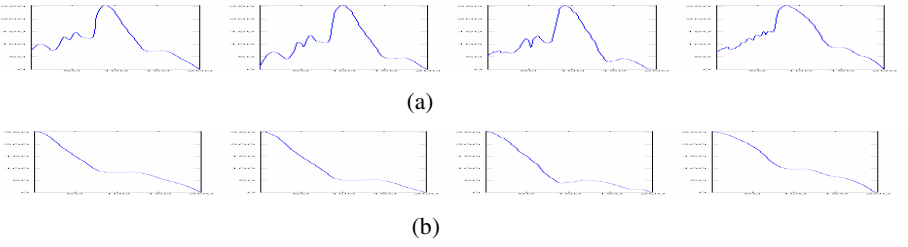


Fig. 4. Some normalized profile

3.2 Feature Extraction and Classify

We compute the independent components $s=[s_1,s_2,\cdots,s_q]$ and principal ocmponents $w=[w_1,w_2,\cdots,w_d]$ of $\{a_t^k\}$ by FastICA [18] and PCA []. Here FastICA uses a contrast function $G(u)=(1/4)u^4$. By projecting a onto s and w, we get the feature vectors $v(v=s^Ta)$ and $v(v=w^Ta)$ of the wrist-pulse $\{a_t^k\}$, respectively.

After transformations by ICA and PCA, feature vectors are obtained for each profile. Then, a SVM classifier is used for classification. Fig 5 shows the gender classification framework.

Fig. 5. Gender classification framework

3.3 Experiment Setups

The Bosphorus 3D face database [22] consists of 105 people in various pose, expression and occlusion. The majority of the subjects are aged between 25 and 35. There are 60 men and 45 women in total, and most of the subjects are Caucasian. Up to 54 face scans are available per subject, but 34 of these subjects have 31 scans. Thus, the number of the total face scans is 4641. Each scans only contains the main face region. In the experiments, we only choose a subset with nature expression, totally 290 scans.

In the feature extraction step, we compare the performances of PCA and ICA.

In PCA, we set the final dimension as 120. In ICA, we set the final dimension as 120.

We use SVM as the gender classifier. We perform a standard 10-fold cross validation to evaluate the prediction error of the proposed method. We use the contributed package “Libsvm” [15] in Matlab for the computation of SVR (svm_type: EPSILON_SVR, kernel_type: RBF Kernel, gamma=0.25, C=64).

3.4 Experimental Results

In the experiment, we compare three normalization methods with no-scaling on either PCA feature or ICA feature for gender classification. The experiment results are shown in Table 1. Fig. 6(a) and Fig. 6(b) show the recognition rates with different feature dimensionality on the whole profiles. Fig. 6(c) and Fig. 6(d) show the recognition rates with different feature dimensionality on the upper profiles. For ICA and PCA features, no-scaling turns out to achieve the best MAE, comparing to the rest normalization methods. Here we compare two normalization methods: Min-Max and Z-score. Note that the Min-Max-[0, 1] with distinct hyper-parameters for SVM.

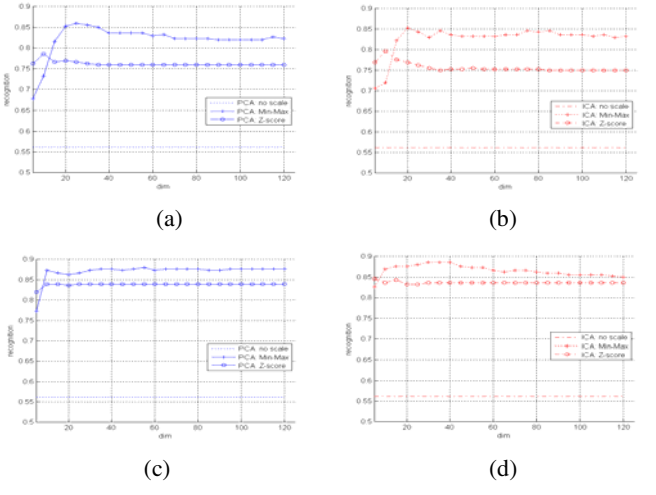


Fig. 6. PCA / ICA recognition vs. dimension

Table 1. Recognition rates on PCA and ICA features, respectively

	PCA^1	PCA^2	PCA^3	ICA^1	ICA^2	ICA^3
$Mean^1$	0.5619	**0.8595**	0.7860	0.5619	**0.8528**	0.7960
Std^1	0.0956	**0.0958**	0.1037	0.0956	**0.1037**	0.1031
$Mean^2$	0.5619	**0.8796**	0.8395	0.5619	**0.8863**	0.8462
Std^2	0.0956	**0.1070**	0.0795	0.0956	**0.0760**	0.1058

In Table 1, PCA/ICA superscript 1 means no scaling, PCA/ICA superscript 2 denotes Min-Max-[0 1], PCA/ICA superscript 3 denotes Z-score to normalize the features. Mean/Std superscript 1 denotes results on the whole profiles, and Mean/Std superscript 2 denotes results on the upper profiles. From Table 1, we can find that: 1)The upper profiles have better performance than the whole profiles since the lower profiles are easy noised by the expression and the upper profiles are more robust. 2) ICA has better performance than PCA.

4 Conclusions

In this paper, we give a roust method to extract 3D face profile, then propose a method to automatic estimate the face gender via SVM, which uses the 3D facial profile and subspace learning methods. The experimental results on the Bosporus 3D face database show that our proposed method has a good performance. In the future work, we will research supervised method to extract feature and feature selection technologies to further improve the performance of the gender classification system.

Acknowledgments

This work is supported by the Intelligence Advanced Research Projects Activity, Federal Bureau of Investigation, and the Biometrics Task Force. The opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of our sponsors.

This work is also supported by NSF of China (90820009, 61005008, 60803049, 60875010), Program for New Century Excellent Talents in University Of China (NCET-08-0106), China Postdoctoral Science Foundation (20100471000) and the Fundamental Research Funds for the Central Universities (2010B10014).

References

1. Albert, A.M., Ricanek, K., Patterson, E.: A review of the literature on the aging adult skull and face: Implications for forensic science research and applications. Forensic Science International 172, 1–9 (2007)
2. Makinen, E., Raisamo, R.: Evaluation of gender classification methods with automatically detected and aligned face. Trans. Pattern Anal. Mach. Intell. 30(3), 541–547 (2008)
3. Baluja, S., Rowley, H.: Boosting sex identification performance. IJCV 71(1), 111–119 (2007)
4. Wild, H.A., Barrett, S.E., Spence, M.J., et al.: Recognition and sex categorization of adults' and children's face: examining performance in the absence of sexstereotyped cues. J. off Exp. Child Psychology 77, 269–291 (2000)
5. Brucelf, V., et al.: Sex discrimination: how do we tell the difference between male and female face? Perception 22, 131–152 (1993)
6. Otoole, A., et al.: Sex classification is better with three dimensional head structure than with image intensity information. Perception 26, 75–84 (1997)
7. Yang, M.H., Moghaddam, B.: Gender classification using support vector machines. In: ICIP 2000, vol. 2, pp. 471–474 (2000)
8. Gutta, S., Wechsler, H.: Gender and ethnic classifications of human faces using hybrid classifiers. In: Proceedings 1999 International Joint Conference on Neural Networks, pp. 4084–4089 (1999)
9. Yang, Z., Li, M., Ai, H.: An experimental study on automatic face gender classification. In: ICPR 2006, pp. 1099–1102 (2006)

10. Gao, W., Ai, H.: Face gender classification on consumer images in a multiethnic environment. In: Tistarelli, M., Nixon, M.S. (eds.) ICB 2009. LNCS, vol. 5558, pp. 169–178. Springer, Heidelberg (2009)
11. Wang, Y., Ricanek, K., Chen, C., Chang, Y.: Gender classification from infants to seniors. In: BTAS 2010 (2010)
12. O'Toole, A.J., et al.: The perception of face gender, the role of shape and texture information in sex classification. Technical Report No. 23 (December 10, 1995)
13. Lu, X., Chen, H., Jain, A.K.: Multimodal Facial Gender and Ethnicity Identification. In: Zhang, D., Jain, A.K. (eds.) ICB 2005. LNCS, vol. 3832, pp. 554–561. Springer, Heidelberg (2005)
14. Wu, J., et al.: Gender classification using shape from shading. In: BMVC 2007, pp. 499–508 (2007)
15. Hu, Y., Yan, J., Shi, P.: A fusion based method for 3D facial gender classification. In: ICCAE 2010, vol. 5, pp. 369–372 (2010)
16. Bartlett, M.S., Movellan, J.R., Sejnowski, T.J.: Face recognition by independent component analysis. IEEE Trans. Neural Network 13(6), 1450–1462 (2002)
17. Hyvärinen, A., Karhunen, J., Oja, E.: Independent component analysis. Wiley, New York (2001)
18. Kim, J., Choi, J.M., Yi, J., Turk, M.: Effective representation using ICA for face recognition robust to local distortion and partial occlusion. IEEE Trans. Pattern Anal. Mach. Intell. 26(1), 131–137 (2004)
19. Vapnik, V.N.: The nature of statistical learning theory (Spring 2000)
20. Li, X., Da, F.: Robust 3D Face Recognition Based on Rejection and Adaptive Region Selection. In: Zha, H., Taniguchi, R.-i., Maybank, S. (eds.) ACCV 2009. LNCS, vol. 5996, pp. 581–590. Springer, Heidelberg (2010)
21. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. A wiley-Interscience Publication, Hoboken (2001)
22. Savran, A., Alyüz, N., Dibeklioğlu, H., Çeliktutan, O., Gökberk, B., Sankur, B., Akarun, L.: Bosphorus Database for 3D Face Analysis. In: Schouten, B., Juul, N.C., Drygajlo, A., Tistarelli, M. (eds.) BIOID 2008. LNCS, vol. 5372, pp. 47–56. Springer, Heidelberg (2008)

Face Recognition Based on Gabor Enhanced Marginal Fisher Model and Error Correction SVM

Yun Xing, Qingshan Yang, and Chengan Guo*

School of Information and Communication Engineering,
Dalian University of Technology, Dalian, Liaoning 116023, China
{xingyun1985,qsyang279}@gmail.com, cguo@dlut.edu.cn

Abstract. In previous work, we proposed the Gabor manifold learning method for feature extraction in face recognition, which combines Gabor filtering with Marginal Fisher Analysis (MFA), and obtained better classification result than conventional subspace analysis methods. In this paper we propose an Enhanced Marginal Fisher Model (EMFM), to improve the performance by selecting eigenvalues in standard MFA procedure, and further combine Gabor filtering and EMFM as Gabor-based Enhanced Marginal Fisher Model (GEMFM) for feature extraction. The GEMFM method has better generalization ability for testing data, and therefore is more capable for the task of feature extraction in face recognition. Then, the GEMFM method is integrated with the error correction SVM classifier to form a new face recognition system. We performed comparative experiments of various face recognition approaches on the ORL, AR and FERET databases. Experimental results show the superiority of the GEMFM features and the new recognition system.

Keywords: Face recognition, Gabor wavelets, Marginal Fisher analysis, Manifold learning, Error correction SVM.

1 Introduction

Face recognition is one of the most challenging research topics in computer vision and machine learning [1]. Two issues are essential in face recognition: the first is to use what features to represent a face that will be more robust for the variations of illumination, express, and pose etc. The second is how to design an efficient classifier to fulfill the discriminative ability of the features. Both the representation and classification methods are very important for face recognition, and an optimization combination of them usually brings better recognition performance.

Face images have a relative high dimensionality, but they usually lie on a lower dimensional subspace or sub-manifold. Thus, subspace learning and manifold learning methods have been broadly studied in face recognition. Eigenface [2] and Fisherface [3] are two typical subspace learning methods. Eigenface method computes the principal vectors from training set and represents each face image as the

* Corresponding author.

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 296–304, 2011.

coefficients of the small set of characteristic facial images. Fisherface method extracts the discriminative information by maximizing the between-class scatter matrix, while minimizing the within-class scatter matrix in the projective subspace. However, both Eigenface and Fisherface methods fail to reveal the underlying structure nonlinearly embedded in high-dimensional space. Hence, the manifold learning methods have been proposed to overcome this problem, e.g., ISOMAP [4], LLE [5], and Laplacian Eigenmap [6]. These manifold learning-based methods have the ability to find the intrinsic structure. Therefore they are superior and more powerful methods than the traditional ones. In [7], we presented a hybrid feature extraction method named Gabor-based Marginal Fisher Analysis (GMFA) for face recognition by combining Gabor filtering with manifold learning method MFA. The GMFA method applies the MFA [8] to the augmented Gabor feature vectors derived from the Gabor wavelet representation of face images, and has been proved as an efficient method for feature extraction. However, the overfitting problem in GMFA method would cause degradation in the generalization ability. The MFA procedure is equivalent to two operations: firstly whitening the intra-class similarity matrix, and then applying PCA on the inter-class difference matrix with the transformed data. While the intra-class similarity matrix tends to capture noise, which causes the transformed inter-class difference matrix to fit for misleading variations, and thereby overfitting occurs. We propose in this paper an improved MFA method, namely Enhanced Marginal Fisher Model (EMFM), to reduce the adverse effect of overfitting by selecting eigenvalues in MFA process. We then further combine Gabor filtering with EMFM method as GEMFM to achieve better separability.

As in any pattern classification task, classifier also plays an important role in face recognition process. The Support Vector Machine (SVM) is an optimal classifier in term of structural risk minimization based on VC theory [9], and has been widely and successfully applied in pattern recognition. However, the SVM was originally designed for binary classification. For multi-class classification problem, it must be realized by a suitable combination of a number of binary SVMs. Several methods have been proposed to solve the multi-class classification problem using binary SVMs, including the M-ary algorithm [10], the One-against-one [11], the One-against-the-others [10], and the error correction SVM [12]. The error correction SVM has the error control ability that can correct a certain number of intermediate misclassifications by training some extra SVMs, and it has been applied to face recognition successfully in our previous work [12]. Based on the GEMFM and the error correction SVM classifier, a new face recognition system is proposed in this paper. Many simulation experiments have been conducted using the ORL, AR and FERET databases in the paper. Experimental results show the superiority of the GEMFM features and the new recognition method.

The rest of the paper is organized as follows: Section 2 describes the new face recognition method in detail, including the Gabor wavelets filtering algorithm, the Gabor based enhanced marginal fisher model, and the error correction SVM classifier. Section 3 gives the application results of the new method to face recognition. A summary and further research direction are given in Section 5.

2 Face Recognition Method Based on GEMFM and SVM

A new method is proposed here by combining the Gabor-based Enhanced Marginal Fisher Model (GEMFM) and the error correction SVM classifier. In the method, the images are first filtered by Gabor wavelets in order to capture salient visual properties such as spatial localization, orientation selectivity, and spatial frequency characteristic. Then, the high-dimensional Gabor representation of the image is processed by the Enhanced Marginal Fisher Model to find the underlying structure and extract low-dimensional features. Finally, the Gabor-based EMFM (GEMFM) feature is input into the error correction SVM classifier to obtain classification result.

2.1 Gabor Feature Representation

The 2-D Gabor wavelets, which can extract the desirable local features at multiple scales and orientations from face images, have been widely and successfully used in face recognition. The Gabor wavelets can be defined as follows [13]:

$$\psi_{\mu,v}(z) = \frac{\left\| k_{\mu,v} \right\|^2}{\sigma^2} e^{-\frac{\left\| k_{\mu,v} \right\|^2 \left\| z \right\|^2}{2\sigma^2}} [e^{ik_{\mu,v} \cdot z} - e^{-\frac{\sigma^2}{2}}], \tag{1}$$

where $z = [x, y]^T$, $k_{\mu,v} = [k_v \cos\phi_\mu, k_v \sin\phi_\mu]^T$, v and μ define the scale and orientation of the Gabor kernels, $k_v = k_{\max} / f^v$, $\phi_\mu = \mu\pi/8$, and f is the spacing factor between kernels in frequency domain. We select the parameters in accordance with [13].

The Gabor wavelet representation of an image is the convolution of the image with the family of Gabor kernels of (1):

$$O_{\mu,v}(z) = I(z) * \psi_{\mu,v}(z), \tag{2}$$

where *I(z)* is the gray level distribution of an image, "$*$" denotes the convolution operator, and $O_{\mu,v}(z)$ denotes the convolution result corresponding to the Gabor kernel at scale v and orientation μ. The augmented Gabor feature of the image could be obtained by concatenating all $O_{\mu,v}(z)$.

Since the Gabor features are extracted from the local regions of face images, they are less sensitive to variations of illumination, expression and pose, etc, which can usually improve the recognition accuracy. However, the augmented Gabor features suffer from the curse of dimensionality, and the dimension needs to be reduced to make the recognition progress computationally feasible. Next, the EMFM will be proposed to perform dimension reduction for the augmented Gabor features.

2.2 Gabor-Based Enhanced Marginal Fisher Model

The feature dimension of the Gabor filtering is usually immense, and here we first down-sample the Gabor feature by a 4×4 window, and conduct PCA transformation to further reduce the dimension of Gabor feature. Suppose we obtain from the above methods N data points $X = [x_1, x_2, \cdots, x_N] \in R^{D\times N}$ that can be divided into C classes, where N is the sample number and D is the PCA feature dimension. However we

cannot employ X directly to the classification before dimensionality reduction, since D is still too high for the classifier, and meanwhile the data points X do not have a favorable separability.

Marginal Fisher Analysis [7-8] has been proved to be an effective and desirable method for dimensionality reduction. Using the graph embedding framework, MFA designs an intra-class compactness graph $G^c = \{X, W^c\}$ as intrinsic graph, and an inter-class separability graph $G^p = \{X, W^p\}$ as penalty graph, where W^c and W^p are similarity matrices. For each sample $x_i \in X$, set $W_{ij}^c = W_{ji}^c = 1$ if x_j is among the k_1-nearest neighbors of x_i in the same class, and set $W_{ij}^c = W_{ji}^c = 0$ otherwise. For each class c, set $W_{ij}^p = W_{ij}^p = 1$ if the pair (x_i, x_j) is among the k_2 shortest pairs among different classes and set $W_{ij}^p = W_{ij}^p = 0$ otherwise. The procedure of Marginal Fisher Analysis algorithm can be summarized as follows:

Step1: Construct the intra-class compactness and inter-class separability graphs by setting the similarity matrices W^c and W^p. The geometrical explanation of neighborhood relation of MFA is given in Fig. 1.

Step2: Find the optimal projection direction by the Marginal Fisher Criterion:

$$w_{MFA} = \arg\min \frac{w^T X(D^c - W^c)X^T w}{w^T X(D^p - W^p)X^T w} \tag{3}$$

where diagonal matrix D^c and D^p are defined by

$$D_{ii}^c = \sum_{j \neq i} W_{ij}^c, \quad D_{ii}^p = \sum_{j \neq i} W_{ij}^p, \quad \forall i . \tag{4}$$

Step 3: Project the high dimensional data point x into lower dimensional space via linear projection:

$$y = w_{MFA}^T x . \tag{5}$$

The MFA method does overcome one limitation of LDA: the data of each class no longer has to be a Gaussian distribution. Therefore MFA is a more general algorithm with a better discriminative ability. Nevertheless, its performance can be further improved.

Inspired by [13], we propose herein an improved MFA method, namely Enhanced Marginal Fisher Model (EMFM). Noticing that the methods combining PCA and LDA could lack in generalization ability due to overfitting to the training data, MFA method does as well have the same problem.

The MFA procedure which involves simultaneous diagonalization of the intra-class similarity matrix $X(D-W)X^T$ and the inter-class difference matrix $X\left(D^p - W^p\right)X^T$ is equivalent to two operations: firstly whitening the intra-class similarity matrix, and then applying PCA on the inter-class difference matrix with the transformed data. Here using too many principal components (PCs) would lead to lower recognition rate. Smaller eigenvalues of the intra-class compactness matrix usually correspond to high-frequency components, which contain much noise that degrades the separability

of data, and if these eigenvalues are brought to describe the PCA subspace, the MFA procedure has to fit for those misleading information, which eventually leads to over-fitting. As shown in Fig.2, although the trailing eigenvalues are rather trivial, the misguiding information should not be ignored.

The EMFM method overcomes overfitting by selecting eigenvalues of the intra-class similarity matrix:

$$\left(X(D-W)X^T\right)U = U\Gamma, U^TU = I. \tag{6}$$

Select s out of m eigenvectors of U corresponding to s largest eigenvalues in the decreasing order $\gamma_1 \geq \gamma_2 \geq ... \geq \gamma_s$, we have

$$U_s = [u_1, u_2, \cdots, u_s],\ \Gamma_s = diag\{\gamma_1, \gamma_2, \cdots, \gamma_s\}. \tag{7}$$

The new inter-class difference matrix in R^s becomes

$$S_p = (\Gamma_s)^{-1/2}(U_s)^T\left(X(D^P - W^P)X^T\right)\cdot U_s \cdot (\Gamma_s)^{-1/2}. \tag{8}$$

Diagonalize S_p by

$$S_p\Theta = \Theta\Lambda,\ \ \Theta^T\Theta = I. \tag{9}$$

The transform matrix of EMFM can be obtained by

$$T_{EMFM} = U_s \cdot (\Gamma_s)^{-1/2} \cdot \Theta. \tag{10}$$

Thus the inter-class difference matrix can be described by more "purified" data.

Finally the EMFM feature vector y is given by

$$y = (T_{EMFM})^T x\ . \tag{11}$$

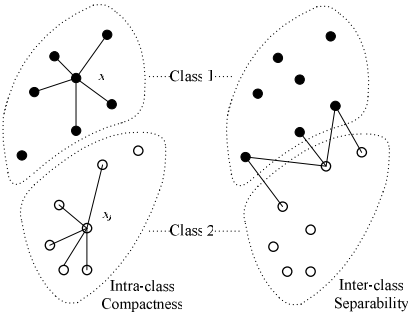


Fig. 1. Neighborhood Graph for MFA

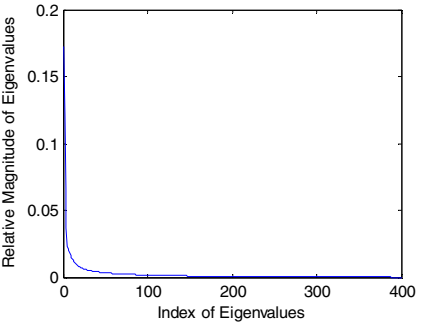


Fig. 2. Relative Magnitude of Eigenvalues in whitening step of MFA

2.3 Error Correction SVM Classifier

The SVM is originally designed for binary classification. For multi-class classification problems, a number of SVMs are required to achieve the classification task. For an m-class classification problem, k binary SVMs, where $k = \lceil \log_2 m \rceil$, is enough in theory for classifying the m classes. However, the classifier with this number of SVMs has no error tolerance, since if one SVM gives a wrong intermediate classification, the final classification result of the SVM-based classifier will be incorrect. This problem can be effectively solved by the error correction SVM algorithm [12].

There are two stages for implementing the error correction SVM classifier: the training (or encoding) stage and the classification (or decoding) stage. For the training stage, the first step is to generate an n-bit BCH code according to the predefined error correction ability; the second step is to assign a unique codeword of the code to each class of the training samples; and the final step is to construct n training sets and use each training set to trained a binary SVM classifier.

For the classification (or decoding) stage, the n trained SVMs are used to classify new samples. In this stage, the first step is to input a sample into each trained SVM to get a binary output, in which n binary values can be obtained. The second step is to construct a codeword using the n binary values in the same way as in the encoding stage, and then decode the codeword to get the possible errors in the codeword corrected by using the error correction algorithm. The final step is to classify the sample into the class denoted by the decoded codeword.

3 Experiments

To verify the effectiveness of the proposed approach, we conducted experiments on three different face databases: ORL, AR and FERET. The feature extraction method based on Gabor Enhanced Marginal Fisher Model (GEMFM) is compared with several classic sub-space learning methods and manifold learning methods, and the error correction SVM classifier is compared with the nearest neighbor classifier and One-Against-One SVM classifier.

Example 1 - Experiment on the ORL Database

The ORL database contains 400 images of 40 people with 10 images for each person. In the experiment, all images are resized to 46×56 pixels. Since there are 40 classes of face images in the database, the BCH (31, 6) code that has 31 bits in total with 6 information bits and the minimum Hamming distance 15 is chosen for the error correction SVM algorithm. Correspondingly, there are 31 SVMs in total for the face recognition problem and up to 7 errors can be corrected by the error correction SVM classifier. For each simulation experiment, 200 samples (5 samples selected randomly for each person) are used as the training set, and the remaining 200 samples are used as the testing set. We conduct the simulation experiments 20 times, and take the average recognition rate as the final result. Table 1 gives the recognition rates for different methods implemented in the experiment on the ORL database.

Table 1. Recognition rates (%) tested on the ORL database

Feature extractor	*Nearest Neighbor*	*One-Against-One*	*Error Correction SVM*
LPP	94.33	95.53	95.45
EFM	95.28	96.05	96.40
MFA	95.48	96.45	96.60
EMFA	95.59	96.96	97.05
Gabor+LPP	98.08	98.13	98.25
Gabor+EFM	98.42	98.88	98.98
Gabor+MFA	98.59	99.05	99.12
Gabor+EMFA	**98.6**	**99.2**	**99.25**

Example 2 - Experiment on the AR Database

For the AR database [14], we chose a subset consisting of 50 male subjects and 50 female subjects. 14 images without occlusion for each subject were selected: the seven images from session 1 for training, and the other seven from session 2 for testing. In this experiment, each image has the size of 64×80. For the error correction SVM classifier, since there are 100 subjects in the database, the BCH (63, 7) code is selected that has 63 bits in total with 7 information bits and the minimum Hamming distance 31. Table 2 shows the recognition rates for various methods conducted in the experiment on the AR database.

Table 2. Recognition rates (%) tested on the AR database

Feature extractor	*Nearest Neighbor*	*One-Against-One*	*Error Correction SVM*
LPP	76.71	85.00	86.29
EFM	82.14	86.14	88.00
MFA	82.57	87.00	88.14
EMFA	85.14	87.43	88.71
Gabor+LPP	81.57	91.29	94.71
Gabor+EFM	88.14	94.14	95.43
Gabor+MFA	89.14	94.57	95.71
Gabor+EMFA	**90.00**	**95.28**	**96.14**

Example 3 - Experiment on the FERET Database

Here we use the pose subset of the FERET database [15], which includes 1400 images from 200 subjects with 7 images for each subject. The subset is composed of the images marked with 'ba', 'bd', 'be', 'bf', 'bg', 'bj' and 'bk'. In the experiment, each image is resized to the size of 80×80. We select images marked with 'ba', 'be' and 'bg' as the training set, and the remaining images as the testing set. Since there are

200 classes of face images in the database, for the error correction SVM classifier, the (127, 8) BCH code is chosen that has 127 bits in total with 8 information bits and the minimum Hamming distance 63. Table 3 gives the recognition rates for various methods conducted in the experiment on the FERET database.

Table 3. Recognition rates (%) tested on the AR database

Feature extractor	*Nearest Neighbor*	*One-Against- One*	*Error Correction SVM*
LPP	69.88	70.38	72.00
EFM	71.50	74.00	74.25
MFA	71.63	74.38	74.62
EMFA	73.13	74.50	75.00
Gabor+LPP	80.38	83.50	84.50
Gabor+EFM	81.50	85.00	85.50
Gabor+MFA	80.25	85.30	85.88
Gabor+EMFA	**83.00**	**85.75**	**86.25**

From Table 1 to Table 3, it can be seen that, for each classifier, the highest recognition rate can always be obtained using the Gabor-based EMFM features compared to using other features. It can also be seen that, by using the same kind of the features (in each row of the table), the error correction SVM classifier can always achieve the highest recognition rate among the 3 kinds of classifiers. By examining all the results, we can see that the combination of the GEMFM feature with the error correction SVM classifier outperforms all the other combinations.

4 Summary and Further Directions

In this paper, we proposed a new face recognition method using Gabor-based Enhanced Marginal Fisher Model and error correction SVM classifier. In the method, the image to be classified is first filtered by Gabor wavelets, and the high-dimensional Gabor feature is then processed by the Enhanced Marginal Fisher Model to find the underlying structure and extract low-dimensional features. Finally, the GEMFM feature vector is input into the error correction SVM classifier to obtain the classification result.

Many simulation experiments have been conducted to valuate the proposed method, and experimental results show that the GEMFM feature can always obtain the higher recognition accuracy than other features. And, the combination of the GEMFM feature with the error correction SVM outperforms all the other methods.

It is noticed that the computational complexity of the proposed method is quite high for the training process. Thus, more efficient algorithms such as parallel algorithms need to be developed to increase computational speed. This is the problem for further study of the paper.

References

1. Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, A.: Face Recognition: A Literature Survey. ACM Computing Surveys 35(4), 399–458 (2003)
2. Turk, M., Pentland, A.: Eigenfaces for Recognition. J. Cognitive Neuroscience 3(1), 71–86 (1991)
3. Belhumem, P., Hespanha, J., Kriegman, D.: Eigenface vs. Fishefaces: Recognition Using Class Specific Linear Projection. IEEE Trans. PAMI 19(7), 711–720 (1997)
4. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290(5500), 2319–2323 (2000)
5. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500), 2323–2326 (2000)
6. Saul, L.K., Roweis, S.T.: Think globally, fit locally: unsupervised learning of low dimensional manifolds. J. Mach. Learn. Res. (4), 119–155 (2003)
7. Wang, C., Guo, C.: Face Recognition Based on Gabor-enhanced Manifold Learning and SVM. In: Zhang, L., Lu, B. (eds.) ISNN 2010. LNCS, vol. 6064, pp. 184–191. Springer, Heidelberg (2010)
8. Yan, S., Xu, D., Zhang, B., Zhang, H.J.: Graph embedding: A General Framework for Dimensionality Reduction. In: IEEE Conf. Computer Vision and Pattern Recognition, pp. 830–837. IEEE Press, New York (2005)
9. Vapnik, V.: Statistical Learning Theory. John Willey and Sons Inc., New York (1998)
10. Sebald, D.J., Bucklew, J.A.: Support Vector Machines and Multiple Hypothesis Test Problem. IEEE Trans. on Signal Processing 49(11), 2865–2872 (2001)
11. Kreβel, U.: Pairwise Classification and Support Vector Machines. In: Schölkopr, B., Burges, J.C., Smola, A.J. (eds.) Advances in Kernel Methods: Support Vector Learning. MIT Press, Cambridge (1999)
12. Wang, C., Guo, C.: An SVM Classification Algorithm with Error Correction Ability Applied to Face Recognition. In: Wang, J., Yi, Z., Żurada, J.M., Lu, B.-L., Yin, H. (eds.) ISNN 2006. LNCS, vol. 3971, pp. 1057–1062. Springer, Heidelberg (2006)
13. Liu, C., Wechsler, H.: Gabor Feature Based Classification Using the Enhanced Fisher Linear Discriminant Model for Face Recognition. IEEE Trans. Image Processing 11(4), 467–476 (2002)
14. Martinez, A., Benavente, R.: The AR Face Database. Technical report (1998)
15. Phillips, P., Moon, H., Rizvi, S., Rauss, P.: The FERET Evaluation Methodology for Face Recognition Algorithms. IEEE Trans. PAMI 22(10), 1090–1104 (2000)

Facial Expression Recognition by Independent Log-Gabor Component Analysis

Siyao Fu*, Xinkai Kuai, and Guosheng Yang

School of Information and Engineering,
The Central University of Nationalities, Beijing 100081, China

Abstract. Research on facial expression recognition is critical for personalized human-computer interaction (HCI). Recent advances in localized, sparse and discriminative image feature descriptors have been proven to be promising in visual recognition, both statically and dynamically, making it quite useful for facial expression recognition. In this paper we show that the independent Log-Gabor feature (IGF), a localized and sparse representation of pattern of interest, can perform conveniently and satisfactorily for facial expression recognition task. In low-level feature extraction, Log-Gabor wavelet features are extracted, then ICA is applied to produce independent image bases that reduce the redundancy, emphasize edge information, while preserving orientation and scale selection property in the image data. In high-level classification, SVM classifies the propagated independent Log-Gabor features features as discriminative components. We demonstrate our algorithm on facial expression databases for recognition tasks, showing that the proposed method is accurate and more efficient than current approaches.

Keywords: ICA, Gabor Wavelet, Facial Expression.

1 Introduction

Facial expression, in which human emotions are uniquely embodied and visually manifested, is one of the most powerful ways that people coordinate conversation and communicate emotions and other mental, social, and physiological cues. Correspondingly, facial expression recognition plays an extremely important role in a variety of applications such as non-verbal behavior analysis (often refer to the interpretation of non-prototypic expression style such as "raised brows" or "stared eyes" [1]), or intelligent human computer interaction (often refer to the analysis of labeled prototypic expression such as "happy" or "Anger") and security (such as video based surveillance and access control), etc. Various methods for recognizing human facial expressions from face images have been proposed and their performance has been evaluated with databases of face images with

* This work was supported in part by 985 Funding Project (2nd Phase) of Central University of Nationalities (Grant 98501-00300107) and Beijing Municipal Public Information Resources Monitoring Project (Grant 104-00102211)).

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 305–312, 2011.

variations in expressions. Detailed survey please refer to [2]. Since any well developed and widely accepted facial expression analysis system consists of the following two parts: salient facial expression feature extraction and discriminative classifier. Our story follows this tradition, but come up with different view points.

Image representation (features) is arguably the most fundamental task in facial expression recognition. Generally, there are two categories of feature representation: geometric feature based and appearance feature based. Appearance features have been demonstrated to be better than geometric features, for the superior insensitivity to noises, especially illumination noise. Gabor wavelets are reasonable models of visual processing in primary visual cortex and are one of the most successful approaches to describe local appearance of the human face, exhibiting powerful characteristics of spatial locality, scale and orientation selectivity [3]. However they fail to provide excellent simultaneous localization of the spatial and frequency information due to the constraints of the narrow spectral bandwidth, which is crucial to the analysis of facial expression in the highly complex scene. The logarithmic Gabor filters proposed by Field *et al* [4] could be viewed as an incremental modification. However, the dimensionality of the resulting data is very high. For this reason, a computationally effective approach is needed. One common choice would be principle component analysis (PCA), however, the global feature vectors provided by PCA may cause the subspace projection vulnerable to small variation of the input. On the other hand, localized, sparse, spatial feature vector extraction techniques may be less susceptible to occlusion and illumination and thus more suitable for the topic.

As a generalization of PCA, Independent Component Analysis (ICA) represents a mechanism by which not only second-order dependencies but also high-order dependencies of the data are concerned and can be described in meaningful patches. When dealing with feature extraction, most of the ICA basis features extracted from original images turns out to be sparse and similar to localized and oriented edge patches, which can provide essential traits for classification or recognition (see fig. 1). As the result, in the context of discriminative image descriptor, ICA has been shown to produce much better results for those obtained by PCA. However, as a categorized unsupervised learning, crucial class information is not taken into consideration when feature extraction is carried out during ICA algorithm performing. Therefore, high separability of extracted features is not always guaranteed. To overcome this problem, a natural solution is turn to the supervised research field for help.

In this paper, we propose a highly discriminative and computationally efficient facial expression recognition method. In particular, our method first derives a Log-Gabor feature vector from a set of downsampled Log-Gabor wavelet representations of face images, then reduces the dimensionality of the vector by means of PCA, and further reduce redundancy by ICA, forming the salient, independent Log-Gabor features that are suitable for facial expression recognition. The experiment results verify the effectiveness of our approach.

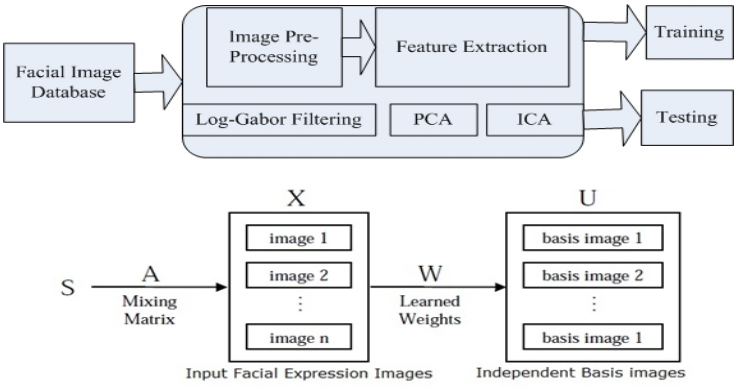


Fig. 1. Framework of the entire facial expression recognition system

The main contribution of this paper is as follows:

1) a hybrid combination of Log-Gabor/ICA feature descriptor is proposed.
2) a verification of our proposed method on a multi-category large scale facial expression dataset is presented.

To our best knowledge, this is the first published work on facial expression recognition using independent Log-Gabor feature descriptor.

2 Detailed Implementation

2.1 Gabor Wavelet Based Feature Extraction

Commonly recognized as presenting the best simultaneous localization of spatial and frequency information, the Gabor wavelets have been found to be particularly suitable for image decomposition and representation when the goal is the derivation of local and discriminating features. Most recently, Donato et al [7] have experimentally shown that the Gabor filter representation offers better performance for classifying facial actions, which provides the theoretical foundations for our contribution.

The Gabor wavelets (kernels, filters) can be defined as follows

$$\psi_{\mu,\nu}(z) = \frac{\|k_(\mu,\nu)\|^2}{\sigma^2} e^{-\frac{\|k_{\mu,\nu}\|^2 \|z\|^2}{2\sigma^2}} [e^{ik_{\mu,\nu}z} - e^{-\frac{\sigma^2}{2}}] \quad (1)$$

where μ, ν define the orientation and scale of the Gabor kernels, $z = (x, y)$, and we have $k_{\mu,\nu} = k_v e^{j}\psi_u$, where $k_v = k_{\max}/f^{''}$ and $\psi_u = \pi u/8$. f is the spacing factor between kernels in the frequency domain. Fig. 2 shows the real part of the Gabor kernels at five scales and eight orientations and their magnitudes (top left part) and a demonstration of the Gabor based feature descriptor.

2.2 Log Gabor Filters

As an enhancement of the Gabor approach, the Log-Gabor functions is adopted in this paper. Being characterized by discarding DC component, which reinforces the contrast ridges and edges of images, and having the transfer function with an extended tail at the high frequency end, which contributes to broad spectral information with localized spatial extent, and preserving true ridge structures of images, the kernels exhibit strong characteristics of spatial locality, scale and orientation selectivity (fig. 2 left bottom part), corresponding to those displayed by Log-Gabor filters, making them a suitable choice for image feature extraction since our goal is to derive local and discriminating features for (facial expression) classification.

The log-Gabor filters defined in the frequency domain using polar coordinates by the transfer function $H(f, \theta)$ can be represented in a polar form as:

$$H(f,\theta) = H_f \times H_\theta = \exp\{\frac{1}{2}\frac{(\ln \frac{f}{f_0})^2}{(\ln \frac{\sigma_f}{f})^2}\} \exp\{-\frac{1}{2}\frac{(\theta-\theta_0)^2}{\sigma_\theta^2}\} \tag{2}$$

the radial component H_f controlling the bandwidth and the angular component H_θ, controlling the spatial orientation that the filter responds to. See Lajevardi's work for detailed explanation in [6].

2.3 Independent Component Log-Gabor Feature Extraction

The obtained Log-Gabor feature vector resides in a space of very high dimensionality. For this reason, dimension reduction techniques are introduced to acquire a more sparse, decorrelated and discriminative subset of the feature set. PCA is a widely used technique for such purpose, achieving optimal signal reconstruction by using a subset of principal components to represent the original signal. However, sometimes the subtle facial expression movements can not be characterized by second order statistics due to the fact that the movements of the muscles which control such subtle facial expressions are actually relatively small and not sufficient to constitute reliable statistics, thus, they can not be captured by the second order statistics based PCA or related methods. To get the tradeoff between discriminativeness and sparsity, independent component analysis (ICA) seems to be a promising solution. Note that dimensionality reduction is the third preprocessing step that ICA use to facilitate the iterative computing procedure (the first and second are the centering and whitening, respectively). Interestingly, using the ICA based techniques to perform dimension reduction and further feature extraction have received little attention in the past perhaps due to the fact that the ICA is not originally developed for that purpose, instead, it focus mainly on separating a multivariate signal into additive subcomponents supposing the mutual statistical independence of the non-Gaussian source signals.

Similar to the independent Gabor feature extraction method proposed in [2], independent Log-Gabor Features (ILGF) method applies the independent component analysis on the (lower dimensional) Gabor feature vector defined by

Eq. 2. In particular, the Log-Gabor feature vector $\Psi(p)$ of an image p is first calculated as detailed in Sect II. PCA then reduces the dimensionality of the acquired Log-Gabor feature vector and derives the lower dimensional feature vector, $S(p)$ (see Eq. 2). Next, the ILGF method derives the overall (the combination of the whitening, rotation, and normalization transformations) ICA transformation matrix, A. The new feature vector, $Z(p)$, of the image p is thus defined as follows:

$$S(p) = AZ(p) \tag{3}$$

Finally, after the extraction of features, we choose SVM as the classifier.

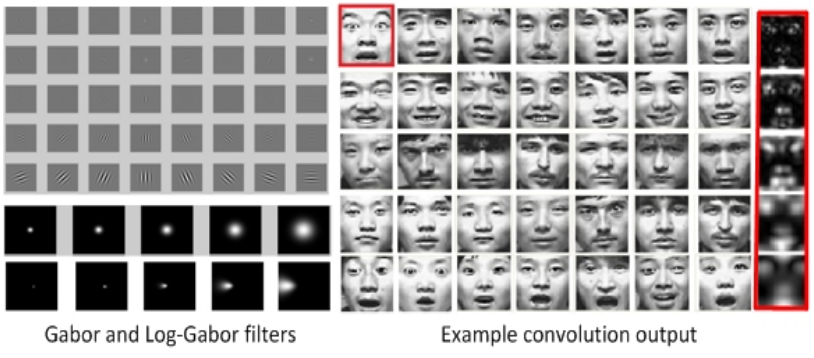


Fig. 2. A large scale facial expression dataset (including different typical facial expressions) and the convolution output of a sample image (the first framed image). Note that the outputs exhibits powerful characteristics of spatial locality, scale and orientation selectivity, providing highly salient local features, such as the eyes, brows, nose, and mouth, essential for facial expression recognition.

3 Experiments

3.1 Database Description

The experiment presented here is the evaluation of the approach on a newly created facial database we have designed and constructed, namely, a large-scale racially diverse face database, the CUN face database, which covers different source of variations, especially in race, facial expression, illumination, backgrounds, pose, accessory, etc. Currently, it contains 112,000 images of 1,120 individuals (560 males and 560 females) from 56 Chinese "nationalities" or ethnic groups.

The aim of the database is listed as follows:

1, to provide the worldwide scholars of face recognition with exhaustive ground-truth information in a cross-race face database. While most of the current face database mainly consists of Caucasian people, we mainly focus on the "cross-race effect" during the face recognition experiment.

2, aimed at understanding culture specific difference in facial expression production and interpretation, which have been long viewed as a crucial interlink between individual and social communication.

Fig. 3 shows the configuration of the photographic room, including lamps, camera system, etc, and some typical example images of subjects belonging to different races in our database.

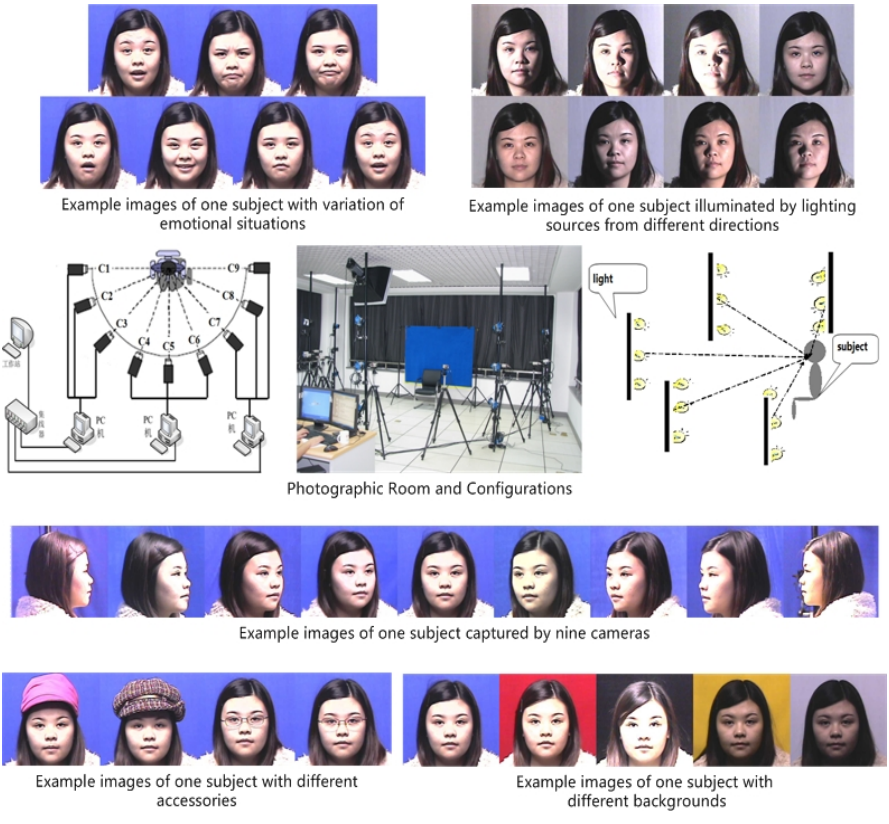


Fig. 3. Diagram showing the whole configuration of CUN face database

The most commonly used baseline facial recognition algorithm (Gabor+ICA) and our proposed method are evaluated on the seven frontal datasets (six different typical facial expressions plus one neutral expression). Before training and testing, all the images are preprocessed (histogram equalization and geometric normalization). We experimented then with the independent Log-Gabor features, which were computed as follows: first, PCA reduced the dimensionality of the Log-Gabor convolution outputs downsampled by a factor 64; and second, ICA derived the independent Log-Gabor features from the reduced convolution

outputs. For the evaluation of our system, we used the publicly available libsvm library. Each experiment was repeated for 200 iterations and average results obtained for overall accuracy.

Fig. 4 shows the performance. As it can be seen from the figure, both methods shows that the performance is best when dealing with happiness and neutral, which are easy to interpret. However, neither methods perform well as dealing with disgust and fear, two reasons can be account for this, first, in eastern Asia, people seldom express negative inner emotions explicitly, esp, when posing facial expression intentionally instead of naturally, making it hard to analysis those facial expressions, whereas in some other datasets consists of mainly Caucasian subjects, it is relatively easy to analyze for negative facial expressions more directives and explicitly. Second, individual subjects may differ from each other when posing facial expressions with different intensity. Nevertheless, the boost of the accuracy still implies that using independent Log-Gabor feature based on ICA instead of conventional Gabor features extracted neither directly from original grayscale images or ICA can yield satisfactory results.

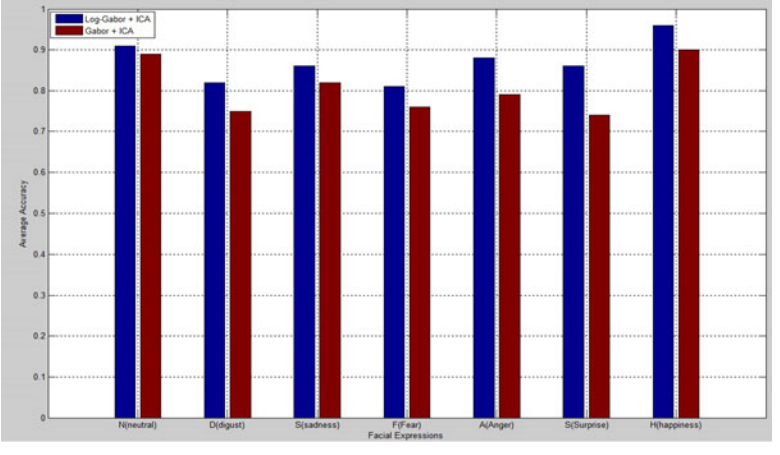


Fig. 4. Empirical recognition result

Fig. 5. Empirical recognition result under different illumination directions

During the experiment, we also found that the independent Log-Gabor features are robust to the background noise such as illumination and pose variation, fig. 5 shows some empirical results and detailed variation recognition results are given in fig. 6.

Illumination / Degree / Facial Expression	30° (150°) (Left Light) (Right Light)	45° (135°) (Left Light) (Right Light)	60° (120°) (Left Light) (Right Light)	90° (frontal Light)
Neutral	6/10	6/10	7/10	9/10
Anger	6/10	6/10	8/10	10/10
Sadness	5/10	7/10	7/10	8/10
Happiness	7/10	7/10	6/10	10/10
Fear	3/10	5/10	6/10	8/10
Disgust	4/10	6/10	6/10	9/10
Surprise	4/10	5/10	8/10	10/10

Fig. 6. Empirical recognition result under different illumination directions

4 Conclusions and Future Work

We have proposed a method for local independent Log-Gabor feature descriptors which can perform better than other descriptors for facial expression recognition, since the ICA-based representation is localized and sparse, providing highly discriminative and efficient feature descriptors. In future work we plan to apply the method to the dynamic facial expression recognition problem.

References

1. Ekman, P.: Emotion in the Human Face. Cambridge University Press, New York (1982)
2. Fasel, B., Luettin, J.: Automatic facial expression analysis: a survey. Pattern Recognition 36(1), 259–275 (2003)
3. Liu, C., Wechsler, H.: Independent Component Analysis of Gabor Features for Face Recognition. IEEE Trans. Neural Networks 14(4), 919–928 (2003)
4. Field, D.J.: Relations between the images and the response properties of cortical cells. Jour. of the Optical Society of America, 2379–2394 (1987)
5. Fu, S.Y., Hou, Z.G., Yang, G.S.: Multiple Kernel Learning with ICA: Local Discriminative Image Descriptors for Recognition. In: Proceedings of the International Joint Conference of Neural Networks (2010)
6. Lajevardi, S.M., Hussain, Z.M.: Facial Expression Recognition Using Log-Gabor Filters and Local Binary Pattern Operators. In: Proceedings of the International Conference on Communication, Computer and Power, pp. 349–353 (2009)
7. Donato, G., Bartlett, M.S., Hager, J.C., Ekman, P., Sejnowski, T.J.: Classifying facial actions. IEEE Trans. Pattern Analysis and Machine Intelligence 21(10), 974–989 (1999)

Learning Hierarchical Dictionary for Shape Patterns

Xiaobing Liu and Bo Zhang

State Key Laboratory of Intelligent Technology and Systems,
Tsinghua National Laboratory for Information Science and Technology,
Department of Computer Science and Technology,
Tsinghua University, Beijing 100084, China
liuxb02@mails.tsinghua.edu.cn, dcszb@mail.tsinghua.edu.cn

Abstract. Shape information is essential for image understanding. Decomposing images into shape patterns using a learned dictionary can provide an effective image representation. However, most of the dictionary based methods retain no structure information between dictionary elements. In this study, We propose Hierarchical Dictionary Shape Decomposition (HiDiShape) to learn a hierarchical dictionary for image shape patterns. Shift Invariant Sparse Coding and HMAX model are combined to decompose image into common shape patterns. And the Sparse Spatial and Hierarchical Regularization (SSHR) is proposed to organize these shape patterns to construct tree structured dictionary. Experiments show that the proposed HiDiShape method can learn tree structured dictionaries for complex shape patterns, and the hierarchical dictionaries improve the performances of corrupted shape reconstruction task.

Keywords: Hierarchical Dictionary, Image Shape, Sparse Coding, Shift Invariant, HMAX, unsupervised learning.

1 Introduction

Shape provides crucial information for object recognition and image understanding. With only shape information, human can quickly and correctly recognize an object from a caricature, a black-white cartoon image, or a sketch with several hand drawn lines. To stimulate V1 cells in human visual cortex, sparse coding [1] is introduced to learn Gabor-like basis patterns. And it decomposes an image patch into the linear combination of several Gabor-like shape patterns. But on learning shape patterns, the classical sparse coding has two drawbacks . (1) It is hard to learn more complex shape patterns than Gabor-like filters. (2) The "flat" dictionary ignores the relationship between the elements.

Shape patterns are not independent from each other. The shape contours of the objects from the same categories may be similar but not identical. And it is possible that some shape patterns are dissimilar but share equivalent semantic meaning, such as sofa and chair. Making an analogy between shape in image

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 313–322, 2011.

understanding and word in text analysis, shape should have its own "word-net" or "shape-net". It should be organized as a hierarchical dictionary containing various meaningful common shape patterns and between-shape dependence relationships (Please refer to Fig.4 as a brief idea). This type of hierarchical dictionary will be helpful for object recognition and image understanding. How can we learn complex and meaningful shape patterns? And how can we organize these shape patterns by hierarchical structure? This is the story of this work.

Most of the previous sparse coding approaches [1] [2] [3] [4] [5] can only learn "flat" dictionaries of Gabor-like patterns. Several works [6] [7] [8] can learn structured dictionaries. However, without any alignment technique, they can only learn pixel level patterns but not complex shape patterns. SISCHMAX [9] can learn complex shape patterns from given images. But it cannot learn a hierarchical dictionary.

We propose Hierarchical Dictionary Shape Decomposition (HiDiShape) to learn a hierarchical dictionary for image shape patterns. Each dictionary element is a vector representation for a shape pattern. We follow our previous work SISCHMAX [9] to handle the global and local position variance of the shape patterns. And we introduce Sparse Spatial and Hierarchical Regularization (SSHR) to organize the dictionary by tree structures. This regularization term encourages the coefficient co-occurrence of shape patterns at the same position in the same dictionary subtree. Therefore, the learned shape patterns in the same dictionary subtree prefer to being similar and synchronous. We test the proposed HiDiShape method on several categories from Caltech 101 dataset. HiDiShape can learn tree structured dictionaries for shape patterns from given images. And the learned hierarchical dictionaries perform better than flat ones (such as the classical sparse coding) on reconstructing corrupted image shapes.

The rest of this paper is organized as following. Section 2 introduces how to learn dictionaries for image shape patterns. And we propose the method to learn hierarchical dictionaries for image shape patterns in Section 3. The experiment results are illustrated in Section 4. Finally, in Section 5, we summarize the conclusions of this paper, and discuss our future work.

2 Learning Image Shape Patterns

A common shape component may occur at any place of the image, with any local position variance. Both the global and local position variances increase the difficulty of learning complex shape contours. If only randomly sampling patches directly from the pixel-level images, the dictionary will turn to Gabor filters at all kinds of positions. Without any consideration for global and local variances, this rough method can only learn lines but not shape contours with meaningful corners or cycle structures.

In this paper, we adopt the SISCHMAX [9] method to learn the "flat" dictionary.(The further extension is proposed in Section 3 to learn the "hierarchical" dictionary.) SISCHMAX combines shift invariant factor (Shift Invariant Sparse Coding [10]) and Local max method (HMAX model [11]), to robustly handle

the global and local position variance of the shape contours. And it can learn common complex shape contours from given images. Since our proposed method is based on SISCHMAX, this section will briefly introduce SISCHMAX method.

2.1 Sparse Coding

Sparse coding [1] is a well established component analysis method. Given a dictionary $\boldsymbol{B} = \{B_t\}$, each sample vector Y^k is reconstructed by the linear combination of the basis vectors B_t and their corresponding coefficients α_t^k. The reconstruction error for Y^k is as following.

$$\min_{\{\alpha_t^k\}} \|Y^k - \sum_t \alpha_t^k B_t\|_2^2 + \lambda \sum_t |\alpha_t^k| \tag{1}$$

L1-regularization leads to a sparse representation. Dictionary can be randomly sampled from the datasets or learned [4, 12]. Sparse coding provide an extensible framework for unsupervised dictionary learning. And HiDiShape is also formulated in sparse coding framework.

2.2 Shift Invariant Sparse Coding

Shift Invariant Sparse Coding (SISC) [10, 13, 14] adds shift invariance to the sparse coding framework. An image may be constructed by several objects. An image pattern is a part of an object, which may occur at any location. Given the target image and a group of basis patterns, reconstructing an image is not only to recover the appearance but also to recover the location of the part. Fig. 1 briefly shows how SISC reconstructs an image, and it also shows sparse coding for comparison. A SISC basis can be placed anywhere in the image.

Learning SISC dictionary is equivalent to optimize Eq. (2). Y^k represents the image feature vector such as pixel values. Denote $\Phi(B_t, z)$ a transformation

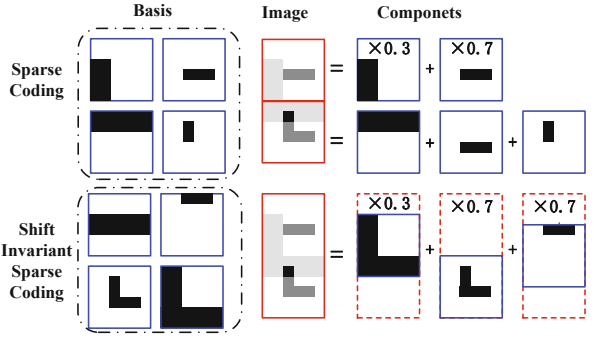


Fig. 1. Illustration of sparse coding and SISC (Best viewed in color). This figure is cited from [9]. A SISC component is a basis occurring at any position in the image. The linear combination of these components approximately reconstructs the image.

which places the basis pattern B_t at the location z on an empty image with the same size of Y. t is the basis index. And z is the location of basis patch in the image. The corresponding coefficient of B_t at location z is $\alpha_{t,z}$. The linear combination of $\Phi(B_t, z)$ and $\alpha_{t,z}$ is used to approximate the image Y. Therefore, the coefficients set $\boldsymbol{\alpha}$ covers all the conditions that any basis occurs at any place. λ is the weight for coefficient sparsity loss. For each $\alpha_{t,z}$, we can consider $X_i = \Phi(B_t, z)$ as a basis in the original sparse coding framework.

$$\min_{\{B_t\},\{\alpha^k_{t,z}\}} \frac{1}{n} \sum_{k=1}^{n} \frac{\|Y^k - \sum_{t,z} \alpha^k_{t,z} \Phi(B_t, z)\|_2^2 + \lambda \sum_{t,z} |\alpha^k_{t,z}|}{Area(k)} \tag{2}$$

There are substantial differences between SISC and the original version of sparse coding. The original sparse coding pastes the basis to the whole target image or patch, while SISC can paste a basis at any position. Overlap is also allowed. SISC can reduce to sparse coding by prohibiting z and restricting the image shares the same size with basis patches.

2.3 HMAX Model

HMAX model is a biological inspired computational model for visual cortex [15] [16] [11]. It stimulates the feed-forward processing of the primary visual cortex (V1) of human. HMAX works very well on the local invariance of position and scale.

The S1 units in HMAX are designed according to the simple cells of Hubel and Wiesel found in the primary visual cortex(V1)[17]. Gabor function [18] [19] is adopted to serve as the response function of S1 units. This gabor like unit can represent a short part of straight line at a specific position and scale. To detect all the line parts at all positions, the filters slide on the whole image. And the filter pyramids are built to capture the responses from different scaled versions of the filters at any position.

The C1 units of HMAX perform like cortical complex cells. The input of C1 layer is the output of S1 layer. A C1 unit is implemented as a maximum operator over a local region of the S1 units within a range of S1 filter scales. It tolerates the local position and scale variance, but it keeps orientation sensitivity and robust output. Only S1 and C1 units of HMAX model are used in SISCHMAX [9].

2.4 Shift Invariant Sparse Coding HMAX

Shape contour is the essential visual representation for the object. However, contour base representation is not as popular as color and texture based features in the computer vision area for last twenty years. Possible reason is that shape contour has a lot of variance, and it is a higher level representation which is hard to extract stably. The local position and scale variance and lack of fine aligned images limits the extraction of common shape contour.

HMAX model is good at tolerating local position and scale variance, while shift invariant sparse coding is capable to latently align the basis patterns. SISCHMAX [9] combines this two work for shape discovering. (Please refer to the

left part of Fig. 3.) The image is first processed by HMAX to produce C1 layer output. This step will robustly extract small parts of lines. After that the response map is processed by shift invariant sparse coding algorithm to discovery common shapes. The different C1 orientation correspond to different feature channels in sparse coding. The bases learned by shift invariant sparse coding are the common shape patterns.

All the parameters for Gabor-filters are the same as [11], except that there are 8 orientations in this paper. Each C1 unit take a 10×10 receptive field over S1 response map. And C1 units are placed every 10 S1 units. Inhibition strategy is adopted as [11], thus the small response values are restricted to zero.

3 Hierarchical Dictionary Shape Decomposition

3.1 Sparse Spatial and Hierarchical Regularization

Instead of the "flat" dictionary, we focus on learning a "hierarchical" structured dictionary. Each basis can be considered as a node. And all the elements of the dictionary can construct one or several trees. The structure of the tree is predefined. In this paper, we organize the dictionary elements by several balanced 2-branch trees. (Please refer to Fig.2 for example). To learn a tree structure, Eq.(3) defines the Sparse Spatial and Hierarchical Regularization (SSHR).

$$\Omega(\boldsymbol{\alpha}) = \sum_{z} \sum_{s} w_s \max_{t \in subtree(s)} \|\alpha_{t,z}\| \tag{3}$$

where $\alpha_{t,z}$ is the coefficient of the basis t at any position z. For any basis node s in the dictionary, $subtree(s)$ is the set of the nodes of the subtree whose root is s. w_s is the weight for the dictionary subtree rooted by s. In this paper, all the w_s is simply set to 1. The max operator works over all the coefficients in a subtree. This design encourages simultaneously occurrence of the coefficients in the same subtree. And the *sum* operator over all subtree cost keeps the sparsity of active positions and subtrees. This regularization formulation derives from the Sparse Hierarchical Dictionary Learning [6] [20] [21]. The position factor is additionally considered in this paper to handle the global shift invariance over [6].

The proposed SSHR drives the learned hierarchical dictionary to have the following properties. (1) The shape patterns in the same subtree are probably similar or complementary with high co-occurrence ratio. Because relative shape patterns probably have large coefficients simultaneously. These patterns will be less punished, if they are in the same subtree. Thus the subtree works like to cluster shape patterns with similar semantic meaning. (2) The shape patterns at tree roots (e.g. node "a" and "h" in Fig.2) have higher occurrence probability than those at tree leaves (e.g. node "m","n" in Fig.2). Because leaf node coefficient is more punished by attending all the maximum operation of its ancestor nodes. It is similar to Huffman Coding. (3) The coefficients retain the sparse property from sparse coding, especially between different trees. (4) The coefficients are sparse over different positions.

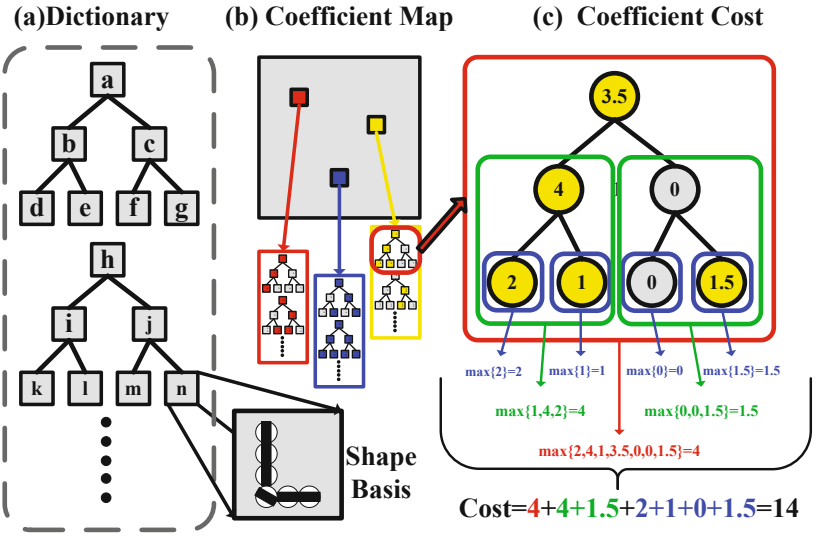


Fig. 2. Illustration for the tree-structured dictionary and costs of the coefficients in the subtrees.(Best viewed in color) (a) Dictionary with tree structures. Each squared rectangle (e.g. "a","b","c") represents a dictionary element or shape patterns. (b) Coefficient map for an image. Each position (e.g. the yellow pixel) can have a group of coefficients. (c) How to compute the coefficient cost of a tree at a position.

3.2 Hierarchical Dictionary Shape Decomposition

To learn a hierarchical dictionary for shape patterns, we extend the SISCHMAX [9] method by introducing the Sparse Spatial and Hierarchical Regularization. The SISCHMAX method aligns the shape patterns during decomposition processing. It keeps the advantages of the Shift Invariant Sparse Coding and HMAX model. It can tolerate the global and local position variance for shape patterns. But it can only learn "flat" dictionary. In our HiDiShape, predefined tree structured dictionary is used instead of "flat" dictionary. And Sparse Spatial and Hierarchical Regularization is introduced to replace L1-norm in SISCHMAX method to encourage co-occurrence coefficients in the same subtree at the same position. Fig.3 describes the overview of the proposed Hierarchical Dictionary Shape Decomposition (HiDiShape) method.

$$\min_{\{B_t\},\{\alpha^k_{t,z}\}} \frac{1}{n}\sum_{k=1}^{n} \frac{\|Y^k - \sum_{t,z} \alpha^k_{t,z}\Phi(B_t,z)\|_2^2 + \lambda\Omega(\boldsymbol{\alpha}^k)}{Area(Y^k)} \tag{4}$$

The cost function for learning hierarchical dictionary is Eq.(4). where B_t is basis and is constrained by $\|B_t\|_2 \leq 1$. z is the position. $\alpha^k_{t,z}$ is the coefficient of the basis t in the image Y^k at position z. Denote $\Phi(B_t, z)$ a transformation which places the basis pattern B_t at the location z on an empty image with the same size of Y^k. n is the number of the training images. And $Area(Y^k)$ is the

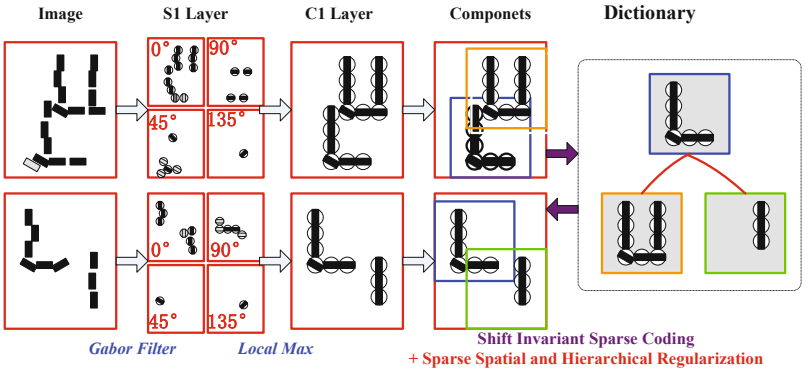


Fig. 3. Overview of the proposed HiDiShape method (Best viewed in color). The two training images contains "L","U" and "I" contour with obvious local variance. After the processing of S1 and C1 layer, local variance are handled. And after shift invariant sparse coding using hierarchical dictionary, common shape patterns are aligned and learned and organized as dictionary trees.

area of image Y^k. We use L2-constraint for the bases. λ is the cost weight for regularization term. Ω is described by Equation 3.

The above function is convex over $\boldsymbol{B} = \{B_t\}$, if the $\boldsymbol{\alpha} = \{\alpha^k_{t,z}\}$ are fixed. And it is also convex over $\boldsymbol{\alpha}$, if $\boldsymbol{B}$ are fixed, although the differential coefficient of $\Omega(\alpha)$ is not continuous. Therefore, we can adopt a two-step strategy. The target function is first optimized over $\boldsymbol{\alpha}$ with $\boldsymbol{B}$ fixed to update the coefficients for each image, and then is optimized over basis $\boldsymbol{B}$ by fixing $\boldsymbol{\alpha}$. These two-step procedure can run iteratively until convergence. And gradient descent method is used in this work to optimize this overall cost function.

The dictionary elements learned by the proposed HiDiShape method are the shape patterns. And to visualize the result, we create a shape map for each learned shape pattern. For each C1 unit at the same position of a basis pattern, only one unit with the maximum response survives to show its corresponding Gabor filter in the contour image. Thus a shape pattern image is created with the same size as the original image. If the shape patterns are common, they will look familiar to human, or are even clear enough to recognize an object.

4 Experiment Results

The experiment images are from Caltech 101 image dataset. All the images we selected are from three categories: "Face_easy", "Motorbikes" and "Yin_yang". For each image, the width or height is resized to 200. And 60 images are randomly sampled to train the shape dictionary. The regularization weight λ is set to 0.5 .

Fig.4 illustrated the dictionary learned by HiDiShape. There are 4 dictionary trees. Each non-leaf node has two child nodes. And each tree is 3 layers with 7 nodes. The shape pattern is represented by 20×20 HMAX C1 cell with 8

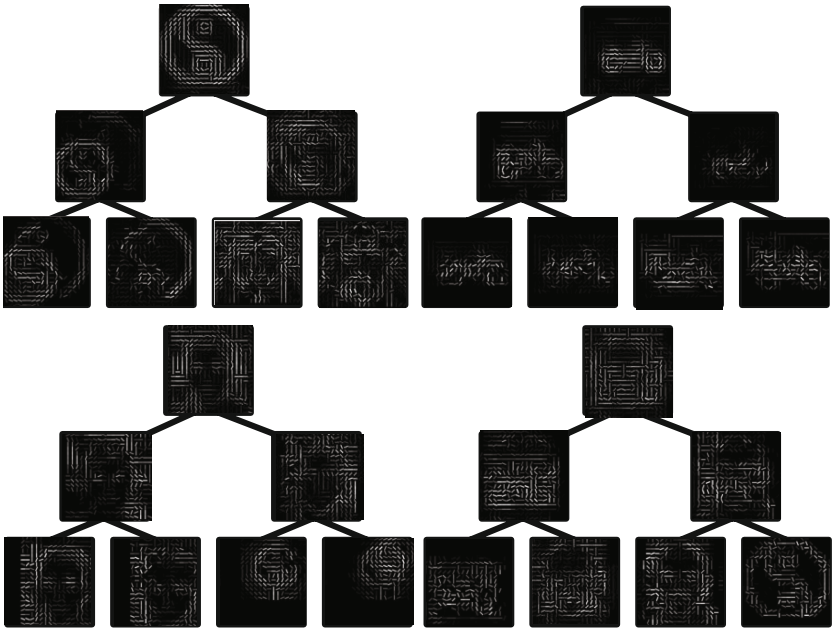


Fig. 4. Dictionary Learned by HiDiShape for "Face_easy", "Motorbikes" and "Yin_yang" categories

orientation. It is obvious that some of the shape patterns in the same tree is very alike. Some of the shape patterns are almost clear enough to recognize as object parts or even the whole object, such as Yin_yang graph , motorbike and face. (Please refer to Fig.4 and view the root nodes of top left tree, top right tree and bottom left tree respectively.) And the most of the root nodes look more clear than its descendant nodes. A possible reason is that these root node shape patterns are more common than other patterns learned in this image dataset.

For corrupted shape reconstruction task, we use 60 images as testing dataset. And the dictionary sizes are all set to 42 for different methods. The shape pattern

Table 1. Quantitative results of the corrupted shape reconstruction task. The First row shows percentages of missing C1 units. And the other rows show mean square errors per C1 unit multiplied 100 for classical sparse coding, SISCHMAX and HiDiShape. HiDiShape(a) contains 14 two-layer trees. HiDiShape(b) have 6 three-layer trees.

noise	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%
Sparse Coding	0.927	0.958	0.998	1.049	1.110	1.183	1.263	1.354	1.439	1.513
SISCHMAX	0.275	0.334	0.414	0.512	0.630	0.764	0.921	1.091	1.263	1.431
HiDiShape(a)	0.263	0.316	0.392	0.484	0.597	0.726	0.879	1.047	1.211	1.388
HiDiShape(b)	0.274	0.327	0.402	0.496	0.606	0.733	0.882	1.044	1.212	1.388

is represented by 8×8 HMAX C1 cell with 8 orientation. Since we focus on shape reconstruction but not image pixel reconstruction, the HMAX C1 output is directly used as the reconstruction unit. For each test image, we randomly select and remove C1 values according to the specific noise percentage (missing rate). The mean square errors are shown in Table 1. The reconstruction errors of HiDiShape are lower than SISCHMAX, and significantly lower than the classical sparse coding. This result shows that hierarchical dictionary and SISCHMAX are both very helpful to learn better shape representations.

5 Conclusion

Shape representation is crucial for image understanding. In this study, we focus on learning tree structured dictionaries for common shape patterns. The Hierarchical Dictionary Shape Decomposition (HiDiShape) is proposed not only to robustly learn complex common shape patterns but also to embed them in tree structures. The experiments illustrated the superiority of the hierarchical dictionary over flat dictionary on corrupted shape reconstruction task.

This approach provides an optional method to build a word-net-like hierarchical dictionary for image shapes, if the images from the internet are collected for training. Improving computational efficiency will be worthful in this case. The Sparse Spatial and Hierarchical regularization is also easy to extend for other types of graph structures. And we also urgently want to know whether and how much these kind of structured dictionaries can help object recognition and image segmentation.

Acknowledgments. The authors would like to thank Dr. Xiaolin Hu and Dr. Jianmin Li for helpful discussion. This work is supported by the National Natural Science Foundation of China under Grant No. 60805023 and No. 90820305, National Basic Research Program (973 Program) of China under Grant No. 2007CB311003, and Basic Research Foundation of Tsinghua National Laboratory for Information Science and Technology (TNList).

References

1. Olshausen, B., Fteld, D.: Sparse coding with an overcomplete basis set: A strategy employed by V1? Vision Research 37(23), 3311–3325 (1997)
2. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online dictionary learning for sparse coding. In: Proceedings of the 26th Annual International Conference on Machine Learning. ACM, New York (2009)
3. Mairal, J., Sapiro, G., Elad, M.: Learning multiscale sparse representations for image and video restoration. SIAM Multiscale Modeling and Simulation 7(1), 214–241 (2008)
4. Raina, R., Battle, A., Lee, H., Packer, B., Ng, A.: Self-taught learning: Transfer learning from unlabeled data. In: Proceedings of the 24th International Conference on Machine Learning, p. 766. ACM, New York (2007)

5. Lee, H., Battle, A., Raina, R., Ng, A.: Efficient sparse coding algorithms. Advances in Neural Information Processing Systems 19, 801 (2007)
6. Jenatton, R., Mairal, J., Obozinski, G., Bach, F.: Proximal Methods for Sparse Hierarchical Dictionary Learning. In: Proceedings of the 27th International Conference on Machine Learning. ACM Haifa, Israel (2010)
7. Kavukcuoglu, K., Ranzato, M., Fergus, R., LeCun, Y.: Learning invariant features through topographic filter maps. In: CVPR, pp. 1605–1612 (2009)
8. Bengio, S., Pereira, F., Singer, Y., Strelow, D.: Group Sparse Coding. In: Neural Information Processing Systems, Whistler, Canada (2009)
9. Liu, X., Zhang, B.: SISCHMAX: Discovering Common Contour Patterns. In: Proceedings of the 9th IEEE International Conference on Cognitive Informatics. Tsinghua University, Beijing (2010)
10. Mørup, M., Schmidt, M., Hansen, L.: Shift invariant sparse coding of image and music data. Submitted to Journal of Machine Learning Research (2008)
11. Mutch, J., Lowe, D.: Object class recognition and localization using sparse features with limited receptive fields. International Journal of Computer Vision 80(1), 45–57 (2008)
12. Labusch, K., Barth, E., Martinetz, T.: Simple method for high-performance digit recognition based on sparse coding. Behaviour 14, 15
13. Potluru, V., Plis, S., Calhoun, V.: Sparse shift-invariant NMF. In: IEEE Southwest Symposium on Image Analysis and Interpretation, SSIAI 2008, pp. 69–72 (2008)
14. Le Roux, J., de Cheveigne, A., Parra, L.: Adaptive template matching with shift-invariant semi-NMF. In: Proc. NIPS, vol. 8 (2008)
15. Riesenhuber, M., Poggio, T.: Hierarchical models of object recognition in cortex. Nature Neuroscience 2, 1019–1025 (1999)
16. Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., Poggio, T.: Robust object recognition with cortex-like mechanisms. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(3), 411 (2007)
17. Hubel, D., Wiesel, T.: Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. The Journal of Physiology 160(1), 106 (1962)
18. Gabor, D., et al.: Theory of communication. SPIE Milestone Series Ms 181, 120 (2006)
19. Jones, J., Palmer, L.: An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex. Journal of Neurophysiology 58(6), 1233 (1987)
20. Zhao, P., Rocha, G., Yu, B.: The composite absolute penalties family for grouped and hierarchical variable selection. Annals of Statistics 37(6A), 3468–3497 (2009)
21. Kim, S., Xing, E.P.: Tree-guided group lasso for multi-task regression with structured sparsity. In: ICML, pp. 543–550 (2010)

Sparse Based Image Classification with Different Keypoints Descriptors

Yuanyuan Zuo* and Bo Zhang

State Key Laboratory of Intelligent Technology and Systems,
Tsinghua National Laboratory for Information Science and Technology,
Department of Computer Science and Technology,
Tsinghua University, Beijing, 100084, China

Abstract. In this paper, we apply the sparse representation based algorithm to the problem of generic image classification. Keypoints with different descriptors are used as the bases of the training matrix and test samples. A learning algorithm is also presented to select the most important keypoints as the bases of the training matrix. Experiments have been done on 25 object categories selected from Caltech101 dataset, with salient region detector and different descriptors. The results show that keypoints with histogram of oriented gradients descriptor can achieve good performance on image categories which have distinctive patterns detected as keypoints. Furthermore, the base learning algorithm is useful for improving the performance while reducing the computational complexity.

Keywords: Image classification; sparse representation; keypoints.

1 Introduction

The task of image classification involves two important issues. One is image representation, the other is classification algorithm.

Recently, keypoints-based image features are getting more and more attention in the computer vision area. Keypoints, also known as interest points or salient regions, refer to local image patches which contain rich information, have some kind of saliency and can be stably detected under a certain degree of variations. Extraction of keypoints-based image feature usually includes two steps. First, keypoints detectors are used to find keypoints automatically. Second, keypoints descriptors are used to represent keypoints features. Ref. [1] and [2] gave a performance evaluation among several different keypoints detectors and descriptors respectively.

Corresponding to the different kinds of image representation, many classification algorithms have been proposed, which can be divided into two classes.

* The work is supported by the National Natural Science Foundation of China under Grant Nos. 90820305; the National Basic Research Program (973 Program) of China under Grant Nos. 2007CB311003.

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 323–330, 2011.

One is generative models such as constellation model [3]. The other is discriminative models, such as support vector machines (SVM), which have been proved to be effective for object classification in [4-6].

Recently, sparse coding has been used for the learning of the codebook and image representation [7]. Wright et al. proposed sparse representation based classification [8] for the recognition of human faces. Although good performances have been achieved with the algorithm, the image database is strictly confined to human frontal faces with only illumination and slight expression changes.

In [9], we have applied the sparse representation based classification algorithm to the problem of generic image classification with a certain degree of background clutter, scale, translation, and rotation variations within the same image class. Bag of visual words features are used in the experiments. Comparable experimental results have been obtained with SVM classifiers under different size of vocabulary and numbers of training images.

However, bag of visual words features quantize local features into a given size of codebook and reflect the distribution of visual words detected on an image. Original local features of keypoints are omitted, which may contain distinctive patterns and be very helpful for recognizing some classes of images. In this paper, we propose local features for image classification based on sparse representation (Local-SRC). A base learning algorithm is also presented in order to select the most important keypoints as the bases of the training matrix.

The remainder of this paper is organized as follows. In section 2, the extraction method of keypoints is described. Section 3 gives a detailed description of the Local-SRC algorithm. A base learning algorithm is presented in section 4, followed by experiments and conclusions in section 5 and 6.

2 Image Keypoints Extraction

The keypoints extraction method includes the following two steps.

1) Keypoints detector. Salient region detector [10] proposed by Kadir et al. is one of the most widely used keypoints detectors. This detector selects regions which exhibit unpredictability both in the local attributes space and scale space. Unpredictability of image regions is measured by Shannon entropy of local image attributes, such as the pixel gray value. A value of saliency is computed for each region and the salient regions are sorted by the value of saliency. The amount of regions which are detected in one image usually varies from dozens to hundreds.
2) Keypoints descriptor. Histograms of Oriented Gradients (HOG) descriptor [11] is used to describe the feature for each keypoint. It computes gradients for every pixel in the keypoint local patch. The orientation of gradients (unsigned $0^o - 180^o$, or signed $0^o - 360^o$) is quantized to a certain number of bins. Local patches can be divided into different size of blocks, on which HOG features are computed. Ref. [11] experimented on different block sizes and normalization schemes. The results show that 2×2 blocks and l^2-norm perform well.

Therefore, an image is represented as a set of keypoint with its local feature and importance $\{(f_1, w_1), (f_2, w_2) \ldots, (f_n, w_n)\}$, in which there are n regions detected as keypoints. $f_i \in R^m$ is the local feature, in which m is the dimension of the local features. The importance w_i is initialized as the value of saliency of the keypoint.

3 Local-SRC Algorithm

In the following, we give a detailed description of the Local-SRC algorithm.

1) Preparing dataset. Randomly select a certain number of images per category as the training set, with the remaining as the testing set.

2) Computing of training feature matrix A. For every training image from category i, load the n_s most important local features f_l, $l = 1, \cdots, n_s$. Local features of images belonging to the same category form the sub-matrix A_i. Given training set from k categories, matrix A is composed of every sub-matrix A_i, $A = [A_1, A_2, \cdots, A_k]$; $A \in R^{m \times n}$, in which m is the dimension of the local features, n is the total number of local features loaded in the training set.

3) Solving the optimization problem. For the given test image, load the local feature y . Solve the l^1-minimization problem in (1) or (2).

$$\widehat{x}_1 = \mathbf{argmin} \|x\|_1 \quad subject\ to\ Ax = y\ . \tag{1}$$

$$\widehat{x}_1 = \mathbf{argmin} \|x\|_1 \quad subject\ to\ \|Ax - y\|_2 \leq \epsilon\ . \tag{2}$$

4) Computing of the residual between y and its estimation for every category.. Let $\delta_i(\widehat{x}_1) \in R^n$ keep only nonzero entries in $\widehat{x}_1$ that are associated with category i. We can approximate the local feature y of the test image as $\widehat{y}_i = A\delta_i(\widehat{x}_1)$, using only the coefficients of which correspond to category i. For every category, compute residuals $r_i = \|y - A\delta_i(\widehat{x}_1)\|_2$ for $i = 1, 2, \ldots, k$.

5) Saving the category label of the local feature y of the test image. The local feature y is assigned to the category i that has the minimum residual between y and $\widehat{y}_i$.

Steps 3) to 5) are repeated for every local feature of the test image. The final label of the test image is assigned to the category that is voted by most local features.

4 Base Learning for Local-SRC Algorithm

In this section, we propose a base learning algorithm for Local-SRC. Corresponding to the region importance and learning algorithm [12], we define the keypoint importance which is initialized as the saliency value of the keypoint. Given a training dataset, the keypoint importance can also be learned through a similar

way as the region importance. The keypoint importance is used to select the local features of the most important keypoints to form the training matrix A.

The basic assumption is that the important keypoint should have as more similar keypoints as possible from the same category. At the same time, the important keypoint cannot have many similar keypoints from the whole image dataset.

Suppose there are N images $\{I_1, I_2, \ldots, I_N\}$ in the training set. From category c, there are N_c images which are denoted as ${I_c}^+ = \{{I_1}^+, {I_2}^+, \ldots, {I_{Nc}}^+\}$. For every keypoints K_i from images in ${I_c}^+$, the Keypoint Frequency (KF) is defined as

$$KF(K_i) = \sum_{j=1}^{N_c} s(K_i, {I_j}^+). \tag{3}$$

If there is a keypoint in image ${I_j}^+$ which is similar as the keypoint K_i, then $s(K_i, {I_j}^+) = 1$; otherwise $s(K_i, {I_j}^+) = 0$. The two keypoints are similar if the Euclidean distance between the local features of the two keypoints is below a designated threshold ε_p. Therefore, the KF represents the frequency of a keypoint occurred in the images from the same category. On the other hand, if a keypoint also emerges many times in the images from other categories, then the discriminative capability of the keypoint is very low. Analogous to the Inverse Image Frequency (IIF) of a region defined in [12], we define the IIF for the keypoint as the equation (4).

$$IIF(K_i) = \log\left(N \Big/ \sum_{j=1}^{N} s(K_i, I_j)\right). \tag{4}$$

As a result, the keypoint importance can be defined as

$$KI(K_i) = KF(K_i) \times IIF(K_i). \tag{5}$$

Since we only use the keypoint importance to select the keypoints from high importance to low importance as the bases of the training matrix A, normalization of the keypoint importance is not necessary.

For every keypoint in the images from the training set, the keypoint importance is calculated by equation (5). The n_s most important local features from every training image are loaded to serve as the bases of the training matrix A.

5 Experiments

5.1 Experiment Dataset

From Caltech 101 dataset [13], we select 25 object categories in which images from the same category do not vary greatly, for example, accordion, airplanes, leopards, pagoda, scissors etc. Each category contains 30 images. The task is to recognize object on the 25 object categories dataset.

5.2 Comparison of Different Descriptors for Local-SRC

In the experiments, salient region detector is used to find keypoints for every image. Two kinds of descriptors are used to represent the keypoints detected. One is 72-dim HOG feature, with the signed orientation ($0^o - 360^o$) quantized to 18 bins and 2×2 blocks division. The other is pixel values of salient regions sampled to the designated size, from$7 \times 7, 8 \times 8, 9 \times 9, 10 \times 10$ to 11×11.

Local features from the training set compose the training feature matrix A for Local-SRC algorithm. For every local feature of the test image, we solve the optimization problem (2) with the error tolerance $\epsilon = 0.05$. For the consideration of computational complexity, we only keep the 10 most important regions for training and testing. The keypoint importance is initialized as the value of saliency.

Fig. 1 and Fig. 2 give performance comparison of different descriptors for Local-SRC. The results show that performance increases slowly as the size of image patch changing from 7×7 to 11×11. Even with 11×11 image patch and 121-dim feature, the mean precision is about 10% lower than 72-dim HOG feature. The lower performances of the sampled pixel values may be caused by no alignment of keypoints detected by salient region detector. On the other hand, HOG calculates gradients on every block divided, which are not very sensitive to pixel alignment. As a result, HOG is selected as the keypoints descriptor in the following experiments.

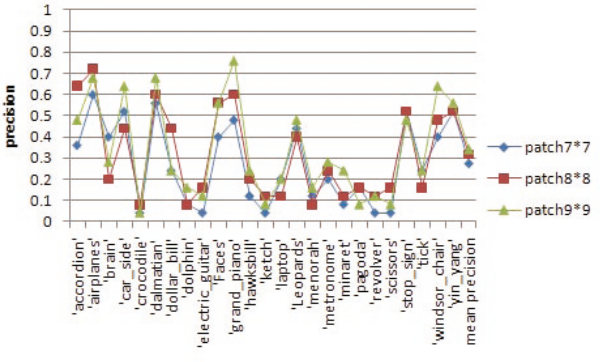


Fig. 1. Performance comparison of different local descriptors (1)

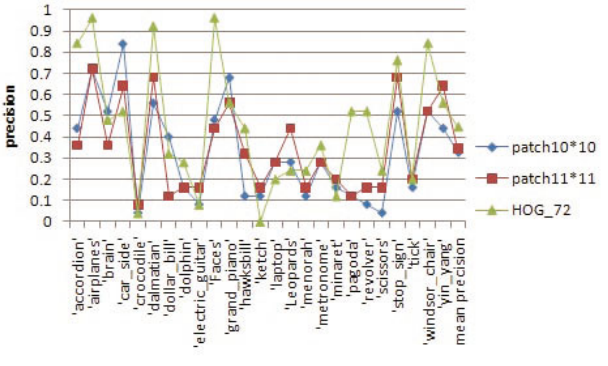


Fig. 2. Performance comparison of different local descriptors (2)

5.3 Comparison of Local-SRC and Local-SRC with Base Learning

In the experiments, salient region detector and HOG descriptor are adopted. The keypoint importance is initialized as the value of saliency. After the procedure of base learning, the keypoint importance is updated by equation (5). The threshold of Euclidean distance between local features of two keypoints to be similar $\epsilon_p = 0.4$.

Performance comparison between Local-SRC algorithm and Local-SRC with base learning (Local-SRC_BL) is shown in Fig. 3, under the condition of 5 training images per category. In order to verify the effectiveness of the base learning algorithm, we experiment with Local-SRC_10, Local-SRC_5 and Local-SRC_BL_5. The suffix indicates the number of local features per training image. The average precisions over the 25 object categories are 0.448, 0.424, and 0.464 respectively. The experimental results shows that with base learning algorithm, the precision of Local-SRC_BL using only 5 most important local features per training image outperforms that of Local-SRC using 10 most salient local features. At the same time, the scale of the training matrix is decreased and computational complexity is reduced.

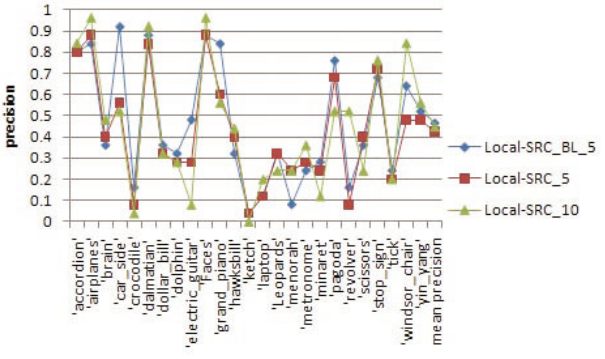


Fig. 3. Performance comparison between Local-SRC and Local-SRC_BL

There is great variation of performance among different object classes. We try to give an explanation why the Local-SRC performs well on some of the objects but not on the others. The Local-SRC algorithm performs well on the categories, which are shown in table 1.

A common property of images from these categories is that keypoints detected by salient region detector have strong discriminative capability. For example, spots on Dalmatian are very distinctive and the outputs of sparse representation of local features are almost identical for the most important regions. Another example is images from face category. Although there are not many keypoints detected by salient region detector in one image, the detected keypoints are usually located on the eyes and hair of the forehead of a person. For different people with different background, these salient regions have very similar features.

The classifiers do not perform well on the objects, such as crocodile, ketch and laptop. Images from these categories do not have obvious textures or distinctive

Table 1. Object categories with good performances

Object category	Precision		
	Local-SRC_BL_5	Local-SRC_5	Local-SRC_10
accordion	0.80	0.80	0.84
airplane	0.84	0.88	0.96
car side	0.92	0.56	0.52
Dalmatian	0.88	0.84	0.92
face	0.88	0.88	0.96
grand piano	0.84	0.60	0.56
pagoda	0.76	0.68	0.52

patterns. In other words, the local features cannot represent the whole structure of objects. It has its limitations in these cases.

6 Conclusion

Local features based sparse representation applied to generic image classification is presented in this paper. Keypoints with different descriptors are used as the bases of the training matrix and test samples. A base learning algorithm is also presented to select the most important keypoints for every training image.

Performances of different descriptors have been compared, which demonstrate that HOG outperforms the feature of sampled pixel values. With salient region detector and HOG descriptor, experiments have been done so that the effectiveness of the base learning algorithm can be verified. The experimental results show that the Local-SRC can achieve good performance on the image classes when the keypoints based local features have strong discriminative capability. The base learning algorithm is effective for improving the performance and reducing the computational complexity.

References

[1] Mikolajczyk, K., Tuytelaars, T., Schmid, C., et al.: A comparison of affine region detectors. International Journal of Computer Vision (2005)
[2] Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Trans. on Pattern Analysis and Machine Intelligence 27(10), 1615–1630 (2005)
[3] Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: IEEE Conf. Computer Vision and Pattern Recognition, pp. 264–271 (2003)
[4] Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: ECCV International Workshop on Statistical Learning in Computer Vision (2004)
[5] Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: A comprehensive study. International Journal of Computer Vision 73(2) (2007)
[6] Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: IEEE Conf. Computer Vision and Pattern Recognition, pp. 2169–2178 (2006)

[7] Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: IEEE Conf. Computer Vision and Pattern Recognition (2009)
[8] Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. IEEE Trans. on Pattern Analysis and Machine Intelligence 31(2), 210–227 (2009)
[9] Zuo, Y., Zhang, B.: General image classifications based on sparse representation. In: IEEE International Conference on Cognitive Informatics (2010)
[10] Kadir, T., Brady, M., Zisserman, A.: An affine invariant salient region detector. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 228–241. Springer, Heidelberg (2004)
[11] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Conf. Computer Vision and Pattern Recognition, vol.1, pp. 886–893 (2005)
[12] Jing, F., Li, M.J., Zhang, H.J., Zhang, B.: Relevance feedback in region-based image retrieval. IEEE Trans. on Circuits and Systems for Video Technology, special issue on audio and video analysis for multimedia interactive services 14(5), 672–681 (2004)
[13] Caltech 101 dataset, http://www.vision.caltech.edu/Image_Datasets/Caltech101

Where-What Network with CUDA: General Object Recognition and Location in Complex Backgrounds

Yuekai Wang[1], Xiaofeng Wu[1,4], Xiaoying Song[2],
Wengqiang Zhang[2], and Juyang Weng[2,3]

[1] Department of Electronic Engineering, Fudan University, Shanghai, 200433, China
[2] School of Computer Science, Fudan University, Shanghai, 200433, China
[3] Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824, USA
[4] Ventural Laboratory, Kyoto Institute of Technology, Kyoto, 606-8585, Japan

Abstract. An effective framework for general object recognition and localization from complex backgrounds had not been found till the brain-inspired Where-What Network (WWN) series by Weng and coworkers. This paper reports two advances along this line. One is the automatic adaptation of the receptive field of each neuron to disregard input dimensions that arise from backgrounds but without a handcrafted object model, since the initial hexagonal receptive field does not fit well the contour of the automatically assigned object view. The other is the hierarchical parallelization technique and its implementation on the GPU-based accelerator using the CUDA parallel language. The experimental results showed that automatic adaptation of the receptive fields led to improvements in the recognition rate. The hierarchical parallelization technique has achieved a speedup of 16 times compared to the C program. This speed-up was employed on the Haibao Robot displayed at the World Expo, Shanghai 2010.

Keywords: WWN; CUDA; GPU; Adaptive receptive field.

1 Introduction

Up to now, the general object recognition in cluttered backgrounds is still a challenging topic although a lot of tries including the methods of conventional computer vision have been done by many researchers. The appearance-based feature descriptors are quite good in object shape selectivity but no satisfying to the object transformations; the histogram-based descriptors, for example, the SIFT features, show great tolerance to the object transformations but are incomplete in the sense that they do not take all useful information in trying to achieve certain invariance using a single type of handcrafted feature detectors [3]. Compared to these artificial vision systems, human vision systems can accomplish such tasks quickly. Therefore, to create a proper network by simulating the human vision systems is thought as one possible approach to address this open yet important vision problem.

In recent decades, with the advances of the studies on object recognition in visual cortex [8] in physiology and neuroscience, several biologically-inspired network models are proposed. One famous model is HMAX, introduced by Riesenhuber and Poggio

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 331–341, 2011.

[7]. It is based on hierarchical feedforward architecture similar to the organization of visual cortex. It analyzes the input image via Gabor function and builds an increasingly complex and invariant feature representation by maximum pooling operation [9]. HMAX is a cortex-like model of the 'what' pathway, only simulating the ventral pathway in primate vision system. The location information is lost.

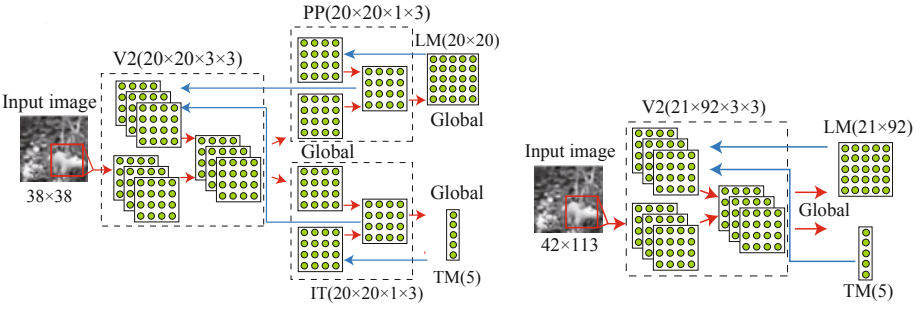
However, recognition is more than the mere detection of a specific object. Everyday, vision solves the problem of "what is where". Also, models mimicking both ventral pathway and dorsal pathway are proposed. One model is Where-What Network (WWN) introduced by Juyang Weng and co-workers [2]. It is a biologically plausible developmental model which is designed to integrate the object recognition and attention (i.e., what and where information in the ventral stream and dorsal stream respectively) interactively for any unspecific task by using both feedforward (bottom-up) and feedback (top-down) connections. Furthermore, WWN develops the features through a Hebbian learning which is very useful for intelligent robots. Up to now, four versions of WWNs have been proposed. WWN-1 [2] can realize object recognition in complex backgrounds performing in two different selective attention modes: top-down position-based which finds a particular object given the location information and top-down object-based which finds the location of the object given the type, but only 5 locations were tested. WWN-2 [1] can additionally perform in the mode of free-viewing, realizing the visual attention and object recognition without the type or location information and all the pixel locations were tested. The third version WWN-3 [4] can deal with multiple objects in natural backgrounds using arbitrary foreground object contours, not the square contours in WWN-1. WWN-4 used and analyzed multiple internal areas [5].

However, for the above versions of WWN, various backgrounds are a serious problem which also exists in other approaches. In real applications, the object contours are arbitrary while the receptive fields are usually regular (e.g., square) in the image scanning. Thus, the leak of pixels of backgrounds into the receptive field is hardly to be avoided which may produce distracter-like patterns. Additionally, training time is another problem. With general CPU-based implementation, it may spend hours or even days to train a network if the size of the input image meets the demands of real application. Considering visual processing in cortex is collective parallel, which means the operations of each neuron are executed concurrently, parallel processing of above models might be best way for real application. At present, multi-core GPUs become more popular which provide a hardware solution to implement parallel computation.

In the remainder of the paper, the architecture of the latest version of WWN is described in Section II. Adaptive receptive field in WWN is presented in Section III. The parallelization of network training is introduced in Section IV. Experiments and results are provided in Section V. Section VI gives the concluding remarks.

2 WWN Structure and Training/Application Procedures

Up to now, four versions of WWN have been proposed. The latest version WWN-4 has the same structure of WWN-3 but mainly concentrate on multiple internal areas. So, in this section, the newest structure of both WWN-3 and WWN-4, and its detailed implementation including network training and recognition is reviewed firstly.



(a) The original architecture (b) The modified architecture

Fig. 1. Illustration diagrams of the WWN architecture

The architecture of WWN-3/WWN-4 is illustrated in Fig. 1(a), in which there are three areas, V2, IT/PP and motor (the stream from V2, through PP, and to PM corresponds to the dorsal pathway and the stream from V2, through IT, and to TM corresponds to the ventral pathway in human vision systems).

Each neuron in V2 area has a local receptive field from the retina (i.e., input image) which perceives $a \times a$ area of the input image. The distance of the two adjacent receptive field centers in horizontal or vertical directions is 1 pixel. Suppose the size of the input image is $w \times h$, and the depth of V2 is c (c layers), therefore, totally $n = (w - a + 1) \times (h - a + 1) \times c$ V2 neurons can cover the entire input image.

After perception, each neuron in V2 will generate the pre-response $z^b_{i,j}(t)$, which can be computed as follows:

$$z^b_{i,j}(t) = \frac{\mathbf{w}^b_{i,j}(t) \cdot \mathbf{x}_{i,j}(t)}{\left\|\mathbf{w}^b_{i,j}(t)\right\| \left\|\mathbf{x}_{i,j}(t)\right\|} \tag{1}$$

where $\mathbf{w}^b_{i,j}(t)$ are the bottom-up weights of the neuron (i, j) and $\mathbf{x}_{i,j}(t)$ are the bottom-up local input (i.e., the $a \times a$ area perceived by neuron (i, j)).

Top-down supervision is essential in the training process and is omitted in the recognition process. It imposes neurons in $3 \times 3 \times c$ area centered at the training position (i.e., the corresponding neuron just covers the object) and suppress the others.

$$z^t_{i,j}(t) = \begin{cases} 1 \text{ if } i, j \in R \\ 0 \text{ otherwise} \end{cases} \tag{2}$$

where $z^t_{i,j}(t)$ is the top-down response of the neuron (i, j), R denotes $3 \times 3 \times c$ area centered at the training position.

With pre-response and top-down supervision input, the paired-response $z^p_{i,j}(t)$ is

$$z^p_{i,j}(t) = \alpha z^b_{i,j}(t) + (1 - \alpha) z^t_{i,j}(t) \tag{3}$$

here $z^p_{i,j}(t)$ is the paired-response of the neuron (i, j) in V2, α is a weight to control the contribution by the bottom-up input versus top-down supervision. In our experiment, $\alpha = 0.25$ in the training process and $\alpha = 1$ in the recognition process.

Lateral inhibition among neurons in the same layer is used to obtain the best feature of the training object. In WWN-3/WWN-4, top-k competition is applied to simulate the lateral inhibition which effectively suppresses weak-responding neurons (measured by paired-response). Usually top-k competition is realized by sorting the paired-responses in the descending order and normalizing the top k values while setting all the others to zero (i.e., the neurons with top k values can fire). The response $z'_{i,j}(t)$ after top-k is

$$z'_{i,j}(t) = \begin{cases} z^p_{i,j}(t)(z_q - z_{k+1})/(z_1 - z_{k+1}) & \text{if } 1 \le q \le k \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where z_1,z_q and z_{k+1} denotes the first, qth, $(k+1)$th paired-responses after sorted in descending order respectively.

Finally, the bottom-up weights $\mathbf{w}^b_{i,j}(t)$ of firing neurons (e.g., neuron (i,j) has fired)in V2 need to be updated by Hebbian learning as the new weights $\mathbf{w}^b_{i,j}(t+1)$ in the next loop of training. In the recognition process, the weights do not need to been updated. The Hebbian learning is described as follows:

$$\mathbf{w}^b_{i,j}(t+1) = w_1(t)\mathbf{w}^b_{i,j}(t) + w_2(t)z_{i,j}(t)\mathbf{x}_{i,j}(t) \tag{5}$$

$w_1(t)$ and $w_2(t)$ are determined by the following formula.

$$w_1(t) = 1 - w_2(t), w_2(t) = \frac{1 + u(n_{i,j})}{u(n_{i,j})} \tag{6}$$

where $n_{i,j}$ is the age of a neuron, or the times of firing of a neuron. $n_{i,j}$ is defined as

$$u(n_{i,j}) = \begin{cases} 0 & \text{if } n_{i,j} \le t_1 \\ c(n_{i,j} - t_1)/(t_2 - t_1) & \text{if } t_1 < n_{i,j} \le t_2 \\ c + (n_{i,j} - t_2)/r & \text{if } t_2 < n_{i,j} \end{cases} \tag{7}$$

where $t_1 = 20, t_2 = 200, c = 2, r = 10000$ in our experiment.

Next to V2 area, IT/PP is usually used to fuse the local features into global features and/or combine the individual features to the multi-feature set. Since only single features exist in our experiment, not shared features, IT/PP can be omitted to achieve a better performance. This has been proved in some experiments whose training curves of recognition rate is shown as Fig. 2(a) and (b).

The motor area TM/PM, output the "what" and "where" information. Each neuron in these two motor areas denotes the type of an object and its position respectively. Similar to V2, weights of the firing neurons in motor areas also need to be updated by Hebbian learning in the training process.

3 Adaptive Receptive Field

In practice, one annoying problem is the leak of background pixels in the receptive field fully covering the foreground target which is mentioned in Section I. In some cases, such a leak can seriously interfere the response of the neurons to the foreground object.

If the network can segment the foreground and the background automatically (i.e., outline the object contours), the irrelevant components (backgrounds) in the receptive fields of V2 will be neglected in the pre-response computation, which reduces the background interference in the process of object recognition. In the training stage, the appearance of each foreground object remains almost the same in any location and any training loop, that is, the foreground pixel values are almost the same, while the backgrounds may be significantly different. Statistically, the mean deviations of the foreground pixels and background pixels in receptive fields are discriminative. Thus, a new set of weights (named "trimmed weights") is introduced into V2 of WWN structure by using this characteristic as a new segmentation mechanism.

3.1 Principle of the Receptive Field Adaptation Mechanism

Assumed that for the neuron (i, j) in original WWN, the bottom-up local input is $\mathbf{x}_{i,j}(t) = (x_1, x_2, x_3, ..., x_d)$, the bottom-up weights are $\mathbf{w}^b_{i,j}(t) = (v_1, v_2, v_3, ..., v_d)$, the trimmed weights are $\mathbf{w}^m_{i,j}(t) = (w_1, w_2, w_3, ..., w_d)$ and the trimmed factors, $\mathbf{f}_{i,j}(t) = (f_1, f_2, f_3, ..., f_d)$, are the variables introduced into WWN structure by new receptive field adaption mechanism. f_m $(m = 1, 2, 3..., d)$ is used to evaluate the deviation of the corresponding pixel. Assumed that the $\bar{f}$ denotes the average value of $f_1, f_2, f_3, ..., f_d$. The trimmed factor f_m is defined as follows:

$$f_m = \begin{cases} 1 & \text{if } r < \beta_s \\ (\beta_b - r)/(\beta_b - \beta_s) & \text{if } \beta_s \le r \le \beta_b \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where $r = f_m/\bar{f}$, $\beta_s = 1.0$, $\beta_b = 1.5$.

The trimmed bottom-up input $\mathbf{x}^{'b}_{i,j}(t) = (x'_1, x'_2, x'_3, ..., x'_d)$ is defined as:

$$x'_m = f_m x_m \tag{9}$$

and the trimmed bottom-up weights $\mathbf{w}^{'b}_{i,j}(t) = (v'_1, v'_2, v'_3, ..., v'_d)$ is defined as:

$$v'_m = f_m w_m, \text{ where } m = 1, 2, ..., d \tag{10}$$

Thus the trimmed bottom-up response $z^{'b}_{i,j}(t)$ is calculated as follows:

$$z^{'b}_{i,j}(t) = \frac{\mathbf{w}^{'b}_{i,j}(t) \cdot \mathbf{x}^{'b}_{i,j}(t)}{\left\|\mathbf{w}^{'b}_{i,j}(t)\right\| \left\|\mathbf{x}^{'b}_{i,j}(t)\right\|} \tag{11}$$

Therefore f_m dynamically determines whether the pixel m provides a full supply, no supply, or in-between to the pre-response of the corresponding V2 neuron.

3.2 Trimmed Weights Learning

Similar to the bottom-up weights of the neurons, the trimmed weights also need to be updated by Hebbian learning as follows:

$$w_m(t) = \begin{cases} 1/\sqrt{12} & \text{if } t \le t_1 \\ w_1(t)w_m(t-1) + w_2(t) \mid x_m - v_m \mid & \text{otherwise} \end{cases} \tag{12}$$

where $w_1(t)$ and $w_2(t)$ are the same definitions in formula (6) and (7). x_m and v_m denotes the local input and the bottom-up weight individually. If the pixel m belongs to the foreground region, $\mid x_m - v_m \mid$ is nearly zero; conversely, if the pixel m belongs to the background region, $\mid x_m - v_m \mid$ is relatively large. In application, usually we set $t_1 = 20$, which means the trimmed weights begin to be updated after the corresponding neurons have fired 20 times in order to wait the bottom-up weights of the neurons to get good estimates of the training objects.

4 Parallelization of Network Training

In the example of WWN-3, a is 19, w and h are 38 and c is 3. It is meaningless for applications on HAIBAO robot with such small input images. Usually, the size of images captured by cameras is 320×240 or 640×480 in general application. The necessary memory required by WWN-3 can be roughly estimated following (suppose all the data in computation using floating point type).

$$\text{memory} \approx 4c[(h-a+1)(w-a+1)]^2 \text{ (byte)} \tag{13}$$

For the 320×240 image, h is 320, w is 240, assumed that c and a is still 3 and 19, then about 54GB memory is needed. That is a huge amount even for main memory on PC unless we use some compression techniques. Considering that usually, the memory on a graphics card is no more than 2GB and the situation of the demonstration in 2010 World Expo (objects only can be learnt in limited area), we finally set the size of input images 42×113. Thus V2 contains $(42-19+1) \times (113-19+1) = 2280$ neurons each layer (i.e., 6840 neurons totally) which is about 5.7 times that of the original network. After expanding the size and omitting IT/PP, the modified network is shown as Fig. 1(b) in which V2 connects to the motor layers directly.

In WWN, the in-place learning algorithm is used so that each neuron is responsible for its own learning through interactions with other neurons in the same layer. Besides, as the number of the learning objects, locations and variations within each type (e.g., rotation, illumination, size) increase, the architecture and the algorithm of the network does not need to be modified but simply increase the number of neurons. Thus, the design of WWN fit to be parallelized.

Further, in WWN, recognition spends much less time (in milliseconds) than training does so that it is not necessary to do the parallelization. Therefore, we only focus on the parallelization of the WWN training. The network training mainly includes three parts, pre-response computation of the neurons in V2, top-k competition for all the neurons in V2 and weight updates of the winners via Hebbian learning. Among them, the pre-response computation and Hebbian learning on each neuron is done independently so that these two processes can be performed in parallel. For top-k competition, although it has to be done after the pre-response computation in sequential,

the core operation is essentially sorting which is a well-known NP problem in computer science and so many parallel algorithms can be chosen. Here, the bitonic sorting network, a famous sorting algorithm designed specially for parallel machines [2] is adopted. Therefore, in detailed algorithm level, the WWN can be also parallelized.

In implementation, as one of attractive parallel processing techniques on PC recently, GPU-based parallel computation is developing rapidly. GPU has evolved into a highly parallel, multithreaded, many core processors (much more than the CPU cores) with very high memory bandwidth. Among the GPU programming techniques, CUDA (Compute Unified Device Architecture), a general purpose parallel computing architecture designed by NVIDIA, is widely accepted by most programmers. In this research, we use GPU/CUDA in parallelization of WWN.

4.1 Parallelization of Pre-response Computation

From formula (1), obviously the inner product of $\mathbf{w}_{i,j}^{b}(t)$(bottom-up weights) and $\mathbf{x}_{i,j}(t)$(bottom-up input) is the major operations in the pre-response computation of each neuron. In our experiments, this computation is assigned to 6840 threads grouped in 27 blocks, that is to say, each thread executes the inner product for one neuron in V2 so that these 6840 same operations can be executed concurrently to save a lot of time.

4.2 Parallelization of the Hebbian Learning

Similar to the parallelization of pre-responses computation, both bottom-up and top-down weights updated by Hebbian learning are assigned to threads too. The only difference is that in this part every thread within a block corresponds to one or two elements of a vector stored in shared memory (low latency) instead of all elements of a vector stored in global memory (high latency) because of the small amount of data. Additionally, considering the branch structure will greatly affect the acceleration of parallel processing, a lookup table of $w_2(t)$ for formula (6) and (7) is transferred from CPU to GPU before the network training.

4.3 Parallelization of the Top-k Competition

The read and write operation is frequent in sorting, so using low latency of the memory units as much as possible is necessary. In CUDA, shared memory is the best candidate. As only the threads within the same block (256 threads per block in the GPU of our experiment) can communicate through shared memory, groups of 256 numbers (6840 numbers in 27 groups totally) are sorted in ascending order independently. Then take out the first eight numbers in each block (i.e., $8 \times 27 = 216$) and fill enough zeros to complete 256 numbers. In the same way, assign these numbers to a single block with 256 threads and sort these numbers in ascending order. Finally, select the top k numbers (i.e., 3) from 256 numbers. As k is 3, 2, and 1 in our experiment, selecting 8 numbers from each block is enough.

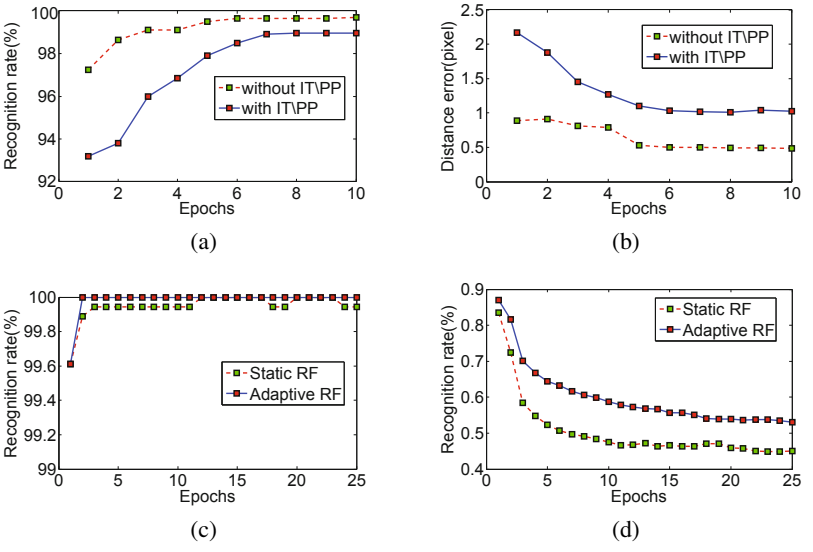


Fig. 2. (a) Recognition rates in 10 epochs with/without IT/PP (b) Distance errors in 10 epochs with/without IT/PP (c) Recognition rates in 15 epochs with adaptive receptive field (RF) and static RF (d) Distance errors in 15 epochs with adaptive RF and static RF

5 Experiments and Results

The backgrounds and foregrounds in the experiments are selected from 13 natural images[1] and the MSU 25-objects dataset [6] respectively.

5.1 Experiments of Receptive Field Adaption

In order to evaluate the effectiveness of receptive field adaption mechanism, the recognition performances of the network with/without this new auxiliary mechanism are compared. Furthermore, the trimmed factors are visualized to observe the behaviors in the network with sufficient/limited resources. The training objects with the size of 19×19 and input images with the size of 38×38 are shown as Fig. 3 (a) and (b).

In WWN, the depth of V2 determines the available feature storage resources in each certain position. For example, in our experimental WWN, the depth of V2 is 3 which means the network can store 3 appearance features for each local receptive field. Thus, if 5 objects need to be learned, there is $(5-3)/5 = 40\%$ resources shortage, called 'limited resource'. Correspondingly, 'sufficient resources' means each neuron in V2 has enough memory space which can store all the objects to be learned. Compared with the objects contours in super resolution (e.g., by Adobe Photoshop) shown as Fig. 3(c), for the network with sufficient resources, the contours of the foreground can be outlined roughly while the effect became a little worse when the network has limited resources as Fig. 3(d) and (e) show. In the case of limited resources, the V2 neurons will remember

[1] Available from http://www.cis.hut.fi/projects/ica/imageica/

Fig. 3. (a) The training samples (b) The input images (c) Object contours in super resolution (by Adobe Photoshop) (d) Visualization of trimmed factors for WWN with sufficient resources (e) Visualization of trimmed factors for WWN with limited resources

the objects as many as possible with the help from their neighbors. Doubtlessly, such help from adjacent neurons will result in precision loss of the object positions.

Fig. 2(c) and (d) shows the performance of the network with/without receptive field adaptation mechanism including recognition rate and position error. It is found that with the new auxiliary mechanism, the recognition rate is a little better than that without the mechanism.

5.2 Parallelization of WWN

The experiments are carried on a machine equipped with an Intel Core 2 Duo 2.99GHz CPU with 3GB memory and a GeForce GT 240 with 1GB memory which possesses 12 Stream Multiprocessors composed of 96 Stream Processors. In order to assess the parallelization effect of the WWN, the network training times in one epoch of GPU-based WWN using CUDA and CPU-based WWN using C language are compared in real environment.

For network training time, the GPU-based WWN using CUDA only spent 91.719 seconds while the CPU-based WWN using C language spent about 1423.235 seconds. Apparently, the GPU-based WWN using CUDA achieved a 16 times acceleration.

For recognition performance, in ideal condition (i.e., the foreground is overlapped on the background), the distance error and recognition rate are shown as Fig. 4. Note that the size of foreground and background is 19×19 and 42×113 respectively, this indicates that at least about 92.4% of the input image is background. So the performance is satisfying. In real environment (i.e., the input image is captured by camera on the robot exhibited in 2010 World Expo in Shanghai), some testing examples in free viewing

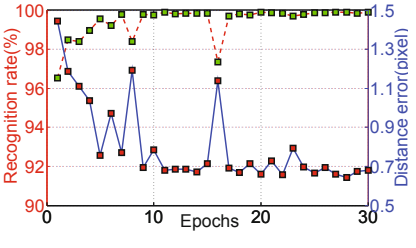


Fig. 4. The performance of the network parallelized with CUDA in 30 epochs

Fig. 5. Examples from robotics demonstration in 2010 Shanghai Expo using modified WWN

mode are shown in Fig. 5. It is found that the network can be tolerant to little changes in scale, illumination and viewpoint. But when those changes became significant, the recognition performance became poor too.

6 Conclusion and Future Work

The WWN framework is moving toward real-time generic object recognition and localization in increasingly more natural settings. The adaptive receptive fields seem to play a positive role in dealing with objects of various shapes in complex backgrounds. The hierarchical parallelization technique reported here takes advantage of the inexpensive GPU computing engines to reach a real-time or nearly real-time speed, paving the way toward real-time learning. While some results on view variance with WWN-3 have been reported before, our work on scale variance has been drafted and will appear elsewhere. Future work includes experimental studies on illumination variation and increased variety of objects, which the WWN framework seems to be able to deal with.

References

1. Ji, Z., Weng, J.: WWN-2: A biologically inspired neural network for concurrent visual attention and recognition. In: Proc. IEEE International Joint Conference on Neural Networks, Barcelona, Spain, July 18-23, pp. 1–8 (2010)
2. Ji, Z., Weng, J., Prokhorov, D.: Where-what network 1: "Where" and "What" assist each other through top-down connections. In: Proc. IEEE International Conference on Development and Learning, Monterey, CA, August 9-12, pp. 61–66 (2008)
3. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proc. International Conference on Computer Vision, Kerkyra, September 20-27, vol. 2, pp. 1150–1157 (1999)
4. Luciw, M., Weng, J.: Where-what network 3: Developmental top-down attention for multiple foregrounds and complex backgrounds. In: Proc. IEEE International Joint Conference on Neural Networks, Barcelona, Spain, July 18-23, pp. 1–8 (2010)
5. Luciw, M., Weng, J.: Where-what network-4: The effect of multiple internal areas. In: Proc. IEEE International Joint Conference on Neural Networks, Ann Arbor, MI, August 18-21, pp. 311–316 (2010)
6. Luciw, M., Weng, J.: Topographic class grouping with applications to 3d object recognition. In: Proc. IEEE International Joint Conference on Neural Networks, Hong Kong, June 1-6, pp. 3987–3994 (2008)

7. Riesenhuber, M., Poggio, T.: Hierachical models of object recognition in cortex. Nature Neuroscience 2(11), 1019–1025 (1999)
8. Roelfsema, P.R.: Cortical algorigthms for perceptual grouping. Annual Review of Neuroscience 29, 203–227 (2006)
9. Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., Poggio, T.: Robust object recognition with cortex-like mechanisms. IEEE Trans. Pattern Analysis and Machine Intelligence 29(3), 411–426 (2007)

Fast Human Detection Based on Enhanced Variable Size HOG Features

Jifeng Shen[1], Changyin Sun[1], Wankou Yang [1], and Zhongxi Sun[1,2]

[1] School of Automation, Southeast University, Nanjing 210096, China
[2] College of Science, Hohai University, Nanjing 210098, China
cysun@seu.edu.cn

Abstract. In this paper, we proposed an enhanced variable size HOG feature based on the boosting framework. The proposed feature utilizes the information which is ignored in quantization gradient orientation that only using one orientation to encode each pixel. Furthermore, we utilized a fixed Gaussian template to convolve with the integral orientation histograms in order to interpolate the weight of each pixel from its surroundings. Either of the two steps have an important effect on the discriminative ability of HOG feature which leads to increase the detection rate. Soft cascade framework is utilized to train our final human detector. The experiment result based on INRIA database shows that our proposed feature improves the detection rate about 5% at the false positive per window rate of 10^{-4} compared to the original feature.

Keywords: human detection, soft cascade, integral HOG feature, enhanced integral HOG feature.

1 Introduction

In the last few years, human detection in still image or video stream has received a lot of attention [1, 2]. It is widely used in visual surveillance, behavior analysis or automated personal assistance field. Detecting humans is a challenge work due to several factors such as articulation, difficult contract and background, occlusion and illumination conditions especially in outdoor scenes.

Recently, Dalas & Triggs presented a human detection method which made use of the Histogram of Orientation features[3] to model human silhouette and apply linear svm to classify the stacked 3780 dimensional features which is generated by dense sampling. This method is very effective in detecting upright fully visible humans and robust to slight deformation. But the evaluation speed is more than 500ms to scan a 240x320 image that only have 1000 detection windows. This baffled its application in many of the field which needs real-time running speed. The key idea to improve running speed is based on Adaboost framework or makes use of simplified HOG features. Zhu [4]proposed a cascade structure based human detection algorithm which using Adaboost to selection discriminative Blocks to exclude negative patches at a low cost at 5-30 fps. Chen [5] presented a meta cascaded structure to combine heterogeneous features and result in a effective detector at 6-8fps. Other method used

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 342–349, 2011.

EOH[6] and Haar features combination[7], Covariance features[8], motion information[9, 10], integral channel features[11] and so on.

In this paper, we propose an improved feature named enhance variable size HOG (EVSHOG) feature which utilizes the information ignored in quantization of gradient orientation and uses Gaussian template to approximate the interpolation between different pixels in order to improve the discriminative of integral HOG features. Furthermore, we apply soft cascade structure to connect weak classifiers which fully make use of the information flow between weak classifiers.

2 Integral HOG Feature

Dalas & Triggs proposed the HOG feature [3] and successfully applied to object detection, but it is rather slow to calculate and cannot use in real-time task. In order to deal with this problem, Zhu proposed a variable size HOG (VSHOG) features[4], which uses one gradient orientation to encode each pixel and ignored the spatial interpolation which is not fit well to integral histogram calculation at the sacrifice of decreasing the detection rate. The gradient histogram for separated channels is shown in figure 1, we only quantize four orientations in 4x4 pixels image for simplicity. Firstly, image gradient in horizontal and vertical orientation are calculated by sobel operator and gradient orientation is calculated by Eq. 2.

$$M(x, y) = \sqrt{I_x^2 + I_y^2} \tag{1}$$

$$O(x, y) = \arctan(\frac{I_y}{I_x}) \tag{2}$$

where $O \in [0,180]$ and [0,180] is equally quantized into N=4 bins.

Secondly, the angle of gradient orientation is trimmed to the nearest bin and the energy of this orientation is represented by Eq. (1).

In order to get fast calculation histogram in image, integral histogram technique[12] is utilized which is widely used in computer vision field . The integral histogram can be defined in Eq. (3). The procedure of calculating integral HOG feature is shown in figure 2 and we can get the histogram of image using only three times float operation in each channel of histogram H_b . so it's very fast to get any sub-histogram in an image.

$$IHist(x, y) = \bigcup_{b=1}^{B} \sum_{x' \le x, y' \le y} H_b(x', y') \tag{3}$$

where B is the number of bins in histogram H and H_b is b^{th} histogram of H.

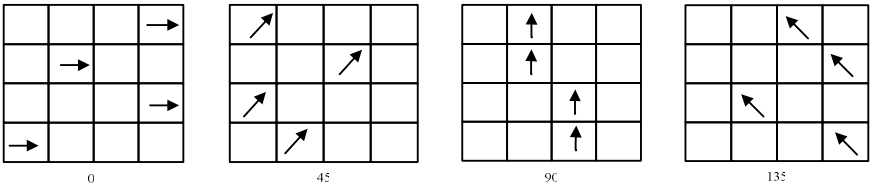


Fig. 1. Gradient histogram in four channels

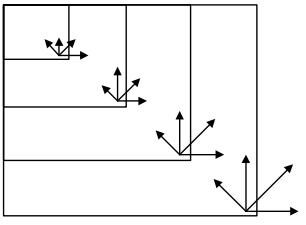


Fig. 2. Integral HOG

3 Enhanced Variable Size HOG

In order to deal with the problem of ignorance in quantizing the gradient orientation and interpolation in spatial positions, we first use linear interpolation to quantize the gradient orientation of each pixel more accurately which is demonstrated in figure 3. In figure 3(b), we can see that degree 30 only trim into interval 1 using original method. Actually it is not very accurate to encode gradient information and distribute energy for each pixel, so we interpolate the degree 30 into two nearby intervals (0 and 1) and weighted by its relevant distance in order to reflect the energy information in local area more accurately. The interpolation formula is show in Eq. 4.

$$f(x+dx) = (1-dx)\bullet f(x) + dx\bullet f(x+1),\ dx \in (0,1) \tag{4}$$

where $f(x)$ is the strength of gradient in pixel x which is defined in Eq. (1).

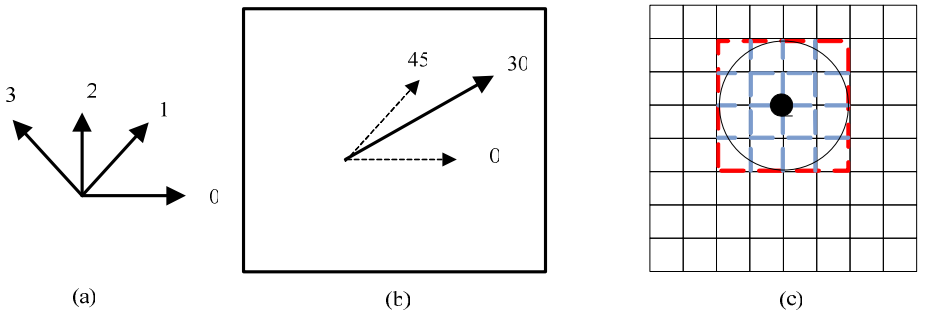


Fig. 3. Linear interpolations in quantization gradient orientation

After linear interpolation of gradient orientation, we further utilized the Gaussian template to bilinear interpolate the weight of pixel with its surrounding pixels in order to smooth the local variation of pixels and decrease the effect of noise. The bilinear interpolation is show in figure 3(c). Finally, we normalized the histogram with L2-hys technique[3] which is show in Eq.5.

$$v = \frac{v}{\sqrt{\| v \|_2^2 + \varepsilon^2}}, v_i = \max(v_i, 0.2), i = 1,..d. \quad v \in R^d \tag{5}$$

It is also mentioned in paper [3], interpolation is very important in generating robust gradient orientation histogram.

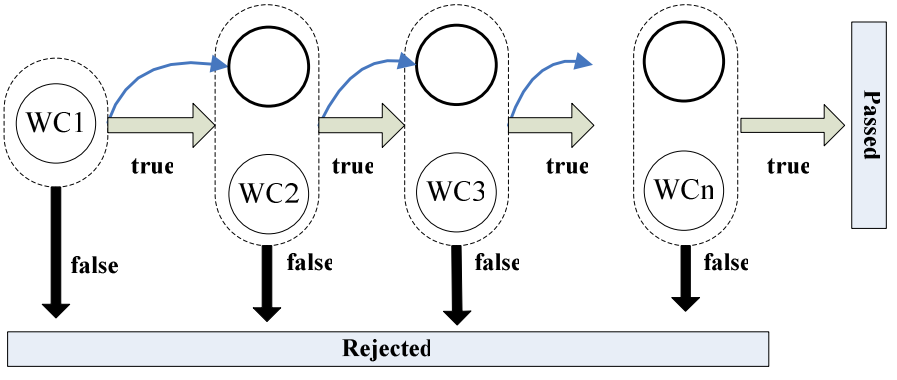


Fig. 4. Soft Cascade

4 Soft Cascades

Soft cascade is proposed by Bourdev [13] who first applied to face detection filed and gained nice performance. It is improved version based on hard cascade [14] and boosting chain [15], and it fully makes use of the information flow between connect strong classifier and exclude the negative patches as earlier as possible in order to improve the detection speed. It is a special case of boosting chain and only uses one weak classifier in each stage. The soft cascade classifier can be defined in Eq. (6):

$$C(x) = \sum_{t=1}^{T} \alpha_t f_{t,\theta_t}(x) \tag{6}$$

where T is the total number of classifier trained and $f_{t,\theta_t}(x)$ is the weak classifier chosen in stage t and θ_t is the threshold to refuse negative patches in stage t. Tuning the parameter θ_t can calibrate the classifier to get a satisfied speed and accuracy. The parameter of θ_t can be modeled as an exponential function family which is defined in Eq. (7):

$$\theta_t = \begin{cases} ke^{-\alpha(1-\tau)} & \alpha < 0 \\ ke^{\alpha(1-\tau)} & \alpha \geq 0 \end{cases} \tag{7}$$

where $\tau = t/T$ and k normalizes the vector sum to satisfy the target detection rate and α is the free parameter for the function family.

The structure of soft cascade is shown in figure 4, where WC indicates the selected weak classifier and the dark arrow means the cumulated confidence value which is generated by each weak classifier. Soft cascade has the advantage of using lesser weak classifier and fast detection speed.

4.1 Fisher Linear Discriminative (FLD) Weak Classifier

The choice of weak classifier has an important effect in the convergence rate of boosting algorithm. For one dimensional feature such as haarlike, decision stump is an efficient choice to get an optimal threshold in $O(n \log n)$ time, where n is number of training data. But for a high dimensional feature $f \in R^d$, find a optimal need $o\left(\begin{pmatrix} n \\ d \end{pmatrix}\right)$ time which is intractable which either n and d is large. Although SVM or Neural Network can deal with classification of high dimensional features, but it is very time consuming in training. But the large number of classifier prohibits the usage of such classifier. In order to deal with high dimensional features, one of the solutions is to find a mapping function $\phi(x): R^d \rightarrow R$ to project d dimensional feature into one dimension and train classifier on this feature after projection.

An efficient classifier for high dimensional features is based on fisher criterion which intent to find a optimal projection vector to maximize the between-class distance and minimize the within-class distance simultaneously. The criterion is formulated in Eq.(8):

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w \mathbf{w}} \tag{8}$$

where $\mathbf{S}_B$ and $\mathbf{S}_w$ is between-class scatter matrix and within-class scatter matrix respectively, $\mathbf{w}$ is the projection vector.

5 Experiments

5.1 Experiment Setting

In order to validate the effectiveness of our proposed features, we conduct experiment on INRIA database which is widely used in evaluating human detector. The information of INRIA benchmark datasets is demonstrated in Table 1.

Table 1. INRIA dataset

Dataset site	http://pascal.inrialpes.fr/data/human/
Train data	2416 human annotations in 614 images 1218 non-human images
Test data	1132 human annotations in 288 images 453 non-human images
Image size	Human image are 70x134 with 3 pixels padding Non-human images are from 214x320 to 648x486

In training our human detector, 2416 human images and 2416 non-human images (randomly sampled from 1218 images) was used in the beginning, then bootrap in the 1218 images in later stage training. All the training samples are cropped into 128x64. The total number of weak classifier is T=600. We use window size=7 and sigma=2 two dimension Gaussian function to generate the template. We quantized the gradient orientation into 9 bins from [0, 180] and using L2-hys method to normalize histogram. Soft cascade[13] and Adaboost algorithm are used to train our human detector and using linear SVM to train each weak classifier.

5.2 Evaluation in INRIA Dataset

We evaluated our proposed EI-HOG features on INRIA datasets, the result is represented by false positive per window (FPPW) versus miss rate curves. The curve is defined as Eq.(9):

$$\text{Miss rate} = \frac{false\quad negatives}{true\quad positives + false\quad negatives}, \text{FPPW} = \frac{false\quad positives}{total\quad windows} \tag{9}$$

The lower of the curve is better. There different algorithms (HOG+SVM[3], Adaboost+ VSHOG [4], Haar[4] and our proposed) are compared which are shown in figure 5.

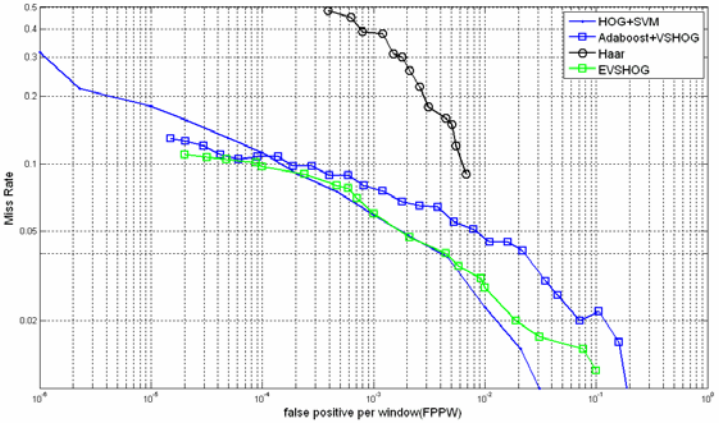


Fig. 5. Comparing the HOG+SVM, Adaboost+VSHOG, Haar and our proposed feature

From figure 5, we can see that our proposed EVSHOG feature is superior than Adaboost+VSHOG and haar features by the use of trilinear interpolation between orientation and spatial positions. It is also worth to mention that our proposed feature has little difference with Dalal's original well tuned HOG feature which means that our proposed feature can approximate the Dalal's feature very well, but is much faster because of using boosting framework at the 10-15 fps. Some of the human detection result on INRIA database is demonstrated on figure 6.

Fig. 6. Some of the human detection results on INRIA dataset

Acknowledgements

This project is supported by NSF of China (90820009, 61005008) and the Fundamental Research Funds for the Central Universities (2010B10014).

References

1. Munder, S., Gavrila, D.M.: An Experimental Study on Pedestrian Classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(11), 1863–1868 (2006)
2. David, G.: Survey of Pedestrian Detection for Advanced Driver Assistance Systems. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(7), 1239–1258 (2009)
3. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proceeding of IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 886–893 (2005)
4. Zhu, Q., et al.: Fast Human Detection Using a Cascade of Histograms of Oriented Gradients. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 1491–1498 (2006)

5. Yu-Ting, C., Chu-Song, C.: Fast Human Detection Using a Novel Boosted Cascading Structure With Meta Stages. IEEE Transactions on Image Processing 17(8), 1452–1464 (2008)
6. Levi, K., Weiss, Y.: Learning object detection from a small number of examples: the importance of good features. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 53–60 (2004)
7. Gerónimo, D., López, A., Ponsa, D., Sappa, A.D.: Haar Wavelets and Edge Orientation Histograms for On-Board Pedestrian Detection. In: Martí, J., Benedí, J.M., Mendonça, A.M., Serrat, J. (eds.) IbPRIA 2007. LNCS, vol. 4477, pp. 418–425. Springer, Heidelberg (2007)
8. Paisitkriangkrai, S., Chunhua, S., Jian, Z.: Fast Pedestrian Detection Using a Cascade of Boosted Covariance Features. IEEE Transactions on Circuits and Systems for Video Technology 18(8), 1140–1151 (2008)
9. Pers, J., et al.: Histograms of optical flow for efficient representation of body motion. Pattern Recognition Letters 31(11), 1369–1376 (2010)
10. Dalal, N., Triggs, B., Schmid, C.: Human detection using oriented histograms of flow and appearance. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 428–441. Springer, Heidelberg (2006)
11. Dollar, P., Tu, Z., Perona, P., Belongie, S.: Integral Channel Features. In: British Machine Vision Conference 2009, London, England (2009)
12. Porikli, F.: Integral Histogram: A Fast Way To Extract Histograms in Cartesian Spaces. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 829–836 (2005)
13. Bourdev, L., Brandt, J.: Robust object detection via soft cascade. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2005)
14. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2001)
15. Rong, X., Long, Z., Hong-Jiang, Z.: Boosting chain learning for object detection. In: Proceedings of the IEEE International Conference on Computer Vision (2003)

A Novel Local Illumination Normalization Approach for Face Recognition

Zhichao Lian, Meng Joo Er, and Juekun Li

School of Electrical and Electronic Engineering, Nanyang Technological University, 639798, Singapore
{LIAN0069,EMJER,LI0009UN}@ntu.edu.sg

Abstract. In this paper, different from most of existing methods, an additive term as noise is considered in the proposed method besides a multiplicative illumination term in the illumination model. Discrete cosine transform coefficients of high frequency band are discarded to eliminate the effect caused by noise. Based on local characteristic of human face, a simple but effective illumination normalization method local relation map is proposed. The experimental results on the Yale B and Extended Yale B prove the outperformance and lower computational burden of the proposed method compared to other existing methods.

Keywords: Face Recognition, Illumination Variation.

1 Introduction

Illumination variation is one of the most challenging issues in face recognition to address. In [1], differences between varying illumination conditions are proven to be more significant than differences between individuals. The existing methods can be generally classified into three categories: face and illumination modeling, illumination invariant features extraction and normalization. Compared to other two categories, the methods of normalization usually take less computational loan and require less training samples. The methods proposed in [2] and [3] are two of the most representative normalization methods. Both of them achieve convincing performances based on their experiments.

In most of existing approaches, the image under different illuminations is simply modeled as

$$f(x,y) = r(x,y) \cdot e(x,y) \tag{1}$$

where *f(x, y)* is the image gray level, *r(x, y)* is the reflectance and *e(x, y)* is the illumination. Based on the model, illumination variations are proposed to mainly lie in the low frequency band [2]. Therefore, low frequency DCT coefficients in the logarithm domain are discarded in [2] to compensate illumination variations. In [3], the noise is considered as an additive term besides the multiplicative term *e(x, y)*. The noise is taken as a constant in a local area. Furthermore, the assumption that

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 350–355, 2011.

illumination is related to low frequency band is extended to that illumination can be considered as a constant (only related to the DC component) in a small local area [3].

In this paper, an additive term as noise is considered in the illumination model. Different previous researchers [3], we propose to apply the high frequency DCT coefficients obtained in the entire image to estimate the noise. After that, a logarithm transform is taken to change the model into an additive model. With the simplified model, a simple but efficient approach, Local Relation Map (LRM), is proposed based on local characteristics of human face. The experimental results on the Yale B and Extended Yale B prove the outperformance and lower computational burden of the proposed method compared to other methods.

The rest of this paper is organized as follows. In Section 2, we introduce our illumination model and novel local illumination normalization method LRM in details. Furthermore, we prove that the LRM is illumination invariant in this section. Experimental results and discussions are presented in Section 3. Finally, conclusions are drawn in Section 4.

2 Local Illumination Normalization Technique

2.1 Face and Illumination Model

As mentioned before, most existing method simply model human face under different model as Eq. (1). With considering the noise, the model can be changed to

$$f(x,y) = r(x,y) \cdot e(x,y) + n(x,y) \tag{2}$$

where *f(x, y)*, *r(x, y)* and *e(x, y)* are still the same as those in Eq. (1), and *n(x, y)* is the additive noise. Considering the local properties of human face, the noise can be modeled as a constant as shown in [3]. However, observing the noise in an entire image, we propose to use high frequency components to model the noise. Experimental results in Section 3 will prove the validity of our assumption.

To remove the effects of noise, discrete cosine transform (DCT) is firstly applied in face images. The values of *k* dimensions of high frequency DCT coefficients are set to zeros in zigzag mode.

2.2 Local Relation Map

After the denoising described in the above section, the model is simplified into

$$f'(x,y) = r(x,y) \cdot e(x,y) \tag{3}$$

Taking logarithm transform on Eq. (3), we have

$$\log f'(x,y) = \log r(x,y) + \log e(x,y). \tag{4}$$

A human face can be treated as a combination of lots of small and flat facets [3]. In such a small facet *W*, the illumination can be considered as a constant as in [3].

Compared with the model (4), the *loge(x, y)* can be taken as a constant *A*. Therefore, for a special illumination condition, a small facet *W* can be modeled as

$$\mathrm{I}(\mathrm{x},\mathrm{y}) = \mathrm{p}(\mathrm{x},\mathrm{y}) + A, (\mathrm{x},\mathrm{y}) \in \mathrm{W} \tag{5}$$

where

$$\mathrm{I}(\mathrm{x},\mathrm{y}) = \log \mathrm{f}^{'}(\mathrm{x},\mathrm{y}) \text{ and } \mathrm{p}(\mathrm{x},\mathrm{y}) = \log \mathrm{e}(\mathrm{x},\mathrm{y})$$

Based on Eq. (5), we propose a simple illumination approach local relation map, which eliminate the effect of *A* by comparing the relation between the gray level of desired point with those of the points in the boundary of W. The details of the LRM are described as following:

1) Given a point (x, y), determine the local facet *W*. In this paper, we mainly focus on a square local facet because it is easier to implement.
2) Determine the boundary points *U* in the facet. For a square facet with size of *n*, there is only *4(n-1)* boundary points.
3) Compare the gray level of point (x, y) with those of points in *U* as

$$\mathrm{I}^{'}(\mathrm{x},\mathrm{y}) = \mathrm{I}(\mathrm{x},\mathrm{y}) - \sum_{(a,b)\in U} I(a,b) \Big/ 4(n-1) \tag{6}$$

4) After all the points on a given image are processed, the normalized image, named local relation map, is obtained.

Please note that only the boundary point in the facet will be involved in the calculation instead of all the points in the facet. The advantage is that it reduces the computational complexity from $O(n^2)$ to $O(n)$. The effect on real computational time will be discussed in the experiment section. Other advantages in adaptive size selection of the facet will be studied in further research.

2.3 Properties of Local Relation Map

Here, we will prove that the LRM is illumination invariant. Given two images of the same person I_1 and I_2, taken under different illumination conditions, for the same point (x, y), we have

$$\mathrm{I}_1(\mathrm{x},\mathrm{y}) = \mathrm{p}(\mathrm{x},\mathrm{y}) + A_1 \tag{7}$$

$$\mathrm{I}_2(\mathrm{x},\mathrm{y}) = \mathrm{p}(\mathrm{x},\mathrm{y}) + A_2 \tag{8}$$

After the calculation of the LRM, we will have

$$\begin{aligned} \mathrm{I}_1^{'}(\mathrm{x},\mathrm{y}) &= \mathrm{I}_1(\mathrm{x},\mathrm{y}) - \sum_{(a,b)\in U} I_1(a,b) \Big/ 4(n-1) \\ &= \mathrm{p}(\mathrm{x},\mathrm{y}) - \sum_{(a,b)\in U} p(a,b) \Big/ 4(n-1) \end{aligned} \tag{9}$$

and

$$\begin{aligned} I_2^{'}(x,y) &= I_2(x,y) - \sum_{(a,b)\in U} I_2(a,b) \Big/ 4(n-1) \\ &= p(x,y) - \sum_{(a,b)\in U} p(a,b) \Big/ 4(n-1) \end{aligned} \tag{10}$$

Easily, we have

$$I_1^{'}(x,y) = I_2^{'}(x,y) \tag{11}$$

This means that the LRM is unrelated to illumination conditions. Therefore, we can use the LRM for further face recognition.

3 Experimental Results and Discussions

3.1 Database

In the experiments, we use the Yale Face database B and Extended Yale Face database B as the test database. In the Yale Face database B, there are 10 persons with 64 different illumination conditions for nine poses per person [4]. In the Extended Yale Face database B, there are 16128 images of 28 persons with the same conditions as Yale B [5]. Because the main concern in this paper is on illumination variation, only 64 frontal face images per person under different illumination conditions are chosen. After combining the Extended Yale B with the Yale B except 18 corrupted images, there are 2414 images of 38 subjects named as the Completed Yale B. The images are divided into 5 subsets based on the angle between the light direction and the camera axis as other methods shown in Table 1. Because of lack of coordinates of the eyes in the Extended Yale B database, we directly use the cropped and aligned images with the size of 192×168 provided by the database [5].

Table 1. Subsets divided based on light source direction

	Subset 1	Subset 2	Subset 3	Subset 4	Subset 5
Light angle	0~12	13~25	26~50	51~77	>77
Number of images in Completed Yale B	263	456	455	526	714

3.2 Experimental Results

In the experiments, only one frontal image per person with normal illumination (0°light angle) is applied as a training sample, which increases the difficulty of recognition. Recognition is performed with the nearest neighbor classifier measured with the Euclidean distance. For comparison, the proposed methods of [2] and [3] are implemented and named as “the DCT” and “the LN”. To compare the illumination model with noise and that without noise, we simply use the LRM in logarithm domain of the images without the step of denoising, and name the method as “LRM without denoising”. All the results are shown in Table 2.

From the table, it is clear that the proposed method achieves the best total performance compared with other methods. The results prove the validity of our assumption that the noise can be modeled based on high frequency components. For small illumination variations such as Subset 3, the DCT and LRM without denoising (they both only model the illumination as Eq. (1)) obtain better performances. For large illumination variations such as Subset 4 and 5, the LN and LRM (they consider the model as Eq. (2)) outperform other two methods. The comparison results demonstrate that the noise does not need to consider when only small illumination variation exists, and the noise needs to be modeled as an additive term when larger illumination variation exists.

Table 2. Performance comparisons of different methods

Method	Error rate (%)			
	Subset 3	Subset 4	Subset 5	Total
The DCT	10.5	10.8	12.6	8.1
The LN	12.3	6.3	8.4	6.2
LRM	11.2	7.6	7.6	6.0
LRM without denoising	10.5	8.2	10.9	7.0

3.3 Computational Complexity

Furthermore, we compare computational time of the LN and that of our proposed method because they achieve a better total recognition performance and apply similar local properties of human face. Suppose that the image size is *m***m* and the size of local area is *n***n*. The real computational time is calculated with the Matlab in a personal computer with a 2.66GHz CPU. The comparison is shown in Table 3. From the table, we can see that our method significantly reduces computational burden and speed up 64%.

Table 3. Comparison of computational complexity

	Computational complexity	Real computational time (per image)
The LN	$O(n^2m^2)$	2.51s
The LRM	$O((\log m + n)m^2)$	0.91s

4 Conclusions

In this paper, a low computation complexity illumination normalization approach for face recognition is proposed to address the problem of illumination variations. Different from most of existing methods, an additive term as noise is considered besides a multiplicative illumination term in illumination model. An appropriate number of high frequency DCT coefficients are zeroed to eliminate the effect caused by the noise. Based on local characteristic of human face, a simple but effective illumination normalization approach, local relation map, is proposed. We prove that

the LRM is robust against illumination variations. The experimental results on the Yale B and Extended Yale B prove the outperformance and lower computational burden of the proposed method compared to other existing methods. Further research on adaptive size selection of local area will be carried out in future.

Acknowledgment

The author would like to thank Yale University for the use of the Yale Face Database B and the Extended Yale Face Database B.

References

1. 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 19, 721–732 (1997)
2. Chen, W., Er, M.J., Wu, S.: Illumination compensation and normalization for robust face recognition using discrete cosine transform in logarithm domain. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 36, 458–466 (2006)
3. Xie, X., Lam, K.M.: An efficient illumination normalization method for face recognition. Pattern Recognition Letters 27, 609–617 (2006)
4. 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 Transactions on Pattern Analysis and Machine Intelligence 23, 643–660 (2001)
5. Lee, K.C., Ho, J., Kriegman, D.: Acquiring Linear Subspaces for Face Recognition under Variable Lighting. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 684–698 (2005)

Rapid Face Detection Algorithm of Color Images under Complex Background*

Chuan Wan[1], Yantao Tian[1,2,**], Hongwei Chen[1], and Xinzhu Wang[1]

[1] School of Communication Engineering, Jilin University, Changchun, 130025
[2] Key Laboratory of Bionic Engineering, Ministry of Education Jilin University
tianyt@jlu.edu.cn

Abstract. Human face detection plays an important role in applications such as video surveillance, human computer interface, face recognition, and face image database management. We propose a face detection algorithm for color images in the presence of varying lighting compensation technique and we segment the skin color of the face image with the improved skin color model which is in the space of YCrCb, and then find the human face in a small area using the template matching method. The application on the skin color information is add to the face detection algorithm, the purpose of which is to pre-judge whether the image exist a face, thus we can exclude non-skin regions to reduce the space to search and improve the efficiency of the face detection algorithm. This paper proposed various adaptive templates to overcome the shortcomings of poor adaptability, the scale of the template can be seized to adjust the size of the area, so the adaptability of template can be increased.

Keywords: Face Detection, Skin Color Clustering, Template Matching.

1 Introduction

Human activity is a major concern in a wide variety of applications such as video surveillance, human computer interface, face recognition [1], [2], [3], and face image database management [4], [5]. Detecting faces is a crucial step in these identification applications. Most face recognition algorithms assume that the face location is known. Similarly, face tracking algorithms often assume the initial face location is known. Therefore, some techniques developed for face recognition (e.g., feature-based approaches [6], and their combination [7], [8], [9], [10]) have also been used to detect faces, but they are computationally very demanding and cannot handle large variations in face images.

In color images, skin color is not sensitive to the attitude change, so we can extraction of color information features easily. Face detection method based on

* This paper is supported by the Key Project of Science and Technology Development Plan for Jilin Province (Grant No.20071152), project 20101027 supported by Graduate Innovation Fund of Jilin University and the Jilin University "985 project" Engineering Bionic Sci. & Tech. Innovation Platform.

** corresponding author

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 356–363, 2011.

template-matching is effective and generally applicable. Use face template matching with the original image can detect faces in images, and then we can locate the organ of face with the location of the template. Template matching method has a huge amount, so it does not apply to real-time systems. Generally, we need to reduce by auxiliary method detection range in a small area and then using the template method to achieve detection speed improved.

This paper sets completing earlier preparation for face recognition as its objection, using skin color clustering, and it has designed and realized human face detection algorithm under a complex background. The System Block Diagram as shown as in Fig.1.

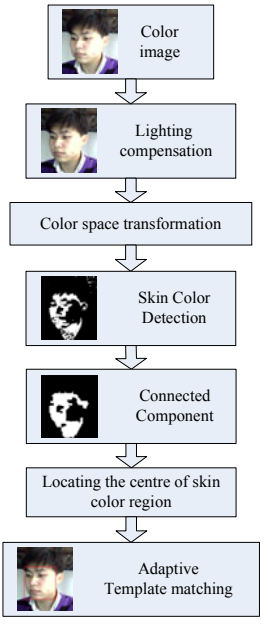


Fig. 1. System Block Diagram

2 Segmentation of Human Face Images Based on Skin Color Information

Skin color segmentation cannot work well in all conditions, nevertheless its low computational complexity makes the method much faster than many other ways of human face detection and using the message of skin color can remove most of the interference and only skin color and approximate skin color remain. According to the positive factors of human face detection based on skin color, it can be used as a preprocessing before some methods whose algorithm may be accurate in the aspect of detection but complexity to narrow the detection area. Through this preprocessing, the overall performance of algorithm will be promoted and it is crucial for the following steps of human face detection.

In order to meet the necessary of our face detection system, we propose a method based on nonlinear skin color clustering and local edge method. The reason why the method proposed here can be described as: (1)As one of the most important information of human faces, skin color hardly depends on the details, rotation angles and different expressions of human faces, so skin color segmentation is stable to some degree and it can tell the differences between the skin color and complex background.. (2)With the simple algorithm, skin color clustering method has less computational cost, so it fit for the demand of real time detection. (3)Because of local discreteness of skin color clustering in color space, using nonlinear transformation can improve the effect of clustering.

2.1 Skin Color Model

People make some color models to unify the expression of different color. Nowadays CIE, RGB, HIS, YUV and YCrCb color models are commonly used and different color methods have different application areas. In skin color detection we normally choose YCrCb color space. It is similar to the HIS color space which can separate luminance information from color imagines. Because of the luminance component Y can be computed through linear transformation of RGB color space, Y is related to chrominance and the skin color clustering area has the trendency of nonlinear.

$$\begin{bmatrix} Y \\ Cb \\ Cr \\ 1 \end{bmatrix} = \begin{bmatrix} 0.2990 & 0.5870 & 0.1140 & 0 \\ -0.1687 & -0.3313 & 0.5000 & 128 \\ 0.5000 & -0.4187 & -0.0813 & 128 \\ 0 & 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} \mathrm{R} \\ \mathrm{G} \\ \mathrm{B} \\ 1 \end{bmatrix} \tag{1}$$

First, during the skin color segmentation, we ignore the effective of Y, thus we transform the problem from 3-D space to 2-D space. In this 2-D space of $C_b - C_r$, skin color is concentrated and normally can be described as Gauss distribution, so that distribution center can be obtained through the training. Then according to the distance of pixels from the center, we can get a similarity distribution compare to the initial image. At last the final area of skin color can be detected after transforming the similarity distribution into binary with specific rules. Considered the effective of light to the luminance Y, we only use the parameters C_b, C_r to make Gaussian model, therefore for pixels (x, y) we have

$$\begin{cases} \hat{C}_b \sim N(\mu_b, \sigma_b^2) \\ \hat{C}_r \sim N(\mu_r, \sigma_r^2) \end{cases} \tag{2}$$

The parameters $\mu_b, \mu_r, \sigma_b, \sigma_r$ mentioned in this formula are the mean and standard deviation of skin chrominance and these parameters can be calculated through our experiment. On the basis of experimental and network human face pictures segmented by people in normal illumination, the initial value of these parameters are

$\mu_b = 115, \mu_r = 148, \sigma_b = 10, \sigma_r = 10$. Whereas the illumination, actually, will be changing with time and the severe illumination change can not only affect the luminance Y, but also the chrominance Cr, Cb. If we keep the Gaussian model remain without any change, the system may fail to find the area of human faces. So in this paper we propose a new model which can update the parameters with the changing of illumination.

$$\begin{cases} \mu_b = 115, \mu_r = 148 & if \quad Y \in [TL_y, TH_y] \\ \mu_b = 115 + C_1(C_b - 115), \mu_r = 148 + C_2(C_r - 148) & otherwise \end{cases} \tag{3}$$

In this formula TL_y, TH_y are the upper threshold and lower threshold. ,

For a video image $F_k'(x, y)$, if C_b any C_r , the value of chrominance of pixels (x, y) , content the Gaussian distribution and they also meet all of the limiting conditions which are $|C_b - \mu_b| \triangleright 2.5\sigma_b$ and $|C_r - \mu_r| \triangleright 2.5\sigma_r$, we deem that the pixels belong to the area of human face image. The other pixels which cannot meet these conditions are defined as dark. After these we can exact the human face area from the whole image.

Inevitable the human face images exacted through the processing have noise interferences, such as the roughen edges, small holes and color spots. To solve the problem we use a method called corrosion and it can well remove these interferences.

2.2 Skin Color Segmentation

Due to the reason that the hands and arms of humans are also skin color and the existence of some approximate skin color areas, we cannot expect get the exact areas of human faces only using skin color segmentation. However human face areas have their unique character which is connectivity, thus we can expel the single pixel point and make those points of density connected. By this way we can not only reduce the noise interference but promote the human face areas. The concrete method is divided into three steps:

(1) Human face area always occupies a certain size in an image. First, we make a count on the size of human faces. Then we set S_r as the threshold of the size. When the size of one area is less than S_r , we define this area as black.

(2) No matter the faces without rotation, extremely rotated faces, upward faces and downward faces, the width ratios of their circum-rectangle are all in a certain range. According to the prior knowledge, we get the range is [1/2, 2/3] and if one area cannot meet the condition, we define it as black.

3 Adaptive Template Matching

Template matching method has the advantages of easy to implement, but it has poor adaptability. According to the size of the region to be seized of the scale was adjusted to increase the adaptability of the template, this paper proposed the method of adaptive template matching.

3.1 Training for Face-Template

In consideration of the computing speed, we use one template only, and for the template can express face better, we need to make face sample cutting, scale changes, and gray distribution standardization for each sample, and then average the gray values of all samples and compress to the needed size as the primitive human face template. The template is constructed by the way of make averaging of many samples, the procedure of operation is:

(1) Choose the human face area of the image as the face sample image; position the eyes manually to ensure that the position of eyes is fixed in the face image;
(2) Standardize the scale and gray of every face sample;
(3) Extract the edge of samples using Sobel operator;
(4) Average the gray values of the images that have been processed by the step 3. the final human face template by training is as shown as Fig.2.

Fig. 2. Training Face Template

Here the major work is the gray standardization of images. For eliminating the effection of light and other conditions on the image acquisition, we need to standardize the gray so as to make the expectation and variance close. Using the vector $x=[x_0, x_1, \cdots x_{n-1}]$ to represent an image, and then its expectation of gray can be expressed as $\overline{\mu}$; the variance can be expressed as $\overline{\sigma}$. For each input sample, to transform the expectation and variance of it into the expected ones, the transformation for each pixel should be done as follows:

$$\hat{x}_i = \frac{\sigma_0}{\overline{\sigma}}(x_i - \overline{\mu}) + \mu_0 \quad 0 \le i < n \tag{4}$$

3.2 Template Scaling

Because that the size of the detected region is not always the same as the template, we have to scale the template in accordance with the detected region when the size of the template and the one of the detected region are different. The concrete process is that, first, to define the location and size of the image according to the center and outside rectangle of the detected region determined by the formula; and then calculate the area of the detected region; finally, determine the scaling ratio of the template by calculating the ratio of the area of the detected region and the one of the template. That can scale the template by the way of changing the size of the image. (X_c, Y_c) is the center of mass of skin region.

$$X_C = \sum_{i=0}^{n} X_i / n\,, \quad Y_C = \sum_{i=0}^{n} Y_i / n \tag{5}$$

Fig. 3. Template Scaling

3.3 Detection

Suppose the gray matrix of face template is $T[M][N]$, gray mean is μ_r, standard deviation is σ_r, the gray matrix of input image area is $R[M][N]$ and then the correlation coefficient between them and pixel gray value corresponds to the average deviation values respectively are:

$$r(T,R)=\frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(T[i][j]-\mu_r)(R[i][j]-\mu_R)}{M\bullet N\bullet\sigma_r\bullet\sigma_R} \tag{6}$$

$$d(T,R)=\sqrt{\frac{\sum_{i=0}^{M-1}\sum_{i=0}^{N-1}(T[i][j]-R[i][j])^2}{M\bullet N}}$$

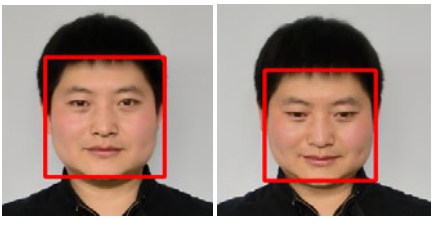
Measure the matching degree depends on the formula (7)

$$D(T,R)=r(T,R)+\frac{\alpha}{1+d(T,R)} \tag{7}$$

Because of skin color image segmentation have multiple color pieces, we should match every pieces. To scan each piece of image, if the face is greater than the relevant threshold in the scan window, we can mark the location of human face.

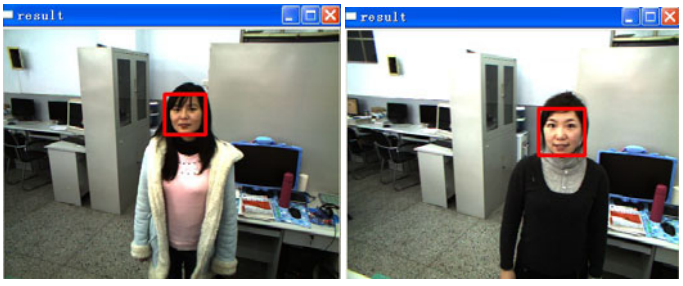
4 Experimental Results and Conclusion

We use the PC with Celeron (R) CPU 2.8G, 2G memory. The system detects a single image use 70ms. We can see that the system can meet the requirements of real-time. In the absence of any tracking algorithm into the case, face detection, the average speed of 6 FPS. In the Fig.6, we list a part of images contain different background and facial stance. The simulation results are given in TABLE I.



(a) Image in Gallery

Fig. 4. A part of Simulation Results are shown here



(b) Real-time Tracking

Fig. 4. (*Continued*)

Table 1. Simulation Results

Image types	Correct number	Wrong number
Rectify face	10	0
Upward face	9	1
Downward face	8	2
Left side face	10	0
Right side face	10	0
With expresstion	10	0
Uneven illumination	7	3

5 Conclusion

The way of face detection mentioned in this paper needs less computational cost, so it can be used to handle images. Additionally, the algorithm based on the combination of skin color segmentation and template matching could not only inherit their speed advantage but also overcome the adverse effect from complex background. From the whole paper we can conclude that the method mentioned in this paper can promote the efficiency and the accuracy in face detection and tracking in images with complex background.This paper proposed various adaptive templates to overcome the shortcomings of poor adaptability, the scale of the template can be seized to adjust the size of the area, so the adaptability of template can be increased.

References

1. Yang, M.-H., Kriegman, D.J., Ahuja, N.: Detecting faces in images: A survey. IEEE Trans. PAMI 24(1), 34–58 (2002)
2. Hsu, R.L., Mottaleb, M.A., Jain, A.K.: Face detection in color images. IEEE Trans. Pattern Analysis and Machine Intelligence 24(5), 696–706 (2002)
3. Hsu, R.-L., Abdel-Mottaleb, M., Jain, A.K.: Face Detection in Color Images. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(5), 696–706 (2002)

4. Hsieh, I.-S., Fan, K.-C., Lin, C.: A Statistic Approach to the Detection of Human Faces in Color Nature Scene. Pattern Recognition 35(7), 1583–1596 (2002)
5. Brand, J.D., Mason, J.S.D.: A Skin Probability Map and Its Use in Face Detection. In: Proceedings of International Conference on Images Processing, vol. 1(7-10), pp. 1034–1037 (2001)
6. Zhang, B.-B., Zhang, C.-S.: Lower bounds estimation to KL transform in face representation and recognition. In: Proc of 2002 International Conference on Machine Learning and Cybernetics, pp. 1314–1318 (2002)
7. Hyungkeun, J., Kyunghee, L., Sungbum, P.: Eye and face detection using SVM. In: Proc of the 2004 Intelligent Sensors, Sensor Networks and Information Processing Conference, pp. 577–580 (2004)
8. Liu, S.-s., Tian, Y.-t., Li, D.: New Research Advances of Facial Expression Recognition. In: Proceedings of the Eighth International Conference on Machine Learning and Cybernetics, Baoding, July 12-15, pp.1150–1155 (2009)
9. Viola, P., Jones, M.: Robust real time object detection. In: 8th IEEE International Conference on Computer Vision (ICCV), Vancouver, British Columbia (2001)
10. Wan, C., Tian, Y.-t., Chen, H., Liu, S.-s.: Based on Local Feature Region Fusion of Facial Expression Recognition. In: The 2nd International Conference on Advanced Computer Control, Shenyang, March 27-29, vol. 1, pp. 202–206 (2010)
11. Nallaperumal, K., Subban, R.: 2006 IFIP International Conference on Human Face Detection in Color Images Using Skin Color and Template Matching Models for Multimedia on the Web Wireless and Optical Communications Networks, April 11-13, pages 5 (2006)
12. Hjelmas, E., Low, B.K.: Face Detection: A Survey. Computer Vision and Image Understanding 83, 236–274 (2001)

An Improvement Method for Daugman's Iris Localization Algorithm

Zhi-yong Peng, Hong-zhou Li, and Jian-ming Liu

Guilin University of Electronic Technology, Guilin, 541004, Guangxi, P.R. China
pzy@guet.edu.cn

Abstract. An improvement method is present in this paper for Daugman's iris localization algorithm. It may make iris localization more rapid and more precise. It may also decrease the computational complexity of the localization algorithm by reducing the search area for the iris boundary center and the radius, In addition, a new method excluding the upper and lower eyelids is proposed. Experiments indicate that the present method have better performance.

Keywords: iris, localization, inner boundary, outer boundary.

1 Introduction

Reliable automatic recognition of persons has long been an attractive goal. The central issue in pattern recognition is the relationship between within-class variability and between-class variability. Objects can be reliably classified only if intra-class variability is less than inter-class variability. While seeking to maximize the between-person variability, biometric templates must also have minimal within-person variability across time and changing conditions of capture [1]. For example, in face recognition, difficulties arise from the fact that the face is a changeable social organ displaying a variety of expressions [2][3]. Against this intra-class (same face) variability, inter-class variability is limited because different faces possess the same basic set of features, in the same canonical geometry. It has been shown that current algorithms can have error rates of 43% to 50%[4]-[7].

For all of these reasons, iris patterns become interesting as an alternative approach to reliable visual recognition of persons, and especially when there is a need to search very large databases without incurring any false matches despite a huge number of possibilities. The iris begins to form in the third month of gestation and the structures creating its pattern are largely complete by the eighth month [8]. Its complex pattern can contain many distinctive features such as arching ligaments, furrows, ridges, crypts, rings, corona, freckles, and a zigzag collarette [9][10]. The number of degrees-of-freedom in Iris Codes is 249[11]. In NIR wavelengths, even darkly pigmented irises reveal rich and complex features [12][13]. Monochrome CCD cameras (768*576) and NIR have been used at distances of 20 centimeters collection the image of iris may be seen in Fig. 1.

In the iris recognition, it is necessary to localize precisely the inner and outer boundaries of the iris, and to detect and exclude eyelids if they intrude. Influences the veracity of iris recognition directly. Daugman's algorithm for recognizing iris patterns has been the executable software used in most iris recognition systems so far

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 364–372, 2011.

deployed commercialllly or in tests, including those by British Telecom, Sandia Labs, U.K, and so on. The Daugman's algorithm operation is very complex. Moreover, the Asian iris color is darker and texture is more blur. At times, using Daugman's algorithm directly maybe happen mistake to localize iris for Asian. At the same time, eyelids are usually covered with eyelash. It is very difficult to use Daugman's algorithm to localize eyelid. So it is needed to improve the Daugman's algorithm. This thesis introduces an improvement method for Daugman's iris localization algorithm and proves it is feasible.

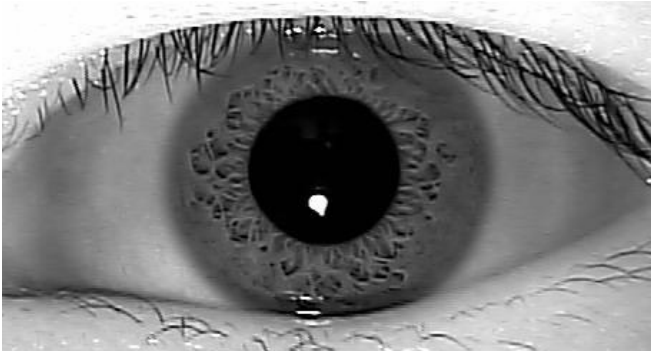


Fig. 1. Example of an iris pattern, imaged monochromatically at a distance of about 20cm

2 Daugman's Algorithm for Iris Localization

The iris localization can be completed after having ascertained the center coordinates (x_0, y_0) and radius(r), which are the three parameters defining the pupillary circle and limbus circle. Daugman's algorithm localizes iris by searching max difference of gray between the arcs. A very effective integrodifferential operator for determining these parameters is: [14]-[17]

$$\max_{(r,x_0,y_0)} \left| G_\sigma(r) * \frac{\partial}{\partial r} \oint_{(r,x_0,y_0)} \frac{I(x,y)}{2\pi r} ds \right| \tag{1}$$

Where $I(x,y)$ is the gray of single-pixel in the image. The operator searches over the image domain (x,y) for the maximum in the blurred partial derivative with respect to increasing radius r, of the normalized contour integral of $I(x,y)$ along a circular arc of radius and center coordinates (x_0, y_0) . The symbol * denotes convolution and $G_\sigma(r)$ is a smoothing function such as a Gaussian of scale σ . The complete operator behaves as a circular edge detector, blurred at a scale set by σ , searching iteratively for the maximal contour integral derivative at successively

finer scales of analysis through the three parameter space of center coordinates and radius (x_0, y_0, r) defining a path of contour integration.

The operator in (1) serves to find both the pupillary boundary and the outer (limbus) boundary of the iris, a similar approach to detecting curvilinear edges is used to localize both the upper and lower eyelid boundaries. The path of contour integration in (1) is changed from circular to arcuate, with spline parameters fitted by statistical estimation methods to model each eyelid boundary.

3 The Improved Method for Daugman's Algorithm

The improvement is based on Daugman's iris localization algorithm, by use of a coarse-to-fine strategy terminating in single-pixel precision estimates of the center coordinates and radius of both the limbus and the pupil

Firstly, localize the inner (pupillary) boundary. Daugman's iris localization algorithm needs to search in the large scope, because we don't know the center coordinates and radius of the pupil. The improvement for this is to localize it by two steps. The first step is coarse localization of the pupil according to area formula of circle. The second step is fine localization by Daugman's algorithm to find the pupil's precise center and radius in small area.

Secondly, localize the outer (limbus) boundary. Because the upper and lower eyelid intrudes, we find the outer boundary from right and left canthus. It also follows two steps from fast coarse localization to fine localization.

Thirdly, localizing both the upper and lower eyelid boundaries. Daugman's algorithm doesn't work very well to localize them because the eyelash usually covers with the eyelid. So many noises are used as iris information. The method used in this thesis excludes directly the area of iris maybe covered by the eyelid.

3.1 Finding the Inner (Pupillary) Boundary of the Iris

As the pupil is always rounding we can localize the pupil if we may ascertain its center and radius. At first we localize the inner boundary coarsely by finding those points in the pupil. Then, we further localize it precisely.

Coarse Localization of the Pupil

The method about ascertaining center and radius is to search all pixels that belong to the pupil in the image and store the X-coordinates and Y-coordinates of each pixel. According to area formula of circle $S = \pi * r^2$ we can get the pupil's radius, where S denotes the number of pixels in the pupil. Get the pupil's center coordinates by computing the mean of all these pixels' X-coordinates and Y-coordinates.

The method about searching those pixels belonging to the pupil is based on the fact that the pupil is blacker than other area of the image. We can easily find that the pupil's gray is smaller than the other area and that the color is almost the same for each pixel in the pupil. The gray change between the pixels is very small. So we can judge whether a point is in the pupil accord to the gray. Set the threshold of gray to be L, the threshold of gray change between pixels $\triangle$L.

$$L = \frac{DC}{5 - 2 * \frac{DC - 128}{128}} \quad (2)$$

Where DC denotes the average gray of the whole image.

The difference of gray $f(I)$ is:

$$\begin{aligned} f(I) = &|I(x,y)-I(x-1,y-1)|+|I(x,y)-I(x,y-1)|+|I(x,y)-I(x+1,y-1)|+|I(x,y)-I(x-1,y)| \\ &+|I(x,y)-I(x+1,y)|+|I(x,y)-I(x-1,y+1)|+|I(x,y)-I(x,y+1)|+|I(x,y)-I(x+1,y+1)| \end{aligned} \quad (3)$$

The average of $f(I)$ is $\overline{f(I)}$

The threshold of gray change between the pixels $\triangle$L is:

$$\triangle L = \overline{f(I)} / 2 \quad (4)$$

If the pixel meets with the following two inequations with simultaneity: I(x, y)<L, f(I)<$\triangle$L, the pixel is concluded to be in the pupil.

Accurate Localization of the Pupil

From Fig.1 we can also see, in fact, that there is facular in the pupil. Facular's gray is much higher than normal pixels' in the pupil, having big gray difference compared to normal pixels in the pupil. So it maybe reckoned as the iris's inner boundary by mistake. Thus, before fine localization we must remove the facular. To do so, we can find the facular in the coarse scope of the pupil according to gray value and set those facular's pixels to be zero gray.

After removing the facular, we then localize the pupil accurately according to the following operator:

$$\max_{(r,x_0,y_0)} \left| \frac{\partial}{\partial r} \oint_{(r,x_0,y_0)} \frac{I(x,y)}{2\pi r} ds \right| \quad (5)$$

Suppose the coarsely localized center of the pupil is $O_1(x_1,y_1)$ and the radius of the pupil is r_1. We generally may assure the actual center of pupil $O_0(x_0,y_0)$ is in the rectangle area $(x_1\text{-}N)<x<(x_1+N)$, $(y_1\text{-}N)<y<(y_1+N)$ and the true radius of pupil r_0 in the area $r_m < r_0 < K \times r_1$, where N and r_m are carefully selected by the resolution of the image and the experiment. We find through many experiments that if the size of image is 768*576 the best value is around $N=20$, $K=1.4$, $r_m=20$.

3.2 Finding the Outer (Limbus) Boundary of the Iris

From Fig.1 we can also find that a part of the iris is usually covered by upper and lower eyelid. Thus we can't make the integral calculus by the whole circumference to

find the iris's outer boundary. However, we can localize it by the gray difference on the arcs of the right and left canthus. So we choose the arc scope of right side canthus to be $-45^0<\theta<+45^0$ and the arc scope of the left side canthus $135^0<\theta<225^0$.

The iris's outer boundary and inner boundary is usually not concentricity, and the pupil center is nasal and inferior to the iris center. So we need to relocate the center of the outer limbus. Because the distance is very small between the center of the pupil and the center of the outer limbus, we can search the center of outer limbus around the pupil's center, which reduces the search scope and hence decreases the complexity of calculation. We also use the coarse-to-fine strategy.

Fast Coarse Localization of the Outer Boundary

Outer boundary localization is based on the above fine pupil localization. We localize outer boundary by magnify the change step of r. Denote the pupil's center to be $O_0(x_0,y_0)$ and the radius to be r_0. Firstly, suppose $O_0(x_0,y_0)$ is the center of outer boundary and then after localizing the arc of the left and right side canthus, the radius is in $K*r_0<r<M$. We get the radius of right arc r_2 and the radius of left arc r_3 according to operator (6).

$$\max_{(r,x_0,y_0)} \left| \oint_{(r,x_0,y_0)} \frac{I(x,y)}{2\pi r} ds - \oint_{(r_1,x_0,y_0)} \frac{I(x,y)}{2\pi r_1} ds \right| r_1 = r - \Delta r \quad . (6)$$

Set the average of r_2 and r_3 as the coarse radius of iris's outer boundary R_0. It means that $R_0=(r_2+r_3)/2$ and the coarse coordinates of center of limbus are: $O(X_0,Y_0)$, where $X_0= x_0+(r_3-r_2)/2$, $Y_0=y_0$. In experiment we find that the best value is M =200, K=1.4, Δr =9, if the size of image is 768*576.

Accurate Localization of the Limbus

After coarsely localizing the iris's outside edge, find the outer boundary of iris accurately according to operator (1). The search scope for the center is $(X_0-A)<x<(X_0+B)$, $(Y_0-C)<y<(Y_0+D)$, the radius scope is: $(R_0-\Delta r)<r<(R_0+\Delta r)$, The search is processed on both left and right canthus synchronously. The distance is very small between the center of the pupil and the center of the outer limbus. We let A=2, B=4, C= 3, D=5 in our experiment. Finally, we can get the fine center and radius of outer boundary of iris.

3.3 Finding the Upper and Lower Eyelid Boundaries

From Fig.1 we can also find that the pixels between iris and eyelids have gray difference, If we use the Daugman's algorithm to exclude the upper and lower eyelid by searching max difference of gray between the arcs, the eyelash will are withheld in the iris. Yet, iris complex pattern can contain many distinctive features. Moreover, the most distinctive features convergence is close to the inner boundary of the iris [18], and in theory [19] only 65% of the iris is quite enough for our recognition. so we can

remove the area of the iris which might be covered with eyelid. It doesn't influence the result of recognition. In this thesis, we extract the iris as follows: Suppose the iris's radius of inner boundary to be R_1 and the iris's radius of outer boundary to be R_2.

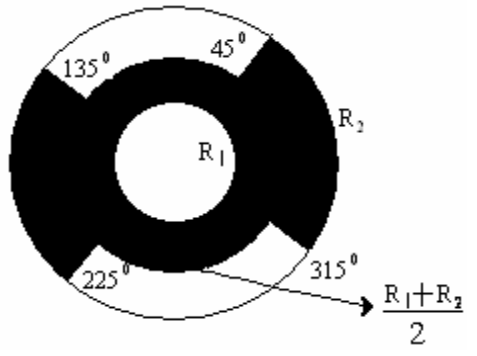


Fig. 2. The extracted area of the iris. R_1 is pupil's radius and R_2 is iris's radius of outer boundary

In the left canthus, choose the iris in the scope $225^0<\theta<315^0$ and $R_1 < r < R_2$. In the right canthus, Choose the iris in the scope $45^0<\theta<135^0$ and $R_1 < r < R_2$. For the area maybe covered with eyelid, Choose the iris in the scope $225^0<\theta<315^0$ and $45^0<\theta<135^0$ and $R_1 < r < (R_1 + R_2)/2$. This extracted area of the iris can be seen in the Fig.2. The black segment is the area of extracted iris.

4 Experimental Results

On a 1.9GHZ CPU and 256M EMS memory computer, our improved Daugman's algorithm needs only 112 ms to localize the iris while the original Daugman's algorithm needs 195ms. Fig. 3 illustrated the result of the localization by Daugman's localization algorithm. We can see that many eyelashes are withheld in the iris. Fig. 4 illustrated the result of the localization by our improved algorithm. We can see that the eyelash noises are excluded effectively. So the improvement method of Daugman's algorithm can accomplish the iris recognition more rapidly and more precisely.

After finding an iris in image, we use 2-D wavelet demodulation to achieve iris feature encoding [20][21]. Then using the simple Boolean Exclusive-OR test statistical independence [22], Hamming Distance (HD) as the measured of the dissimilarity between any two irises [23]. The end set the judge criterion of HD to get the result of recognition.

We have token pictures of 500 eyes to do iris recognition experiments; Distribution of HDs from all 124750 possible comparisons between different pairs of irises is shown in Fig.5.

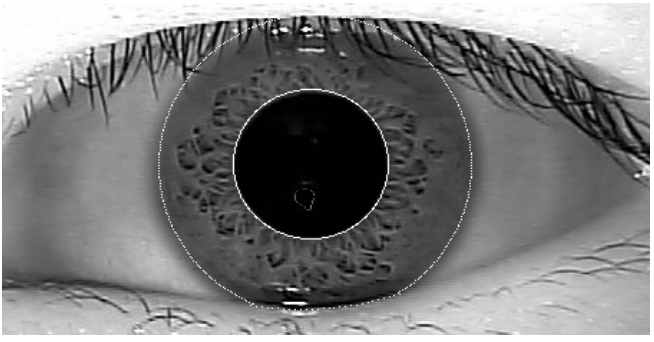


Fig. 3. The results of the iris localization by Daugman's algorithm

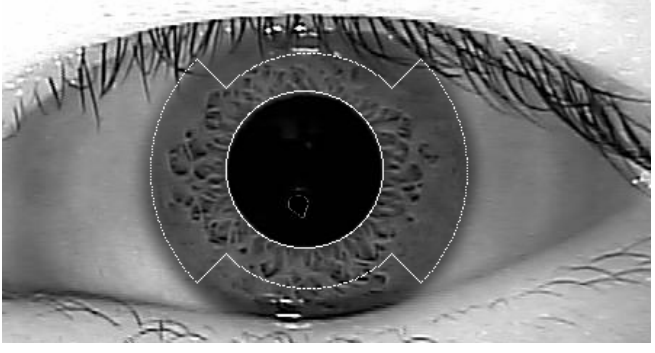


Fig. 4. The results of the iris localization by the improved algorithm

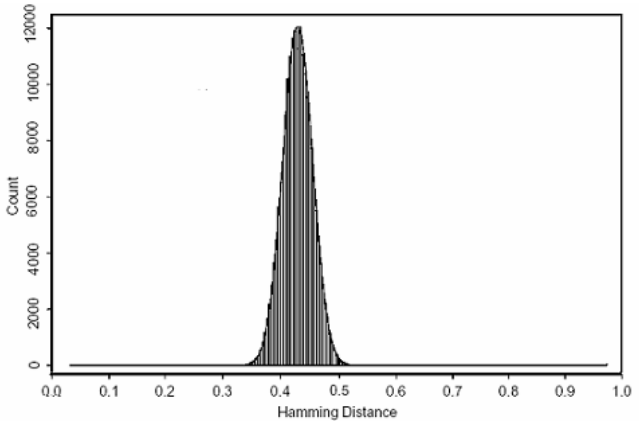


Fig. 5. Distribution of HDs from all 124750 possible comparisons between different pairs of irises. The histogram forms a perfect binomial distribution with p=0.5 and N=249 degrees-of-freedom, mean=0.451.

We set HD criterion to be 0.32 and the false accept rate is zero. Indicating that the improvement method of Daugman's algorithm can be applied to iris recognition very reliably.

5 Conclusions

In this paper we improved Daugman's iris localization algorithm in two aspects. One is to localize the iris firstly by fast coarse localization and then by fine localization. The other is to exclude the area of the iris that might be covered by the eyelid. These improvements may make iris localization faster and more precise than the original Daugman's algorithm. Our experiments have shown that the improved algorithm could be reliably used for iris recognition

References

1. Zhang, D.: Automated Biometrics: Technologies and Systems. Kluwer, Dordrecht (2000)
2. Adini, Y., Moses, Y., Ullman, S.: Face recognition: the problem of compensating for changes in illumination direction. IEEE Trans. Pattern Anal. Machine Intell. 19, 721–732 (1997)
3. Blanz, V., Romdhani, S., Vetter, T.: Face Identification across. Different Poses and Illuminations with a 3D Morphable Model. In: Proc. Fifth Int'l. Conf. Automatic Face and Gesture Recognition, pp. 202–207 (2002)
4. Phillips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J.: The FERET evaluation methodology for face-recognition algorithms. IEEE Trans. Pattern Anal. Mach. Intell. 22(10), 1090–1104 (2000)
5. Pentland, A., Choudhury, T.: Face recognition for smart environments. Computer 33(2), 50–55 (2000)
6. 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. Pattern Analysis and Machine Intelligence 23(6), 643–660 (2001)
7. Edwards, G.J., Cootes, T.F., Taylor, C.J.: Face Recognition Using Active Appearance Models. In: Burkhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1407, p. 581. Springer, Heidelberg (1998)
8. Kronfeld, P.: Gross anatomy and embryology of the eye. In: Davson, H. (ed.) The Eye. Academic, London (1962)
9. Adler, F.: Physiology of the Eye: Clinical Application, 4th edn. The C.V. Mosby Company, London (1965)
10. Johnson, R.G.: Can iris patterns be used to identify people, Los Alamos National Laboratory, CA, Chemical and Laser Sciences Division, Rep. LA-12331-PR (1991)
11. Daugman, J.: The importance of being random: Statistical principles of iris recognition. Pattern Recognition 36(2), 279–291 (2003)
12. Wildes, R.P.: Iris recognition: an emerging biometric technology. Proceeding of The IEEE 85(9), 1348–1363 (1997)
13. Wildes, R.P., Asmuth, J.C., Green, G.L., Hsu, S.C., Kolczynski, R.J., Matey, J.R., McBride, S.E.: A machine vision system for iris recognition. Mach. Vision Applicat. 9, 1–8 (1996)

14. Daugman, J.: How iris recognition works. IEEE Trans. Circuits and Syst. for Video Tech. 14(1), 21–30 (2004)
15. Daugman, J.: High confidence visual recognition of persons by a test of statistical independence. IEEE Trans. Pattern Anal. Machine Intell. 15, 1148–1161 (1993)
16. Daugman, J.: Biometric Personal Identification System Based on Iris Analysis. U.S. Patent, 291–560 (1994)
17. Daugman, J.: Statistical richness of visual phase information: Update on recognizing persons by their iris patterns. Int. J. Computer Vision 45(1), 25–38 (2001)
18. Seal, C., Gifford, M., McCartney, D.: Iris recognition for user validation. British Telecommunications Engineering Journal 16(7), 113–117 (1997)
19. Daugman, J.: Demodulation by complex-valued wavelets for stochastic pattern recognition. Int. J. Wavelets, Multiresolution Inf. Processing, 1–17 (2003)
20. Daugman, J.: Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. J. Opt. Soc. Amer. 2(7), 1160–1169 (1985)
21. Daugman, J.: Complete discrete 2D gabor transforms by neural networks for image analysis and compression. IEEE Trans. Acoustic, Speech and Signal Processing 36, 1169–1179 (1988)
22. Gerald, O.: Williams, Iris Recognition Technology. IEEE Aerospace and Electronic Systems Magazine 12(4), 23–29 (1997)
23. Wildes, R.P., Asmuth, J.C., Hsu, S.C., Kolczynski, R.J., Matey, J.R., McBride, S.E.: Automated, noninvasive iris recognition system and method. U.S. Patent 5, 572–596 (1996)

Fire Detection with Video Using Fuzzy c-Means and Back-Propagation Neural Network

Tung Xuan Truong and Jong-Myon Kim*

School of Computer Engineering and Information Technology,
University of Ulsan, Korea
{tl20003,jongmyon.kim}@gmail.com

Abstract. In this paper, we propose an effective method that detects fire automatically. The proposed algorithm is composed of four stages. In the first stage, an approximate median method is used to detect moving regions. In the second stage, a fuzzy c-means (FCM) algorithm based on the color of fire is used to select candidate fire regions from these moving regions. In the third stage, a discrete wavelet transform (DWT) is used to derive the approximated and detailed wavelet coefficients of sub-image. In the fourth stage, using these wavelet coefficients, a back-propagation neural network (BPNN) is utilized to distinguish between fire and non-fire. Experimental results indicate that the proposed method outperforms other fire detection algorithms, providing high reliability and low false alarm rate.

Keywords: fire detection, color segmentation, fuzzy c-means algorithm, back-propagation neural network.

1 Introduction

Fire detection becomes more and more appealing because of its important application in surveillance systems. Thus, it is very attractive for the personal security and commercial application. Several conventional methods were proposed to detect fire. However, most of these methods require a close proximity to the source of the fire and are based on particle sensors. Therefore, they cannot detect fire in open or large spaces and cannot provide additional information regarding the process of burning. To overcome these weaknesses, video fire detection is a suitable candidate.

A lot of fire detection algorithms in video have been proposed. Most of these algorithms are based on the color pixel recognition, on motion detection, or on both of them. In [1], a dynamic analysis of flames with the growth of pixels using an RGB/HIS color model was used to check for the existence of a fire. However, the decision rule of this method is not good at distinguishing real fire regions from moving regions or noise since they measured the flame difference between only two consecutive frames. In [2] and [5], the boundary of flames was represented in the wavelet domain and the high frequency nature of the boundaries of fire regions was also used as a clue to model the flame flicker spatially, yielding good results.

* Corresponding author.

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 373–380, 2011.

However, this algorithm was performed with a stationary camera, and it required high computational complexity despite of working in real-time. In [3], a new method of flame detection was proposed that use the pixel color properties of the flame. In [4], fire detection based on vision sensor and support vector machines was proposed. In this paper, a non-linear classification method using support vector machines and luminescence maps was proposed, showing robust results in several scenarios compared to features used earlier for flame detection. In [6], a probabilistic approach for vision-based fire detection was proposed. In this paper, a probabilistic model for color-based fire detection was utilized to extract the candidate fire regions. In addition, four parameters were extracted from the features of candidate fire regions, such as area size, surface coarseness, boundary roughness, and skewness. Moreover, Bayes classifier was used to distinguish between fire and non-fire. Some of the above algorithms were applied to the real system, achieving considerable successes. However, these algorithms have limited application and lacked enough robustness. In order to enhance the performance of fire detection, we proposed an effective four stage fire detection method that investigates characteristics of fire using an approximate median, fuzzy c-means, discrete wavelet transform, and back-propagation neural network algorithms.

The rest of this paper is organized as follows. Section 2 introduces the feature of fire. Section 3 represents the proposed four stage fire detection method. Section 4 discusses experimental results of the proposed method and compares the performance of the proposed method with other fire detection algorithms, and Section 5 concludes this paper.

2 Features of Fire

In practice, most fuels will burn under appropriate conditions, reacting oxygen from the air, generating combustion products, emitting light, and releasing heat. When fire appears, the color of fire usually range from red to yellow, and it may become white when the temperature is high. The size of area in the fire regions will be changed contingently from frame to frame. The surface and the boundary of the fire regions are usually rough and coarse.

3 The Proposed Fire Detection Method

The proposed fire detection method in video consists of four stages: (1) moving region detection using an approximate median method, (2) color segmentation of fire using the fuzzy c-means (FCM) clustering, (3) parameters extraction from the candidate fire regions using the discrete wavelet transform (DWT), and (4) fire identification using the back-propagation neural network (BPNN). In the following sections, the proposed fire detection method is presented in detail.

3.1 Moving Region Detection

The detection of moving regions is a fundamental key in video fire detection, which is the first stage of the proposed method. In this stage, an approximate median method is used to balance between the accuracy and the computational time [7].

In the experiment, we utilized only the gray image. Let $I_n(i, j)$ be the intensity value of the pixel at the location (i, j) in the n^{th} video frame. The estimated background intensity value $B_{n+1}(i, j)$ at the same position is calculated as follows:

$$B_{n+1}(i,j) = \begin{cases} B_n(i,j)+1 & \text{if } I_n(i,j) > B_n(i,j) \\ B_n(i,j)-1 & \text{if } I_n(i,j) < B_n(i,j) \end{cases} \tag{1}$$

where $B_n(i, j)$ is the previous estimate of the background intensity value at the same pixel position. From (1), we observe that the background is updated after every frame. In this way, the background eventually converges to an estimate where half of the input pixels are greater than the background, and half are less than the background. Initially, $B_1(i, j)$ is set to the first image frame $I_1(i, j)$. A pixel positioned at (i, j) is assumed to be moving if:

$$\left| I_n(i,j) - B_n(i,j) \right| > T \tag{2}$$

where T is a threshold that is picked by guesswork and experience.

3.2 Color Segmentation of Fire Using the FCM Algorithm

In practice, there are several objects in video which move along with fire such as people, vehicles, birds, cloud, and smoke. However, most the colors of these objects differ from the color of fire. Because of this fact, we take into account for the color segmentation of fire in this study. The basic idea in this stage is composed of two steps: (1) the pixels in the moving regions are distributed into groups and then (2) the groups having the similar color of fire are selected. To accomplish this, the well-known fuzzy c-means (FCM) is employed [8, 9]. In addition, since the RGB color space is not device independent and is deficient in enough robustness, the CIE LAB color space is used which is completely device independent and is the most effective for the physical vision with higher accuracy. The CIE LAB color components can be obtained by converting from RGB to CIE LAB [10]. It consists of three components L, A and B where L indicates the luminosity of pixels, A and B indicate the color of pixels. The chrominance components A and B are considered as an input for the FCM algorithm. The output of the FCM algorithm is the clusters of pixels in the moving regions. The steps followed by the FCM algorithm are listed as follows:

1. Compute the number of groups c and initialize centroid $V^{(0)} = \{v_1^{(0)}, v_2^{(0)}, \ldots, v_c^{(0)}\}$.
2. Compute the membership values u_{ij} for each data element using the following equation:

$$u_{ij} = \left[\sum_{k=1}^{c} \left(\frac{d^2(x_j, v_i)}{d^2(x_j, v_k)} \right)^{\frac{1}{m-1}} \right]^{-1} \tag{3}$$

3. Update the centroid value v_i as follows:

$$v_i = \frac{\sum_{j=1}^{n} u_{ij}^m x_j}{\sum_{j=1}^{n} u_{ij}^m}, \; 1 \le \mathrm{i} \le \mathrm{c} \tag{4}$$

4. Evaluate the terminating condition $\max_{1 \le i \le c} \left\{ \left\| v_i^{(t)} - v_i^{(t-1)} \right\| \right\} < \varepsilon$ (where ||.|| is the Euclidean norm). The iteration stops when it is satisfied, otherwise go to step 2.
5. Assign all pixels to each cluster according to the corresponding maximum membership values.

In order to improve the accuracy of clustering, it is necessary to compute the precise number of clusters and the initial values of centroid in the clusters. To compute the initialization of the aforementioned parameters, we utilized the empirical method in [11]. The value of centroid of each cluster is compared with the color of fire. The clusters, with the values of centroid approximated to the fire color, are selected for processing in the next step of the proposed method. If there is no suitable centroid, it can be concluded that the objects in the moving regions are not fire.

3.3 Parameter Extraction Using a Discrete Wavelet Transform

In the previous step, we can select the candidate fire regions from moving regions. However, the candidate regions still can be fire or non-fire because there are several moving objects whose color is the same as the color of fire, such as vehicles, people, and the light of vehicles. Thus, we utilize the discrete wavelet transform (DWT) algorithm to extract special parameters of fire for distinguishing between fire and non-fire. In this step, the candidate regions are divided into a block of 16x16 pixels. We then apply the discrete wavelet transform for each block.

Wavelets [12,13] are mathematical functions that decompose the data into different frequency components and study each component with a resolution matched to its scale. This is a fast, linear, and invertible orthogonal transform with the basic idea of

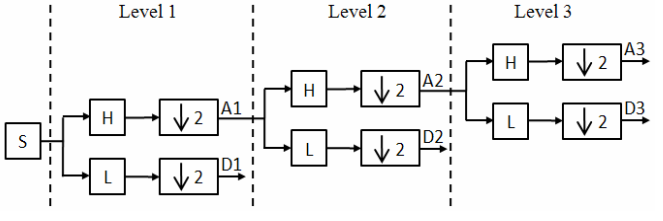


Fig. 1. Multi-resolution wavelet decomposition, where L is low pass decomposition filter, H is high pass decomposition filter, ↓2 is down sampling operation, A_1, A_2, A_3 are the approximated coefficients of the original signal S at level 1, 2 and 3, and D_1, D_2, D_3 are the detailed coefficients at levels 1, 2, 3

defining a time-scale representation of a signal by decomposing it onto a set of basic functions, called wavelets. The discrete wavelet transform is based on sub-band coding. It gives a time-scale representation of the digital signal using digital filtering techniques. The wavelet transform decomposition is computed by successive low-pass and high-pass filtering of the discrete time-domain signal [14]. The wavelet decomposition results in the levels of approximated and detailed coefficients. The algorithm of the decomposition of 3-level wavelet transform on the signal *S* is shown in the following Figure 1.

The decomposition of k-level wavelet transform on the original image will be represented by (3k+1) sub-images. The output of DWT we received is the parameters $[A_k, (H_i, V_i, D_i)_{i=1...k}]$, where A_k is a low resolution approximation of the original image, and H_i, V_i, D_i are the wavelet sub-images containing the image details in horizontal, vertical and diagonal directions, respectively, at the i-level decomposition. In this study, we applied that *k* equals to 3 for each block, resulting in 10 sub-blocks. At each level, the sub-blocks after transformation contain information in the horizontal, vertical and diagonal directions. However, the use of all these coefficients as features is exhaustive and time consuming for processing. In order to reduce the number of features and give better representation, six derived features are calculated from the coefficients of the sub-bands. The chosen six features for each sub-block are arithmetic mean, geometric mean, standard deviation, skewness, kurtosis and entropy, resulting in 60 parameters for the three levels. These 60 parameters are used as an input to the classifier of neural network in the following stage.

3.4 Fire Identification Using Back-Propagation Neural Networks

For the final stage of the proposed fire detection method, the back-propagation neural network (BPNN) is utilized to distinguish between fire and non-fire, which is one of commonly used neural network models in the fields of image processing, such as preprocessing, feature extraction, image segmentation, object classification, pattern recognition [15]. BPNN is composed of an input layer, one or more hidden layer, and an output layer. It is fully connected between upper and lower layer, and no connections between neurons in each layer. The topology of BPNN is shown in Figure 2.

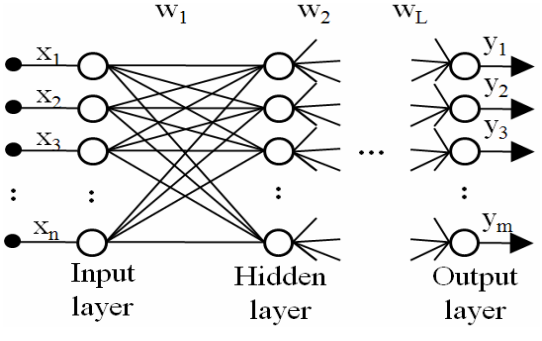


Fig. 2. The topology of the back-propagation neural network, where x_1, x_2,…x_n are input, y_1, y_2, …y_m are output and w_1, w_2, …, w_L are the weight matrix

During the training process, an input pattern or a set of patterns is presented to the network and propagated forward to determine the resulting signals at output units. The difference between the actual output and the desired output represents an error which is then back-propagated through the network in order to adjust the connection weights among artificial neurons in adjacent layers. Each weight is adjusted in proportion to the descent gradient of the sum of the squared errors, where the proportionality constant, η is called the learning rate. More detailed information of the training process is available at [16, 17].

In the previous section, for each sub-image of 16x16 pixels, the parameter vector whose dimension is 60 elements was extracted. This vector is used as an input vector of the BPNN. The dimension of BPNN's output vector is 1. Therefore, we select the 60-60-1 topology of BPNN as shown in Figure 3. The activation function, which is used in our proposed method, is log-sigmoid. Thus, the output value *y* of the network is constrained between 0 and 1. When *y* is close to 0, it means that the possibility of fire is low. On the other hand, when the output value *y* is close to 1, it means that the possibility of fire is high in the current sub-image. By observing several video clips, we make the following rule for fire or non-fire:

$$\begin{cases} Fire \quad if \ 0.75 < y \le 1 \\ Warning \quad if \ 0.35 \le y \le 0.75 \\ Non-Fire \quad if \ y < 0.35 \end{cases} \tag{5}$$

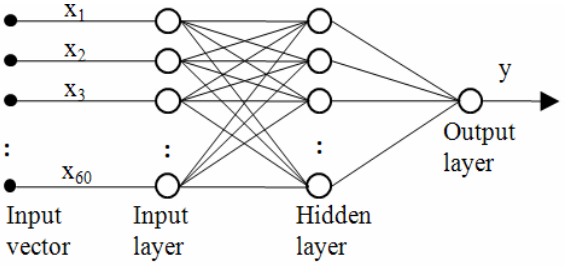
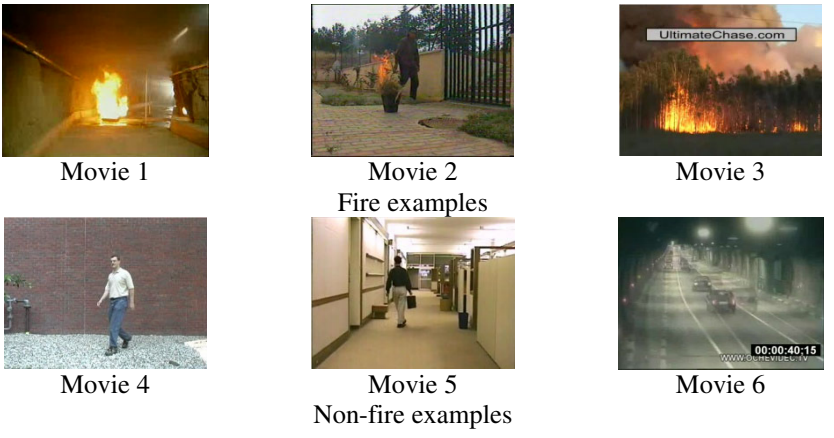


Fig. 3. The topology of the back-propagation neural network of the proposed method

4 Experimental Results

To evaluate the performance of the proposed fire detection method, we implemented the proposed method using MATLAB on a PC platform. The resolution of the selected movies is 320x240 pixels. With many simulations on several video clips, we selected optimal parameters for the proposed method as follows: the threshold *T* equals to 5, the exponent weight factor *m* equals to 2, the terminal condition ε equals to 0.001, and the learning rate η equals to 0.05. In addition, Daubechies second order moments (DB2) was chosen as mother wavelets for the discrete wavelet transform. To train back-propagation neural network, we simulated with several video clips which include fire or non-fire. The number of fire and non-fire samples is 25,000 and 15,000, respectively. Figure 4 shows examples of test videos used in this study.



Movie 1 Movie 2 Movie 3

Fire examples

Movie 4 Movie 5 Movie 6

Non-fire examples

Fig. 4. Examples of test videos

In addtion, we compared our proposed method with other four fire detection algorithms, called Algorithm 1 [3], Algorithm 2 [4], Algortithm 3 [5], and Algorithm 4 [6] with the same test videos. Table 1 shows the performance comparison of the proposed method and other fire detection algorithms. The propsed method outperforms other algorithms in terms of consistantly increasing accuracy of fire detection with fire movies and decreasing an error rate of fire detection incorrectly with non-fire movies. The proposed method was also tested with several videos with the same result

Table 1. Performance comparison of the proposed method and other fire detection algorithms

Algorithm	*Accuracy (%)*	*Error (%)*
Algorithm 1	93.72	3.42
Algorithm 2	93.76	3.85
Algorithm 3	94.27	2.63
Algorithm 4	94.36	2.87
Proposed Algorithm	**94.86**	**2.58**

5 Conclusion

In this paper, we proposed a novel approach for fire detection using FCM and BPNN. The proposed approach consists of four stages: (1) an approximate median method for detecting moving regions, (2) the FCM algorithm for selecting candidate fire regions from these moving regions based on the color of fire, (3) the DWT algorithm for deriving the approximated and detailed wavelet coefficients to be used as an input for the neural network, and (4) the back-propagation neural network for distinguishing between fire and non-fire. Experimental results showed that the proposed approach outperforms other state-of-the-art fire detection algorithms in terms of fire detection accuracy, providing a low false alarm rate and high reliability in open and large spaces.

Acknowledgement

This work (Grants No. 000406420110) was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2010.

References

1. Chen, T.H., Wu, P.H., Chiou, Y.C.: An early fire-detection method based on image processing. In: IEEE. Int. Conf. on Image Processing, vol. 3, pp. 1707–1710 (2004)
2. Toreyin, B.U., Centin, A.E.: Online detection of fire in video. In: IEEE. Conf. on Computer Vision and Pattern Recognition, pp. 1–5 (2007)
3. Celik, T., Demirel, H.: Fire detection in video sequences using a generic color model. Fire Safety Journal 44, 147–158 (2009)
4. Ko, B.C., Cheong, K.H., Nam, J.Y.: Fire detection based on vision sensor a support vector machines. Fire Safety Journal 44, 322–329 (2009)
5. Toreyin, B.U., Dedeoglu, Y., Gudukbay, U., Centin, A.E.: Computer vision-based method for real-time fire and flame detection. Pattern Recognition Letter 27, 49–58 (2006)
6. Borges, P.V.K., Izquierdo, E.: A probabilistic approach for vision-based fire detection in videos. IEEE. Trans. on Circuits and Systems for Video Technology 20, 721–731 (2010)
7. McFarlane, N.J.B., Schofield, C.P.: Segmentation and tracking of piglets in images. Machine Vision and Application 8(3), 187–193 (1995)
8. Bezdek, J.C.: Pattern recognition with fuzzy objective function algorithms. Pleum Press, New York (1981)
9. Bezdek, J.C., Keller, J., Krisnapuram, R., Pal, N.: Fuzzy models and algorithms for pattern recognition and image processing. Springer, Heidelberg (2005)
10. Ohta, Y., Kanade, T., Sakai, T.: Color information for region segmentation. In: Computer Graphics and Image Processing, pp. 222–241 (1980)
11. Tan, K.S., Isa, N.A.M.: Color image segmentation using histogram thresholding – fuzzy c-means hybrid approach. Pattern Recognition 44, 1–15 (2010)
12. Chan, Y.T.: Wavelet basics. Springer, Heidelberg (1994)
13. David, F.W.: An introduction to wavelet analysis. Brikhauser, Boston (2001)
14. Mallat, S.G.: A theory for multiresolution signal decomposition: the wavelet representation. IEEE. Trans. on Pattern Analysis and Machine Intelligence 11(7), 674–693 (1989)
15. Petersen, M.E., Ridder, D.D., Handels, H.: Image processing with neural networks – a review. Pattern Recognition, 2279–2301 (2002)
16. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. In: Rumelhart, D.E., McClelland, J.L. (eds.) Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1, pp. 319–362. MIT Press, Cambridge (1986)
17. Lippmann, R.P.: An introduction to computing with neural nets. IEEE ASSP Magazine 4, 4–22 (1987)

Multiple Kernel Active Learning for Facial Expression Analysis

Siyao Fu*, Xinkai Kuai, and Guosheng Yang

School of Information and Engineering, the Central University of Nationalities, Beijing 100081, China

Abstract. Multiple Kernel Learning (MKL) approaches aim at determine the optimal combination of similarity matrices (since each representation leads to a different similarity measure between images, thus, kernel functions) and the optimal classifier simultaneously. However, the combination of "passive" kernels learning scheme limits MKL's efficiency because side information is provided beforehand. A framework of Multiple Kernel Active Learning (MKAL) is presented in this paper, in which the most informative exemplars are efficiently selected by min − max algorithm, the margin ratio is used for querying next instance. We demonstrate our algorithm on facial expression categorization tasks, showing that the proposed method is accurate and more efficient than current approaches.

Keywords: Active Learning, Supervised Learning.

1 Introduction

Kernel based learning is arguably the most crucial issue concerned in the machine learning field. Recent works [1, 2, 3, 4] have shown that using a sparse combination of multiple kernels can enhance interpretability of the decision function and improve classifier performance. By specifying the coefficients for the classifier and the weights for the kernels in a convex optimizing problem in linear case, the algorithm searches for the linear combination of base kernel functions which maximizes a generalized performance measure (see fig. 1 for systematic illustration), which is known as the multiple kernel learning (MKL) problem. However, MKL is plagued by its inefficient learning procedure. During the training process, labeling failure instances (common at early training stage) to create a training set is overwhelmingly time consuming for the random sampling scheme for traditional, passive SVM used as classifier, thus, finding modified enhancement to minimize the number of labeled instances for MKL framework is by no means beneficial and necessary. Unfortunately, little works have been reported on that, which is crucial for learning efficient classifier over large-scale dataset.

* This work was supported in part by 985 Funding Project (2nd Phase) of Central University of Nationalities (Grant 98501-00300107) and Beijing Municipal Public Information Resources Monitoring Project (Grant 104-00102211).

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 381–387, 2011.

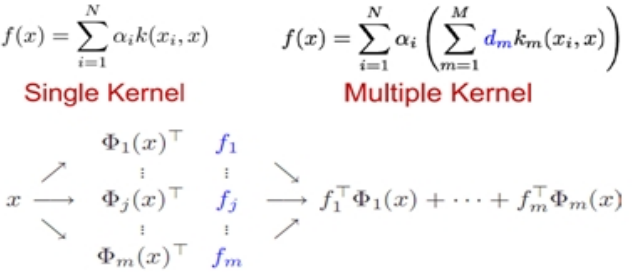


Fig. 1. MKL framework [6]

So what is the alternative? A natural choice would be seeking the trade-off between sparsity and efficiency, thus, we try to maintain the sparse essence of the MKL while simultaneously reducing computational burden, facilitating the learning procedure.

In this paper, we present an efficient approach for the multiple kernel learning problem, active learning approach is introduced, particularly, a pool based active learning algorithm for choosing which instances to request next in order to reduce the learner's need for large quantities of labeled data, and avoid the potential poor performance for randomly selecting labeled training set. The experiment results verify the effectiveness of our approach.

After submission of this paper, we learned about a related approach proposed in [8], in which the MKL proposed in [4] and active learning approach had been combined together to form a framework for image classification. However, the main difference between our work and theirs lies in that we choose the MKL structure proposed from [2], which has been reported gaining more efficiency in learning, especially for large scale database, and we adopt pool based active learning scheme (using margin ratio query method) from [5].

2 Overview of the Approach

In general, MKL falls into the category of finding sparse solution, which is being regard as one of the hot topics of research interest in machine learning. And active learning, in essence, seeks for the most informative sample during the training procedure. A natural choice is to try the combination of both methods. But doing so we still have to answer the three subsequent questions:

(1): What kinds of the MKL framework (with different regularization terms) is the best and should be chosen?

(2): Is there an efficient algorithm for solving the problem?

(3): How is active learning combined with the learning procedure?

In the following section, our aim is to provide a satisfactory answer to the above questions. We first give MKL learning framework, including the prime problem and dual problem, then a detailed solution is presented. Finally, active learning is encapsulated to the learning framework.

Algorithm 1. The multiple kernel active learning algorithm

Input: Gram matrix K_0, constraint matrices A_i
$k = 1$
$\beta_k^1 = \frac{1}{M}$ for k=1,...,M
Define map $P = \arg\min\{||z - \beta||, z \in X\}$
repeat
  for $t = 1, 2, ...$ do
  Solve the Active SVM with $K = \sum_k \beta_k^t K_k$ (see algorithm 2)
  Compute the gradient $\frac{\partial J}{\partial \beta_k}$ using standard method, for $k = 1, 2, ..., M$
  $\beta_k^{t+1} \leftarrow \beta_k^t + \tau_t D_{t,k}$
  Map to X we obtain
  $\beta_k^{t+1} \leftarrow P(\beta_k^t + \tau_t D_{t,k})$ for map P
  Find the step size τ for the direction $D_{t,k}$.
  if stopping criterion **then**
    break
  end if
  end for
until $\|\beta_k\| < \epsilon$

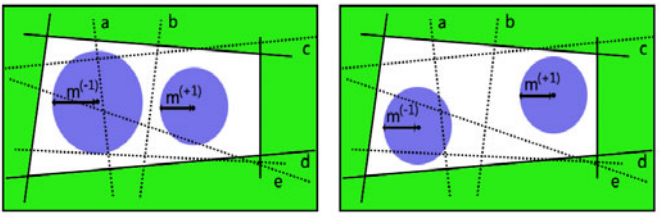


Fig. 2. (a) MaxMin Margin will query b. The two SVMs with margins m^- and m^+ for b are shown. (b) Ratio Margin will query e. The two SVMs with margins m^- and m^+ for e are shown. See Tong's work in [5] for a detailed illustration.

2.1 MKL Prime Problem

When dealing with multiple kernels, in [6] the primal model of MKL can be formulated as follows,

$$\min \ \frac{1}{2}\left(\sum_{k=1}^{K}||w^k||_2\right)^2 + C\sum_{i}^{N}\xi_i \tag{1}$$

$$s.t. \quad y_i\left(\sum_{k=1}^{K}(w^k, \psi_k(x_i)) + b\right) \geq 1 - \xi_i, \tag{2}$$

$$\xi_i \geq 0. \forall i \in 1, ..., N$$

for which multiple kernels K_i belong to different RKHS $\mathcal{H}_i$. MKL is actually the form of $k(x_i, x_j) = \sum_{k=1}^{K} \beta_k k_k(x_i, x_j)$, where β_i denotes the weights (positive) associated with each kernel. So the problem can be reformulated as

Algorithm 2. SVM with active learning scheme

Input: labeled dataset X, label set $+1, -1$, unlabeled instances pool **U**
Define active learner **AL** with three components (f, q, X)
Define classifier $f : X \rightarrow -1, +1$, querying function $q(X)$
repeat
for $i = 1, 2, ...$ do
$V_i^+ = V_i \cap \{\omega \in W| + (\omega \cdot \Phi(x_{i+1})) > 0\}$
$V_i^- = V_i \cap \{\omega \in W| - (\omega \cdot \Phi(x_{i+1})) < 0\}$
$m^+ \leftarrow$ radius of V_i^+, $m^- \leftarrow$ radius of V_i^-
Training **AL** using ratio margin function $\max\min(\frac{m^-}{m^+}, \frac{m^+}{m^-})$
if stopping criterion **then**
break
end if
end for
until $\|\beta_k\| < \epsilon$

$$\min_{f_i, b, \xi, \beta} \frac{1}{2}\left(\sum_i \frac{1}{\beta_i}||f_i||^2_{\mathcal{H}_i}\right) + C\sum_i \xi_i \tag{3}$$

$$s.t. \quad y_i\left(\sum_{i=1}^{m} f_i(x_i) + b\right) \geq 1 - \xi_i, \forall i \in 1, ..., m$$

$$\sum_i \beta_i = 1, \beta_i \geq 0, \forall i \in 1, ..., m$$

The non-negative parameter β_i renders that the combined regularizer is convex, and so the resulting kernel is positive semi-definite. ξ is slack variables for soft margin. We refer the readers to [7] for the details of MKL.

2.2 Solving the Problem

The coefficients and kernel weights can be learnt by solving a joint dual optimization problem as follows:

$$\min\max J = \min\max\left(-\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \sum_{i=1}^{m}\beta_i k_i(x_i, x_j) + \sum_i \alpha_i\right) \tag{4}$$

with $\beta_i \geq 0, \sum_i \alpha_i y_i = 0, 0 \leq \alpha_i \leq C, \forall i \in 1, ..., m$. Recall in the formulation section the combined kernel $K(x, y)$ takes the form of $\sum_i \beta_i K_i(x, y)$, then it can be seen as the standard SVM dual problem's formulation, and can be solved by any classical SVM algorithm. Detailed algorithm refers to [6] and algorithm 1.

2.3 Active Learning Scheme

As discussed earlier, MKL tries to find the sparse solution for the learnt kernel weights, while simultaneously retaining computational burden, however, this

do not always perform well, especially for the sample selection part in large-scale tasks. During the training procedure, the conventional random training set selection is plagued for causing labeling instances to create training set time-consuming and costly. A natural choice would be the introducing the active learning, which tackles the problem of finding the most crucial data in a huge set of unlabeled examples so that the classification system maximizes the benefit of the discriminatory capability given the label of that example. Since in this paper we only explore categorization using MKL. Active learning strategy is employed in this paper to make an optimal data selection through performing limited number of queries. The motivation comes from Tong *et al* [6], the max − min Margin approximation approach is adopted in the paper, the idea is to query the unlabeled exemplar with the largest $\min(m^+, m^-)$ approximated by choosing the instance with the largest margin ratio $\min(\frac{m^-}{m^+}, \frac{m^+}{m^-})$ (m^-, m^+ stands for the margin obtained by adding point x as $+1$ or -1, respectively.), see algorithm 2 for details. Fig.2 illustrates the two simple cases (original max min margin and margin ratio, respectively, we choose latter for the experiment).

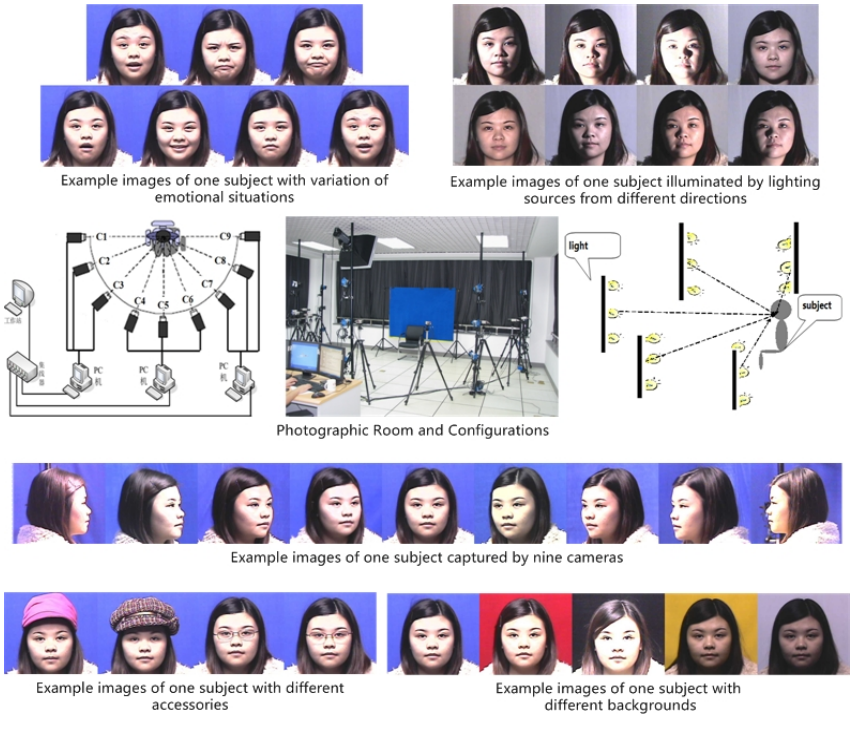


Fig. 3. Diagram showing the whole configuration of CUN face database

3 Experiments

The experiment we consider here is the evaluation of the approach on a newly created face database we have designed and constructed, namely, a large-scale racially diverse face database, the CUN face database, which covers different source of variations, especially in race, facial expression, illumination, backgrounds, pose, accessory, etc. Details please refer to [7]. We choose a subset which contains the typical facial expressions for categorization. For efficiency, we limit the number of test images to 50 per class. 280 images from 7 categories (typical facial expressions for 10 subjects) are mixed up to form the unlabeled data pool for active learning. The remaining 70 images are used for testing. The image is decomposed into sets of SIFT features, which is used as low-level features, the dimensionality is reduced by PCA to 10.

The experiment investigates the overall categorization ability among traditional MKL and MKL with active learning. At the stage of active learning, we

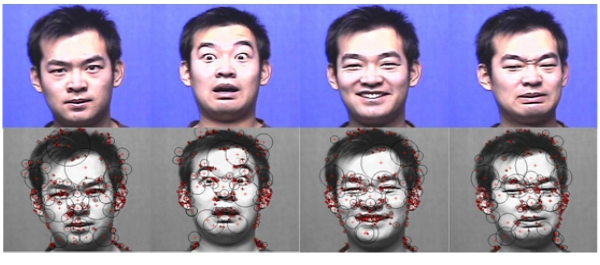


Fig. 4. Sample facial expression and SIFT feature descriptor detection results. Note that the most discriminative feature points are scatted around the eyes, mouth, nose, all are essential for detecting facial expression variations.

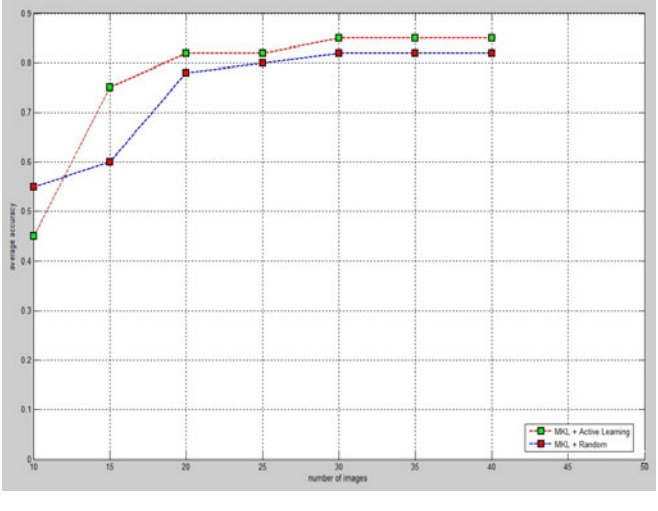


Fig. 5. Empirical comparison results

query and label 35 images (on average 5 samples for each of the 7 categories) at each cycle in total 5 rounds. Fig. 5 shows that active learning based MKL performs competitively with the original one and using active learning further improves the performance. In fact we can see that a mean accuracy per class close to a 82% can be obtained with just 20 labeled examples, whereas the non-active learners achieve around 78% accuracy for the same amount of labeled data, the difference may be attributed to the benefit of MKAL in selecting more informative samples than the random selection in each round. This demonstrates that active learning can provide a significant boost in accuracy, and makes it possible for the case where labeled data are rare and expensive while unlabeled data are relatively easy to acquire.

4 Conclusions

Kernel learning methods can be viewed as a computational shortcut which makes it possible to represent linear patterns efficiently in high-dimensional feature spaces to ensure adequate representation power. We observe that compared with MKL using traditionally random sampling approach, the performance of MKAL is satisfactory using less but specifically selected data with lower computational burden as well. We conclude that the introduction of an active learning paradigm for kernel based learning can optimally selects unlabeled test points for interactive labeling, with active learning small amounts of interactively labeled data one can provide very accurate categorization performance.

References

1. Lanckriet, G., Cristinanini, N., El Ghaoui, L., Bartlett, P., Jordan, M.I.: Learning the kernel matrix with semi-definite programming. Journal of Machine Learning Research, 27–72 (2004)
2. Rakotomamonjy, A., Bach, F., Canu, S., Grandvalet, Y.: More efficiency in Multiple Kernel Learning. In: Proceedings of Twenty-Fourth International Conference on Machine Learning (2007)
3. Bach, F.R., Lanckriet, G.G., Jordan, M.: Multiple kernel learning, conic duality, and the SMO algorithm. In: Proceedings of the Twenty-First International Conference on Machine Learning, pp. 775–782 (2004)
4. Sonnerburg, S., Raetsch, G., Schaefer, C., Scholkopf, B.: Large scale multiple kernel learning. Journal of Machine Learning Research, 1531–1565 (2006)
5. Tong, S., Koller, D.: Support Vector Machine Active Learning with Applications to Text Classification. Journal of Machine Learning Research, 45–66 (2001)
6. Fu, S.Y., Yang, G.S., Hou, Z.G.: Image category learning and classification via optimal linear combination of multiple partially matching kernels. Soft Computing 14(2) (2010)
7. Fu, S.Y., Yang, G.S., Hou, Z.G.: Multiple Kernel Learning with ICA: Local Discriminative Image Descriptors for Recognition. In: Proceedings of International Joint Conference on Neural Networks (2010)
8. Yang, J.J., Li, Y.N., Tian, Y.H., Duan, L.Y., Gao, W.: Multiple Kernel Active Learning For Image Classification. In: Proceedings of the 2009 IEEE International Conference on Multimedia and Expo (2009)

Fast Moving Target Detection Based on Gray Correlation Analysis and Background Subtraction

Zheng Dang, Songyun Xie, Ge Wang, and Fahad Raza

School of Electronics and Information
Northwestern Polytechnical University, Xi'an 710129, P.R. China
syxiegm@gmail.com

Abstract. Nowadays, a higher accuracy with lower time consuming detection method is required extremely. To achieve that, a moving target detection method based on gray correlation analysis and background subtraction is proposed in this paper. It mixes them together in order to improve the quality of the results. From the experiments, the method shows a much higher accuracy of detection and lower time consuming. In addition, the algorithm is insensitive to noise and the shadow, and can be used in real-time processing.

Keywords: Real-time moving target detection, background subtraction, gray correlation analysis.

1 Introduction

In recent years, with the rapid development of computer technology, moving target detection has been more and more popular in military, bank management, traffic monitor and so on. It is the base of target recognition and tracking. Accuracy and time consuming are concerned. So, a good algorithm in moving target detection must have a higher accuracy of detection and a lower time consuming.

Traditional methods, such as background subtraction, frame-to-frame differences, optical flow, are also used. But they have appeared many limitations. The background subtraction and frame-to-frame differences have a much lower time consuming, but lack of accuracy while the optical flow has a higher accuracy but spending a lot of time [1]. Wei Zhang, Q.M. jonathan use adaptive motion histogram to detect the moving vehicles [2]. It takes the advantage of the gray histogram information in order to find out the foreground picture. Li Yi, Zhengxing Sun mix the frame-to-frame difference method and background subtraction method together [3]. It has a good result of denoising. Aurelie Bugeau, Patrick Rerez use background modeling and background subtraction method to detect the moving object in complex environment [4] and its improved denoising method makes it be used in complex environment.

Gray correlation analysis, which is first inferred by J. Deng, has been used in many areas for its superiority in lack of recognition researches. It has a high accurate result in edge detection [5] and moving target detection [6]. But it costs a lot of time to process. So, how to improve the time consuming is an important point in detection area using gray correlation analysis.

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 388–394, 2011.

In this paper, a real-time moving target detection method is firstly proposed and studied. Based on background subtraction and gray correlation analysis, we can get a high accurate result with low time consuming. Because gray correlation analysis needs to calculate the degree of correlation from every pixel between background image and foreground image, it spends a lot of time. If we use background subtraction to detect the target first and then use gray correlation analysis to process the pixels that the gray levels are not zero (target or noise), we can accelerate the algorithm. For one hand, background subtraction decreases its time of processing. For another, gray correlation analysis ensures the detection accuracy. Therefore, a fast moving target detection method can be achieved.

2 Method

2.1 Gray Correlation Analysis Theory

Gray correlation analysis (GCA) is a method that describes the variation trend of a system and makes a comparison [7]. Its aim is to find out the degree of correlation between each element. Then, catch the features of the elements accordingly. The basic idea of GCA is to figure out the similarity from the geometry of the data sequences in order to judge whether they are similar.

The steps of processing using GCA are described as follow. First, establish the gray system. Ensure the features that are suitable for the objects. It is so important that it can impact the results good or not. Second, establish the reference sequences and comparison sequences. Then, find out the gray correlation coefficient. The gray correlation coefficient is defined by

$$\delta\left(x_j(k), y_i(k)\right) = \frac{\min\limits_i \min\limits_j \left|x_j(k) - y_i(k)\right| + \zeta \max\limits_i \max\limits_j \left|x_j(k) - y_i(k)\right|}{\left|x_j(k) - y_i(k)\right| + \zeta \max\limits_i \max\limits_j \left|x_j(k) - y_i(k)\right|} \tag{1}$$

where $x_j(k)$ is the reference sequence. $y_i(k)$ is the comparison sequence. ζ is the resolution coefficient which is between 0 to 1 (normally is 0.5). At last, calculate the degree of gray correlation. We defined the degree of correlation between two systems or two elements by

$$\gamma_{ji} = \gamma\left(x_j, y_i\right) = \frac{1}{N}\sum_{i=1}^{N} \delta\left(x_j(k), y_i(k)\right) = \frac{1}{N}\sum_{i=1}^{N} \delta_{ji}(k) \tag{2}$$

where $\gamma\left(x_j, y_i\right)$ is the degree of correlation. N is the length of the data.

After having done these steps, we can get the degree of gray correlation between two elements. From the degree of gray correlation, we can judge whether the two data sequences are similar.

2.2 Background Subtraction Method

Background subtraction is widely used in engineering area recently. It can be used in real-time image processing. The idea of this method is to use the current image (I_t)

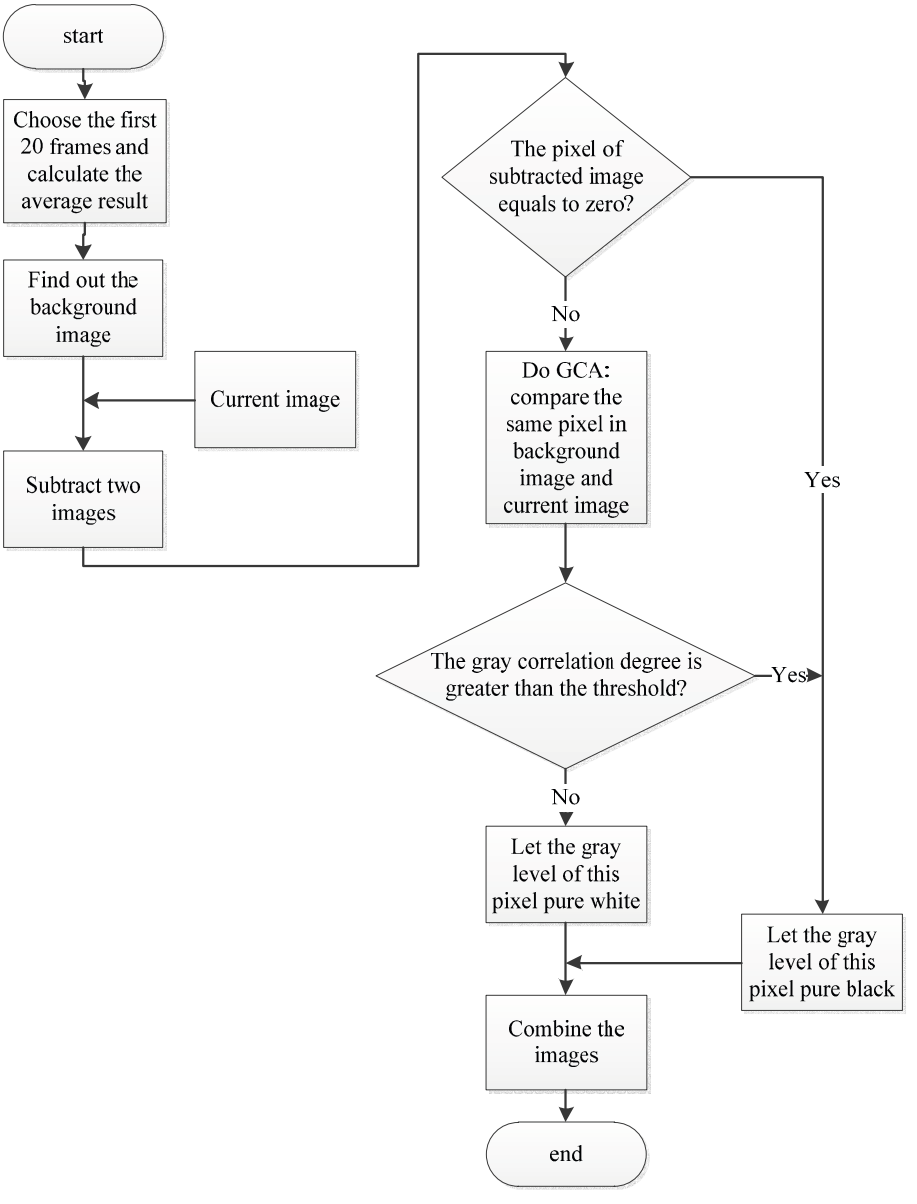


Fig. 1. The FGCA algorithm flow chat

and background image (B) and then, find out the different area as its moving area. Background subtraction is defined by

$$F_t(x,y)=\begin{cases}1 & if\ \left|I_t(x,y)-B(x,y)\right|\geq T\\ 0 & \text{el se}\end{cases} \tag{3}$$

Where (x, y) is the coordinate of the image matrix. $F_t(x, y)$ is the foreground area (include moving object, cast shadow and other noise). T is the threshold (this paper uses OTSU method as its threshold method [9]).

From Eq.3, we can see much clearly that the background subtraction method has a higher speed because of its simple algorithm. However, the quality of the result may be affected by threshold and the quality of the image. So, the detection may often have much noise. Also, this method couldn't remove the cast shadow which caused by the moving target.

2.3 Moving Target Detection Based on Gray Correlation Analysis and Background Subtraction

Background subtraction [8], as we all know, is a fast method in moving target detection. But its detective accuracy is not good. Gray correlation analysis can achieve a good detection for its high accuracy and insensitive to noise. So, mix them together can get a much better detection not only in accuracy, but also in time consuming.

Fig. 1 shows the whole detection processing flow based on gray correlation analysis and background subtraction (here, we called this algorithm as FGCA).The steps of moving detection algorithm based on gray correlation analysis and background subtraction is as follow:

First, calculate the background image with accumulating the first 20 frames, getting the average result as its background image.

Second, make a subtraction between background image and current image. Because background subtraction may have a low quality image if the original image is full of noise. Any gray level of pixel which is not zero is considered as the moving target or the noise.

Then, use GCA to process these pixels in order to increase its accuracy (removing the noise including the shadow). After having finished this step, a degree of gray correlation image will be created.

At last, a threshold algorithm (this paper use OTSU algorithm [9]) is used to segment.

Therefore, we use not only GCA to make sure the accuracy of detection, but also background subtraction to decrease its time consuming.

3 Experimental Results

The video is taken indoors. There are 240*320 pixels in the images. We convert every RGB frames into gray-scale image and each pixel is represented by 256 gray-scales. This issue requires MATLAB environment.

Due to moving target detection, we suppose the background image as its comparison image, the current image as its reference image. Fig.2 (a) shows the original background image and Fig.2 (b) displays the current image. The background image is captured by the first 20 frames (getting their mean image).

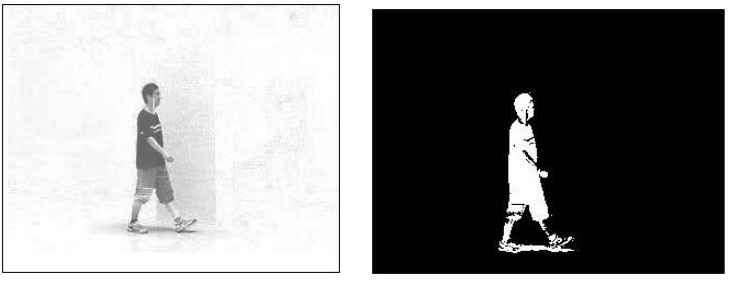
(a) background image (b) current image

Fig. 2. The preprocessed background image and the current image

First, use background subtraction to pre-detect. Fig.3 shows the result of this step.

Fig. 3. Background subtraction pre-detection

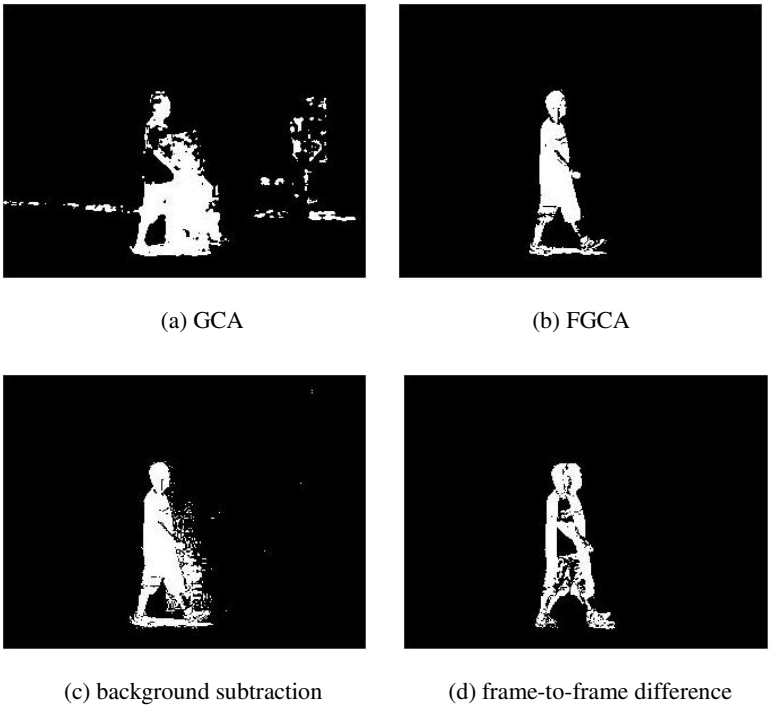
Second, use GCA to detect the target. Fig.4 shows the result of this step where Fig.4 (a) gives the gray correlation analysis degree image and Fig.4 (b) shows the thresh result.



(a) (b)

Fig. 4. The threshold result of detection

Fig.5 shows the comparison with other traditional methods in accuracy where GCA is the result using Li Nan's method [6]. Fast Gray Correlation Analysis (FGCA) is the result using GCA and background subtraction.



(a) GCA (b) FGCA

(c) background subtraction (d) frame-to-frame difference

Fig. 5. Comparison with traditional methods

Table 1. Time consuming of each method (/s)

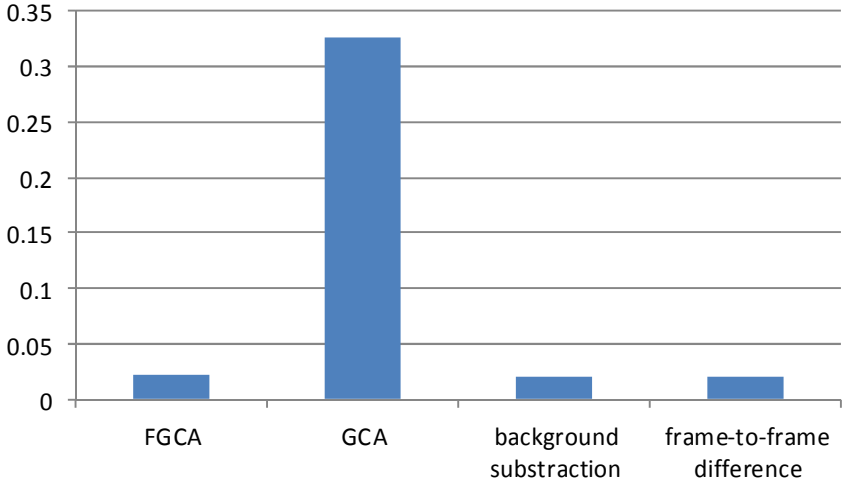


Fig. 5 illustrates that using the method this paper proposed is much clearer with a higher accuracy. Table.1 shows the time consuming of each method which has been mentioned above.

From table.1, it gives the time of processing to each frame. We can see that using FGCA is faster than GCA. But it takes a little more time than background subtraction and frame-to-frame differences (because it gives up a part of time to increase the detective accuracy). The time of processing is nearly 20ms. So, it can be used in real-time image processing.

4 Conclusion

The moving target detection based on gray correlation analysis and background subtraction is first proposed and studied in this paper. The method gives a much higher accuracy and lower time consuming. It can be used in real-time image processing. Also, from the result, we can see much clearly that the cast shadow can be removed and this method is insensitive to noise. But, this method doesn't include the background adaption block. It couldn't be used in the situation that the background is changing. Our further research is to find a proper background adaption algorithm to solve this problem.

References

1. Ding, Z.X.: Survey on Moving Object Detection Methods for Video Surveillance Images. Video Application & Project 32, 72–76 (2008) (in chinese)
2. Zhang, W., Wu, Q.M.J., Yin, H.B.: Moving vehicles detection based on adaptive motion histogram. Digital Signal Processing 20, 793–805 (2010)
3. Li, Y., Sun, Z.X., Yuan, B., Zhang, Y.: An improved method for motion detection by frame difference and background subtraction. Journal of Image and Graphics 14(6), 1162–1168 (2009)
4. Bugeau, A., Rerez, P.: Detection and segmentation of moving objects in complex scenes. Computer Vision and Image Understanding 113, 459–476 (2009)
5. Ma, M., Fan, Y.Y., Xie, S.Y.: A novel algorithm of image edge detection based on gray system theory. Journal of Image and Graphics 8, 1136–1139 (2003) (in chinese)
6. Li, N., Xie, S.Y., Xie, Y.B.: Application of gray relational theory in moving object detection from video sequences. Journal of Computer-aided Design & Computer Graphics 21, 663–667 (2009) (in chinese)
7. Deng, J.L.: Gray system basic method. Central China University for Science and Technology Press, Wuhan (1996) (in chinese)
8. Chen, P.J.: Moving Object Detection Based on Background extraction. In: Computer Network and Multimedia Technology, CNMT 2009 (2009)
9. Huang, D.Y., Wang, C.H.: Optimal multi-level thresholding using a two-stage Otsu optimization approach. Pattern Recognition Letters 30, 275–284 (2009)

Shadow Removal Based on Gray Correlation Analysis and Sobel Edge Detection Algorithm

Feng Ji[1], Xinbo Gao[1], Zheng Dang[2], and Songyun Xie[2]

[1] School of Electronic Engineering, Xidian University,
Xi'an 710071, P.R. China
drjif@189.cn
[2] Northwestern Polytechnical University, School of Electronics and Information,
Xi'an 710129, P.R. China
syxiegm@gmail.com

Abstract. Nowadays, moving target detection plays an important part in our daily life. Time consuming and detective accuracy are concerned. Gray correlation analysis, which has been proved to be used in moving object detection, really has a high accuracy. But, this higher accuracy often brings cast shadow disturbance and it is always misclassified as part of moving target. In order to solve this problem, an efficient cast shadow removal method using Sobel edge detection algorithm in moving target detection is proposed in this paper. The experiments show this method can remove the cast shadow efficiently and it can be used in real-time processing.

Keywords: Moving target detection, cast shadow removal, edge detection, gray correlation analysis.

1 Introduction

Shadow is popular in natural world. It is created by the light source which is kept out by object. Shadow may produce different effect on different computer vision areas. Some are advantages. For example, shadow can increase the stereoscopic impression and sense of reality in virtual reality or 3D games. But in most cases, it may bring a lot of disadvantages. In aviation imaging area, the existing cast shadow may influence the remote sensing of image processing, such as image matching, pattern recognition and physical extracting. In medical imaging area, shadow may influence the judgment of diseased image for doctors. In video monitor area, shadow may influence the result of moving target detection and leads to missing moving target in tracking area. So, it is necessary to analysis the shadow in moving target detection, weaken or even eliminate the shadow.

In recent years, there are many achievements in cast shadow removal area. Elena Salvador put forward a proposal in using photometric color invariants to remove the cast shadow [1]. Because the invariant color features of image are not sensitive to the light, change the RGB space into any spaces that are not sensitive to light can eliminate the shadow. It is a fast algorithm. But single light source, the object and the shadow coexisting and smooth surface of the shadow area constraints the algorithm.

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 395–401, 2011.

In Kobus Barnard and Martin D. Levine's paper, they use color ratio feature to detect the shadow [2]. The algorithm is simple and efficient. It needs all the color ratio values although it is impossible. So, it leads some edges of shadow not be detected. G.D. Finlayson uses illuminant invariant image to detect the shadow [3] [4]. It can get a much complete edge in complex scenes. But it needs the photographic coefficient of the camera. Different camera may have different photographic coefficient. What's more, he and C.Fredembach use one-dimensional integral based on Hamiltonian path to remove the shadow [5]. Using the algorithm can get a good result of shadow removal but with a high complexity.

Gray correlation analysis really has a high accuracy in moving object detection [6]. Because of this, the cast shadow of the target also will be detected although it is not the real target. In this paper, we present a new method for cast shadow detection and removal in moving target detection. First, use gray correlation analysis to detect the targets (which contains the shadow). Then, use the method which this paper referred to remove the cast shadow. The method improves on increasing the degree of cast shadow detection with little time consuming. We can use it to not only detect the cast shadow, but also remove the shadow. Experiments on the video sequences show that our method has a much better detecting result in cast shadow removal and can be used in real-time processing.

2 Method

We set gray correlation analysis as its moving target detection algorithm. The algorithm is first inferred by Julong Deng. It has been widely used in security science, medical diagnosis and so on. Also, the gray correlation analysis has been used in image processing [6] [7] and shows a high accuracy.

The method begins from the gray correlation analysis. First, use gray correlation analysis to get the gray correlation degree image. It is an image that shows the relation degree between the background image and the foreground image. So, the pixels which have a lower relation degree are the moving target. Figure.1 shows the gray correlation degree image. Figure.2 shows the threshold image using OTSU algorithm [8]. We can see there is so much cast shadow that is detected in figure.2.

(a) original image (b) gray correlation degree image

Fig. 1. Gray correlation degree image

Fig. 2. Threshold image

Second, after getting the gray correlation degree image, we need to extract the contour of the moving target. There are so many edge detection algorithms, such as Robert algorithm, Sobel algorithm, Prewitt algorithm, LOG algorithm, Canny algorithm and so on [8]. The edge detection algorithm, we want, is to find the edge of target and not sensitive to the cast shadow. Sobel algorithm is the edge detection algorithm this paper is used. Its aim is to find the image edge information from horizontal direction or vertical direction. Because Sobel algorithm is a 3*3 convolution kernel, it is good at the gray scale of pixels that are gradual change. In the gray correlation degree image, background is extremely weakened. But if we use other algorithms, the background can also be detected. After do some experiments, we select Sobel algorithm as its edge detection algorithm.

Third, an edge connection is needed. This step gives service to fill the target. In order to fill the target, we have to connect the contour of the target.

At last, fill the target with pure white (show the target area and remove the shadow).

Figure.3 shows the flow of the method this paper mentioned.

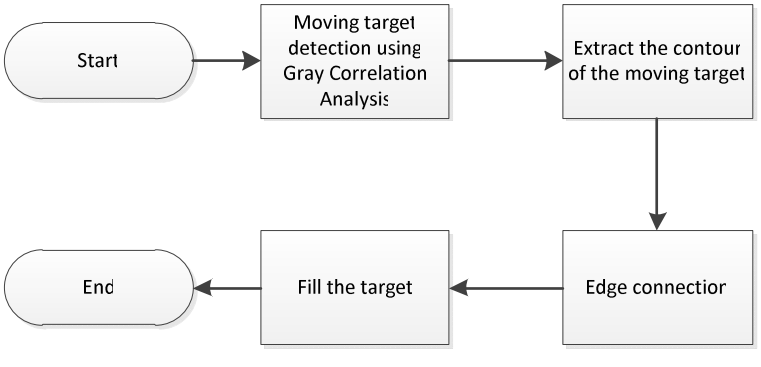


Fig. 3. Flow of the method

3 Experimental Results

The video is downloaded from the Internet [10]. There are 240*320 pixels in the images, and each pixel is represented by 256 gray-scales. This issue requires MATLAB environment.

First, we give the background image and the foreground image. Figure.4 (a) shows the original background image and Figure.4 (b) shows the current (foreground) image.

(a) (b)

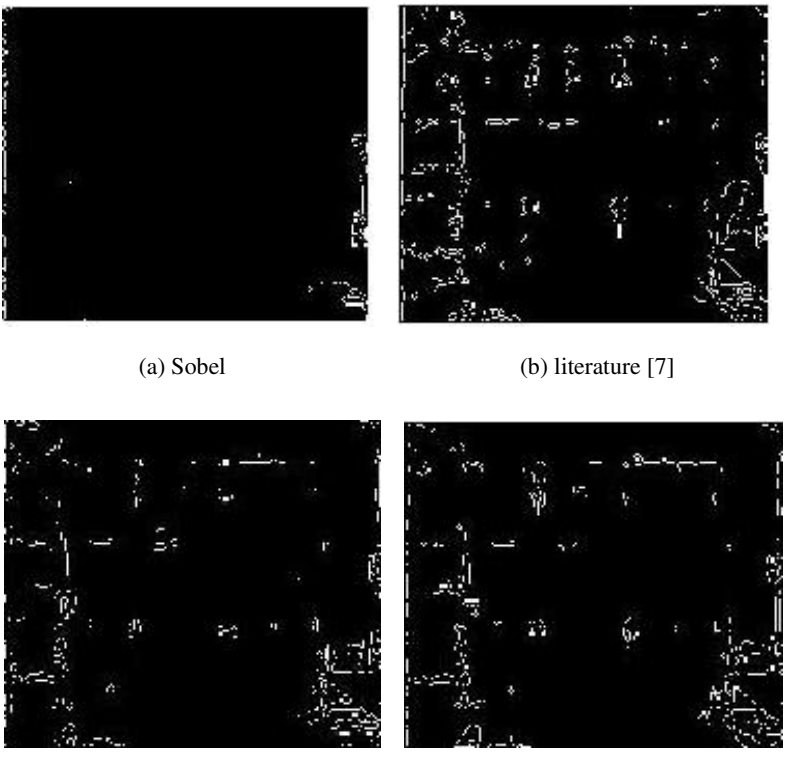
Fig .4. Background and foreground image

Then, the result that Li Nan's paper inferred is given [6]. Figure.5 shows the gray correlation degree image and its threshold image.

(a) gray correlation degree image (b) threshold image

Fig. 5. Literature [6] result

Third, use edge detection algorithms to detect the edge of the moving target. Figure.6 shows the results of this step. From the picture, we can see using Sobel algorithm is better than any other algorithms because it can detect the edge of the target and not sensitive to the cast shadow. So, we choose Sobel algorithm as its edge detection algorithm.

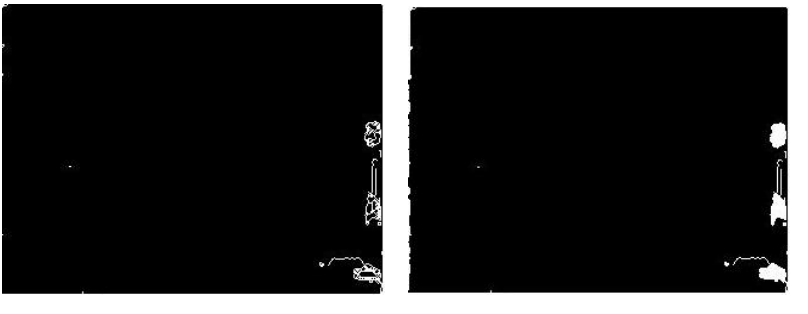


(a) Sobel (b) literature [7]

(c) LOG (d) Canny

Fig. 6. Results of each edge detection algorithm

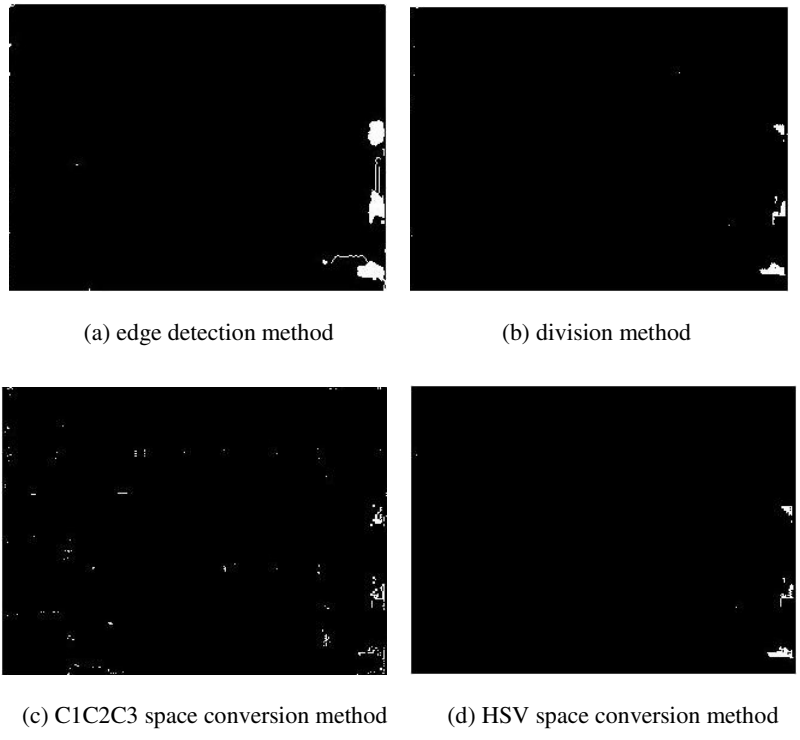
Forth, edge connection. Figure.7 (a) shows the result. At last, fill the image. After have done this step, the cast shadow removal will be finished. Figure.9 (b) gives the result.



(a) (b)

Fig. 7. The result of edge connection and filled image

From the result Figure.7 (b), we can see that the cast shadow is removed efficiently. The computer spends 0.382048s to run the whole steps. Figre.8 shows the comparison between the method this paper proposed and any other traditional method. Figure.9 gives the result corresponding to the original foreground image.



(a) edge detection method (b) division method

(c) C1C2C3 space conversion method (d) HSV space conversion method

Fig. 8. The comparison between each method

Fig. 9. Corresponding to the foreground image

4 Conclusion

Today, cast shadow removal is a very hot research. How to remove the shadow efficiently and fast is concerned. In this paper, first, we use gray correlation analysis as its moving target detection method. Then, a cast shadow removal method using Sobel edge detection and morphological method is proposed. From the experiment, we can see that the cast shadow is removed efficiently and it can be used in real-time image processing.

To use this method, we need to conduct gray correlation analysis method to detect the moving target first. It has a pretty good result in shadow removal. Also, we can see from Figure.8 (a) that there are some noises in shadow removal results. So, further research is to increase its accuracy of the algorithm and let this algorithm be used much robust.

References

1. Salvador, E., Cavallaro, A., Ebrahimi, T.: Cast Shadow Segmentation Using Invariant Color Features. Computer Vision and Image Understanding 95(2), 238–259 (2004)
2. Levine, M.D., Bhattacharyya, J.: Removing Shadows. Pattern Recognition Letters 26(3), 251–265 (2005)
3. Finlayson, G.D., Hordley, S.D., Lu, C., Drew, M.S.: On the Removal of Shadows from Images. IEEE Trans. on PAMI 28(1), 59–68 (2006)
4. Finlayson, G.D., Hordley, S.D.: Color Constancy at a Pixel. Journal of the Optical Society of America 18(2), 253–264 (2001)
5. Fredembach, C., Finlayson, G.D.: Hamiltonian Path Based Shadow Removal. In: British Machine Vision Conference (2005)
6. Li, N., Xie, S.Y., Xie, Y.B.: Application of gray relational theory in moving object detection from video sequences. Journal of Computer-aided Design & Computer Graphics 21, 663–667 (2009) (in chinese)
7. Ma, M., Fan, Y.Y., Xie, S.Y.: A novel algorithm of image edge detection based on gray system theory. Journal of Image and Graphics 8, 1136–1139 (2003) (in chinese)
8. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man, and Cybernetics 9(1), 62–66 (1979)
9. Yuan, C.L., Xiong, Z.L., Zhou, X.H., Peng, X.H.: Study of Infrared Image Edge Detection Based on Sobel Operator. Laser & Infrared 39(1), 85–87 (2009)
10. Video Surveillance Online Repository, `http://imagelab.ing.unimore.it/visor/`

Moving Object Detecting System with Phase Discrepancy

Qiang Wang, Wenjun Zhu, and Liqing Zhang

Department of Computer Science and Engineering,
Shanghai Jiao Tong Uiniversity,
800 Dongchuan Road, Shanghai, China
{wq1018,wilsonzhu}@sjtu.edu.cn, zhang-lq@cs.sjtu.edu.cn

Abstract. This paper proposes an efficient moving object detecting system that detects moving objects in dynamic scene. The system consists of three parts: motion saliency calculation, moving area extraction and bounding box generation. We further analyze the the phase discrepancy algorithm and use it to get the motion saliency map from adjacent images. We use Canny-like salient area extraction algorithm to extract moving segments from motion saliency map. We then use graph based image segmentation algorithm to extend salient areas to bounding boxes. Computer simulations are given to demonstrate the high performance in detecting moving objects.

1 Introduction

Moving object detection in complex scenes is an important and challenging problem in computer vision. It is related to many applications such as object tracking, traffic monitoring and robotics. There are many approaches to solve the problem, such as background modeling, detection by recognition, view geometry and phase discrepancy.

The background substraction method is proposed under certain assumption, such as a stationary camera. The main idea is to learn the appearance model of the background [2] [3]. A moving object can not match the background well in the scene captured by a stationary camera. However, this kind of approaches don't work well in the scene captured by a moving camera.

Some other approaches come from object detection and recognition. The algorithm can detect moving object from particular categories, such as faces or pedestrians [4], with pre-trained detectors. The algorithms usually require offline training and can only handle particular object categories.

Detecting motion from camera geometry is another kind of approach. This method estimate the camera parameters under certain geometric constraints, use there parameters to compensate for camera induced motion, and separate the moving object from the residual motion in the scene [5].

Detecting moving objects in dynamic backgrounds using phase discrepancy is a novel method proposed by Zhou *et.al* [1]. The method is based on 1). the displacement of the foreground and the background can be represented by the

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 402–411, 2011.

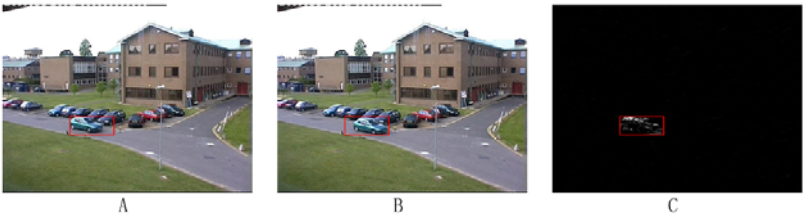


Fig. 1. An illustration of phase discrepancy with fast-moving object. A and B: Two adjacent frames. C: The motion saliency map of A and B. The red box is the bounding box of salient area.

phase change of Fourier spectra; 2). the motion of background objects can be extracted by phase discrepancy in an efficient and robust way. The algorithm does not rely on prior training on particular features or categories and can be implemented in 9 lines of MATLAB code. The algorithm is very efficient, it is a good choice for the implementation of an efficient moving object detection system.

However, it is not an easy task to use the algorithm directly in a moving object detection system for the following reasons.

First, the motion saliency map is generated using two images, the salient area covers the areas of moving object in both images. The bounding box of salient area is significantly larger than the object in each of the image if the displacement of the object is large. As illustrated in Fig.1, the bounding box covers the position of the moving car in both A and B, this leads to large False Alarm Rate(FAR).

Second, the saliency map is a pixel based representation, it favors moving parts of an object over the entire object [1]. Assume $I_{i-1}(\mathbf{a})$ and $I_i(\mathbf{a})$ are the intensity of a pixel at position $\mathbf{a}$ in two adjacent frames. If the difference $d_{i-1}(\mathbf{a}) = \|I_i(\mathbf{a}) - I_{i-1}(\mathbf{a})\|$ is small, the saliency value of position $\mathbf{a}$ is also small. Thus, the most salient areas of saliency map are the edges on moving objects and boundaries between moving objects and background. Salient areas cannot cover the whole moving object in two situations: 1). The moving object is composed of segments that have about the uniform intensity. As in Fig.2, many parts of the

Fig. 2. An illustration of phase discrepancy. A and B: Two adjacent frames. C: The motion saliency map of A and B using phase discrepancy.

right leg and upper body of the pedestrian are not salient. 2). The slow-moving parts of a moving object is not salient. As in Fig.2, the left leg of pedestrian is not salient though it is a part of the moving object.

From the discussion, we know that the extraction of moving objects from the saliency map of phase discrepancy is not a trivial problem. We describe our moving object detecting system in the following sections.

2 System Overview

Our moving object detecting system consists of three modules in general. They are saliency map calculation, salient area extraction, and image segmentation. As illustrated in Fig.3, the input of the system is two adjacent images. First, the phase discrepancy module generates one motion saliency map from the two adjacent input images. Next, Canny-like salient area extraction module extracts the salient areas from the saliency map. And then we use the result of image segmentation to connect different salient areas and generate the final bounding boxes of moving objects. The detail is discussed in Sect. 3 and 4.

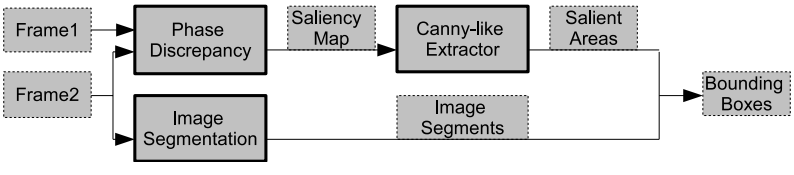


Fig. 3. Flowchart of the system

3 Canny-like Salient Area Extractor

Non-maximal suppression [6] is used in [1] to extract moving objects and generate bounding boxes from saliency maps. However, it is not suitable for the task. In this section, we first analyze the non-maximal suppression algorithm and then explain our Canny-like salient area extractor.

3.1 Flaw of Non-maximal Suppression

Non-maximal suppression is used in [1], it has three parameters θ_1, θ_2 and θ_3. θ_1 is the radius, θ_2 is the threshold for seeds, θ_3 is the binarizing threshold. All the parameters have to be tuned properly to get good results. The parameters was chosen using cross validation to minimize error in [1]. However, images are only available on the fly in a real system and the noise caused by background movement is indeterminable, it is difficult to get the proper parameters.

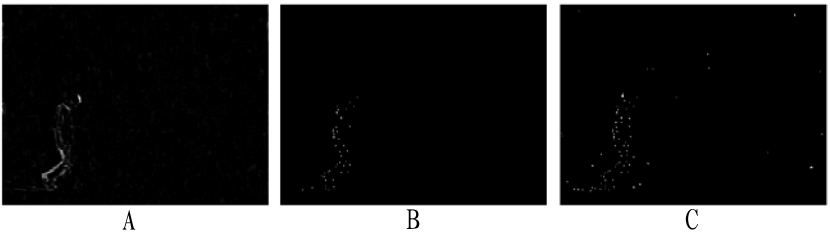


Fig. 4. Illustration of non-maximal suppression in detecting moving objects from motion saliency map. A: Motion saliency map of a pedestrian. B: Salient points generated by non-maximal suppression with large θ_2. C: Salient points generated by non-maximal suppression with small θ_2.

As illustrated in Fig.4, the result is strongly affected by the parameters. If θ_2 is set too large, salient points will be sparse and may result in several small bounding boxes that split the moving object into several parts. This also occurs when different parts of the object are moving at different speed, such as a pedestrian, because different parts of the moving object have different saliency values. If θ_2 is set too small, objects in background may be marked as moving objects.

We have analyzed the non-maximal suppression algorithm in extracting moving objects from motion saliency maps. Clearly, it is not the best choice for the task. So we designed the Canny-like salient area extractor, which performs better in a real system.

3.2 Canny-like Salient Area Extraction Algorithm

Canny edge detector is a state-of-art edge detector proposed by Canny [7]. The detector has four stages. The first stage is noise reduction. The detector uses a first derivative Gaussian filter to blur the input image and suppress the noise. The second stage is the finding of intensity gradient of the image. The detector use filters to detect horizontal, vertical and diagonal edges in the blurred image. The third stage is to do non-maximal suppression on the gradient magnitude to find the local maximal. The final stage is to trace the edges with hysteresis.

The detector has three parameters: θ_c, t_{low} and t_{high}. θ_c is the the standard deviation of the Gaussian filter used in the first stage. t_{high} is the high threshold used in hysteresis to mark out the points that are surely on edges. t_{low} is the low threshold used in hysteresis to trace the edges.

As discussed in Sect. 1, the boundaries related to moving objects have the strongest saliency values. They are similar to edges in the Canny edge detecting algorithm. The saliency values near the boundaries are smaller, but still larger than most parts of background. So we can trace from the points on the boundaries to other parts of moving objects using hysteresis. We adopt the thought of Canny edge detector and designed the Canny-like salient area extractor.

Our extractor has three stages. The first stage is noise reduction, like Canny edge detector, we use a first derivative Gaussian filter to blur the saliency map to suppress noise. The second stage is the finding of local maximal. Non-maximal suppression is used in this stage. Unlike Canny edge detector, our extractor do not need direction information, so we use the intensity of blurred saliency map directly instead of gradient magnitude. The final stage is to trace salient areas using hysteresis.

Like Canny edge detector, the extractor needs three parameters: θ_c, t_{high} and t_{low}. θ_c is the standard deviation of the Gaussian filter used in the first stage. t_{high} is the high threshold used in hysteresis to mark out the points that are surely on moving objects. t_{low} is the low threshold used in hysteresis to trace the salient areas.

We denote the number of local maximal points having saliency values below v by

$$H(v) = \sum_{v_i < v} N(v_i) \tag{1}$$

where $N(v_i)$ is the number of local maximal points that have saliency value v_i. We get the high threshold v_h through

$$\begin{cases} H(v_h) \geq t_{high} \times N_l \\ H(v_h - 1) < t_{high} \times N_l \end{cases} \tag{2}$$

where N_l is the number of all local maximal points. The local maximal points above or equal to v_h will be treated as points on the moving object, the collection of them is denoted by V_h. We then get the low threshold through

$$v_l = v_h \times t_{low} \tag{3}$$

The points having saliency value above or equal to v_l will be traced, the collection of them is denoted by V_l.

The pseudo code of Canny-like salient area extractor is shown below. The final `salient_segs` is a integral matrix with 0 denotes background, other values denote different salient areas respectively. The bounding box of each salient area can be generated simply from the matrix.

```
function salient_area_extractor(salieny_map, theta, thigh, tlow)
  salient_segs = zeros(saliency_map.size)
  seg_label = 0
  smoothed = gaussian_smooth(saliency_map, theta)
  local_max = non_maximal_suppression(smoothed)
  Vh = get_Vh() # Get Vh from (2)
  Vl = get_Vl() # Get Vl from (3)
  foreach vh in Vh:
    if salient_segs(vh.position) == 0:
      seg_label += 1
      salient_segs(vh.position) = seg_label
      follow(saliency_map, salient_segs, seg_label, Vl)
    end if
```

```
  end for
  return salient_segs
end function

function follow_segs(saliency_map, salient_segs, seg_label, vh, Vl)
  foreach v near vh:
    if v in Vl and salient_segs(v.position) == 0:
      salient_segs(v.position) = seg_label
      follow_segs(saliency_map, salient_segs, seg_label, v, Vl)
    end if
  end for
end function
```

4 Generation of Bounding Boxes

The Canny-like salient area extractor can extract many parts of salient area from motion saliency map. However, the saliency values of some parts on moving objects are about the same magnitude with background. It is hard to extract moving object as a whole with only saliency maps. We incorporate image segmentation algorithm and use bounding box intersection to enhance the performance of the system.

4.1 Connecting Salient Areas

As discussed in Sect. 1, the boundaries related to moving objects are likely to have the largest saliency values. The smooth parts of moving objects are likely to have smaller saliency values. It is possible that two salient areas are the different boundaries of the same segment on the moving object. Inspired by this, we use the result of image segmentation to connect different salient areas on the same object.

Image segmentation is the process of partitioning a digital image into multiple segments. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze [8]. There exists several general-purpose algorithms and techniques for image segmentation, such as clustering method, compression-based method [10], histogram-based method [8] [11] and graph partitioning method [9]. We choose the Efficient Graph-based Image Segmentation algorithm [9] in our system. For the algorithm is efficient while having good properties.

We define an salient area A_i *connect to* a segment S_m if there exists a point p that $p \in A_i$ and $p \in S_m$. We define two salient areas A_i and A_j are *connected* if they *connect to* the same segment S_m or they are *connected* with the same salient area A_k. A salient area can be *connected* with any number of other salient areas.

We merge all the *connected* salient areas and generate the bounding boxes for each merged area. Experiments show that our generated boxes cover the whole moving objects in most cases.

4.2 Refining Bounding Boxes

Although the bounding boxes generated using Canny-like salient area extractor and image segmentation cover the whole moving objects in most cases, they are not the best bounding boxes. As discussed in Sect. 1, the salient areas cover the moving objects in both the images, so the bounding boxes often cover larger area than moving objects and leads to large False Alarm Rate(FAR) with the criterion of [1].

Let's denote the ith input image by I_i, the saliency map generated from I_i and I_{i+1} by M_i, the salient area of M_i by A_i, the area of moving object in I_i by O_i. We have $O_i \subset A_{i-1}$ and $O_i \subset A_i$. If the moving direction of the object doesn't change or changes smoothly, then $(A_{i-1} - O_i) \cap (A_i - O_i)$ tends to be $\emptyset$, which suggests that $A_{i-1} \cap A_i \approx O_i$. So we intersect the bounding boxes generated from two adjacent motion saliency maps to produce more accurate bounding boxes for I_i.

This approach dose not add much computational burden, because M_i and A_i are used in the generation of bounding boxes of I_{i+1}.

5 Experiments

We introduce the experiments and results of our system in this section.

5.1 Experiment Setup

Phase discrepancy can detect moving objects when camera is moving. Many existing databases such as PETS [12] and CAVIAR [13] consider less about the situation. We choose the database and criterion introduced in [1]. The database consists of indoor/outfoor scenes. Different categories of objects are included in the video clip, such as walking pedestrians, cars and bicycles, and sport players. The criterion use Detection Rate(DR) and False Alarm Rate(FAR) to evaluate models. DR and FAR are defined by:

$$\mathrm{DR} = \frac{\sum_i \mathrm{TP}^i}{\sum_i \mathrm{GT}^i} \qquad \mathrm{FAR} = \frac{\sum_i \mathrm{FP}^i}{\sum_i \mathrm{TP}^i + \mathrm{FP}^i} \tag{4}$$

where $\mathrm{GT}^i, \mathrm{TP}^i, \mathrm{FP}^i$ denotes the number of ground truth, true positive, and false positive bounding boxes respectively. A detection is true positive if:

$$\frac{Area(R_{\mathrm{GT}} \cap R_D)}{Area(R_{\mathrm{GT}} \cup R_D)} \geq \mathrm{Th} \tag{5}$$

where R_{GT} and R_D denotes the ground truth and result generated by the algorithm respectively, Th is a threshold defines the tolerance of a system [1].

As discussed in Sect. 3, Canny-like salient area extractor has three parameters. θ_c affects the size of Gaussian filter, large θ_c causes more blurring. Small θ_c allow

extraction of small salient areas. Since noise of saliency map are often spines and salient areas are clusters that stand against smoothing, we use relatively large θ_c to reduce noise.

t_{high} decides the number of seeds to be selected. Since the amount of salient area is limited, we set t_{high} with a large value. Experiments show that the best t_{high} is between 0.9975 and 0.9995. We use 0.999 in the comparing with other methods.

t_{low} affects the size of each salient area. Small t_{low} leads to large area and False Alarm Rate(FAR), large t_{low} leads to small salient area and Detection Rate(DR). As discussed in Sect. 1, the saliency values of smooth parts on moving object are slightly large than the background, so we choose a relatively small t_{low}, 0.3 for all the experiments.

The image segmentation module of the system has two parameters, σ is the standard deviation of smooth filter, k is the constant to punish small segments [9]. Image segmentation is only used to connect different salient areas of the same object in the system. We find the parameters affect little about the final result through experiments, so we use the suggested parameters mentioned in [9].

5.2 Results

Fig.5 illustrates some results of our system. We can see that the bounding boxes generated by our system is very close to the ground truth bounding boxes of moving object.

We compare our system with the non-maximal suppression approach [1] with different Th. Table 1 is the result of comparison. Results show the DR of our system is about the as the non-maximal approach while FAR is smaller. When Th is large (greater than 0.4), our system performs much better. This suggests that our approach is more suitable for a real system, where large Th is needed. [1] have shown that other kinds of popular approaches perform poorly compared with non-maximal approach. So our system is also better than them in detecting moving objects in dynamic scene.

Despite of having many modules, our system is efficient. It performs 5 frames per second on a 2.2 GHz Core 2 Duo personal computer with image size 160×120.

Table 1. Comparison of different approaches

	Th	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Our System	DR	0.85	0.82	0.76	0.71	0.65	0.57	0.47	0.37	0.27	0.15	0.00
	FAR	0.03	0.05	0.12	0.18	0.25	0.33	0.42	0.54	0.68	0.83	1.00
Non-maximal Suppression Approach [1]	DR	0.83	0.82	0.80	0.75	0.63	0.46	0.20	0.07	0.02	0.00	0.00
	FAR	0.18	0.20	0.24	0.31	0.43	0.58	0.80	0.93	0.98	1.00	1.00

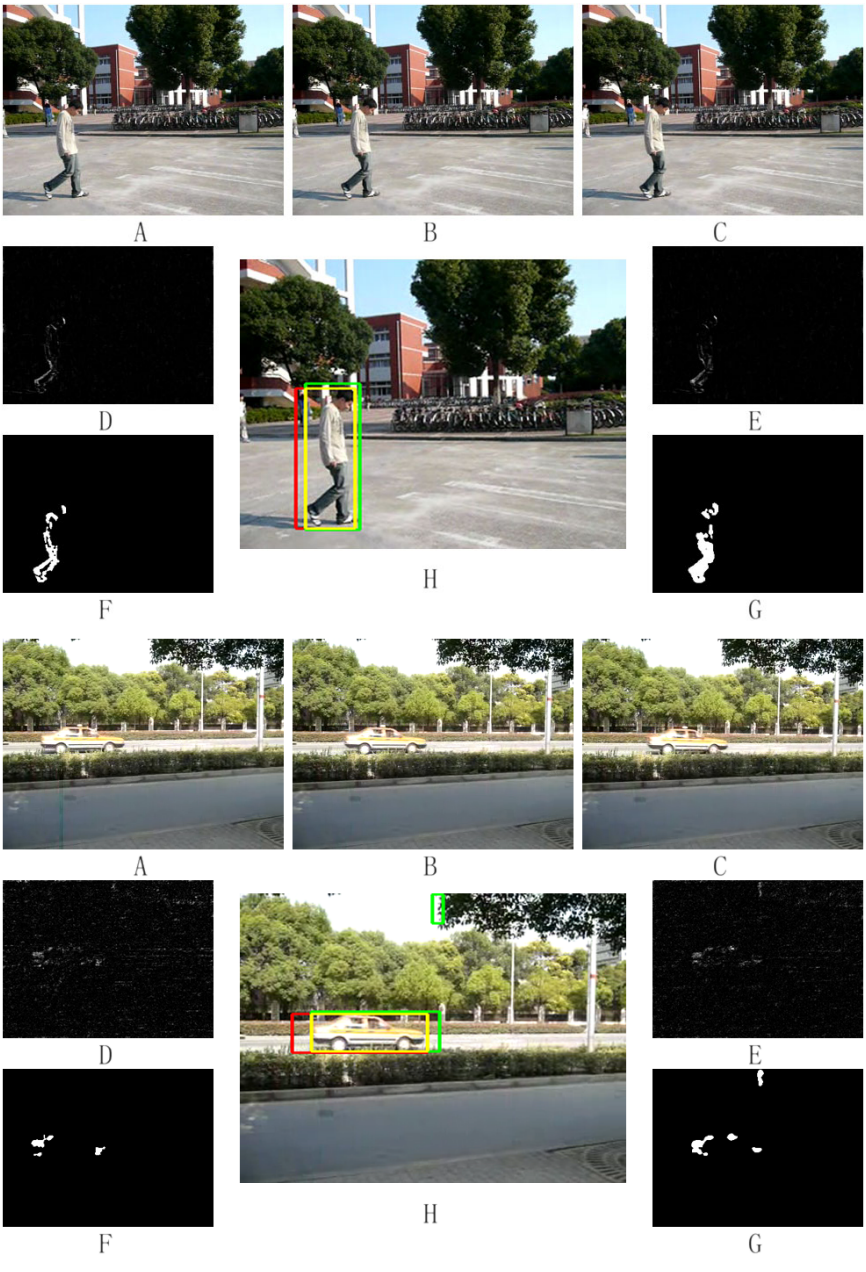


Fig. 5. Sample results of our system. A,B,C: Three adjacent frames. D: Motion saliency map of A and B. E: Motion saliency map of B and C. F: Salient areas of D. G: Salient areas of E. H: Moving object, red box is the bounding box of F, green box is the bounding box of G, yellow box is the final bounding box.

6 Conclusion

In this paper, we proposed an efficient moving object detecting system. The system use phase discrepancy to generate motion saliency map, Canny-like salient area extractor and image segmentation to generate bounding box, rectangular intersection to refine results. Experiments show that the system performs better than existing approaches in the detection of moving objects in dynamic scene.

Acknowledgement

The work was supported by the National Natural Science Foundation of China (Grant No. 90920014 and 60775007) and the Science & Technology Commission of Shanghai Municipality, China (Grant No. 08511501701).

References

1. Zhou, B., Hou, X., Zhang, L.: A phase discrepancy analysis of object motion. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010, Part III. LNCS, vol. 6494, pp. 225–238. Springer, Heidelberg (2011)
2. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, vol. 2, pp. 246–252 (1999)
3. Mittal, A., Paragos, N.: Motion-based background substraction using adaptive kernal density estimation. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, vol. 2 (2004)
4. Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrain detection: A benchmark. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 304–311 (2009)
5. Han, M., Kanade, T.: Reconstruction of a scene with multiple linearly moving objects. International Journal of Computer Vision 59, 285–300 (2004)
6. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis, and Machine Vision. Cengage-Engineering (2007)
7. Canny, J.: A Conputational approach to edge detection. Pattern Analysis and Machine Intelligence 8(6), 679–698 (1986)
8. Linda, G., George, C.: Computer Vision, pp. 279–325. Prentice-Hall, Englewood Cliffs (2001)
9. Felzenszwalb, P., Huttenlocher, D.: Efficient graph-based image segmentation. International Journal of Computer Vision 59(2), 167–181 (2004)
10. Rao, S.R., Mobahi, H., Yang, A.Y., Sastry, S.S., Ma, Y.: Natural image segmentation with adaptive texture and boundary encoding. In: Zha, H., Taniguchi, R.-i., Maybank, S. (eds.) ACCV 2009. LNCS, vol. 5994, pp. 135–146. Springer, Heidelberg (2010)
11. Ohlander, R., Price, K., Reddy, D.: Picture segmentation using as recursive region splitting method. Computer Graphics and Image Processing 8, 313–333 (1978)
12. http://ftp.pets.rdg.ac.uk
13. http://homepages.inf.ed.ac.uk/rbf/caviar/

Automation of Virtual Interview System Using the Gesture Recognition of Particle Filter

Yang Weon Lee

Department of Information and Communication Engineering, Honam University, Seobongdong, Gwangsangu, Gwangju, 506-714, South Korea

Abstract. This paper describes a gesture recognition algorithm based on the particle filters for automation of virtual interview system. The particle filter is well operated for human gesture recognition than any other recognition algorithm. Through the experiments, we show that the proposed scheme is stable and works well in virtual interview system's environments.

1 Introduction

In this paper, we focused into the development of hand gesture recognition using particle filter. It is applied for virtual interview system automation. Particle filter[1] is based on the Bayesian conditional probability such as *prior* distribution and *posterior* distribution. First of all, we expanded the existing algorithm[2] to derive the CONDENSATION-based particle filter for hand gesture recognition. Also, we adopt the two hand motion model to confirm the algorithm performance such as leftover and paddle. The overall scheme for the gesture recognition system is shown in Figure1.

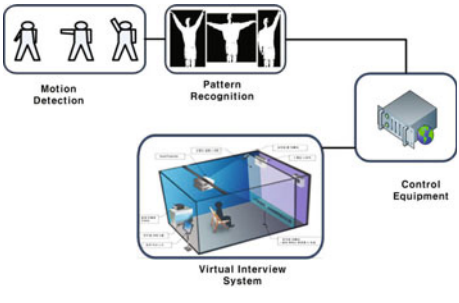


Fig. 1. Overall operation block diagram of recognition system

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 412–419, 2011.

2 CONDENSATION Algorithm

2.1 CONDENSATION Algorithm

The particle filter approach to track motion, also known as the condensation algorithm [1] and Monte Carlo localisation , uses a large number of particles to explore the state space. Each particle represents a hypothesised target location in state space. Initially the particles are uniformly randomly distributed across the state space, and each subsequent frame the algorithm cycles through the steps illustrated in Figure 2:

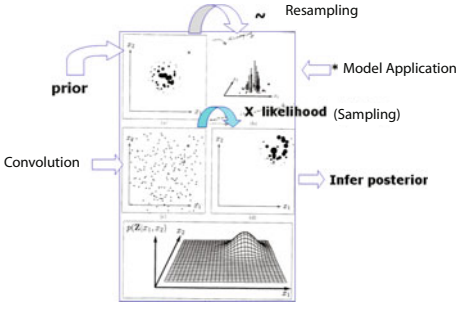


Fig. 2. Process of particle filter calculation

1. Deterministic drift: particles are moved according to a deterministic motion model (a damped constant velocity motion model was used).
2. Update probability density function (PDF): Determine the probability for every new particle location.
3. Resample particles: 90with replacement, such that the probability of choosing a particular sample is equal to the PDF at that point; the remaining 10throughout the state space.
4. Diffuse particles: particles are moved a small distance in state space under Brownian motion.

This results in particles congregating in regions of high probability and dispersing from other regions, thus the particle density indicates the most likely target states. See [3] for a comprehensive discussion of this method. The key strengths of the particle filter approach to localisation and tracking are its scalability (computational requirement varies linearly with the number of particles), and its ability to deal with multiple hypotheses (and thus more readily recover from tracking errors). However, the particle filter was applied here for several additional reasons:

– it provides an efficient means of searching for a target in a multi-dimensional state space.

– reduces the search problem to a verification problem, ie. is a given hypothesis face-like according to the sensor information?
– allows fusion of cues running at different frequencies.

2.2 Application of CONDENSATION for the Gesture Recognition

In order to apply the Condensation Algorithm to gesture recognition, we extend the methods described by Black and Jepson [2]. Specifically, a state at time t is described as a parameter vector: $s_t = (\mu, \phi^i, \alpha^i, \rho^i)$ where: μ is the integer index of the predictive model, ϕ^i indicates the current position in the model, α^i refers to an amplitudal scaling factor and ρ^i is a scale factor in the time dimension. Note that i indicates which hand's motion trajectory this ϕ^*, α^*, or ρ^* refers to left and right hand where $i \in \{l, r\}$. My models contain data about the motion trajectory of both the left hand and the right hand; by allowing two sets of parameters, I allow the motion trajectory of the left hand to be scaled and shifted separately from the motion trajectory of the right hand (so, for example,ϕ^l refers to the current position in the model for the left hand's trajectory, while ϕ^r refers to the position in the model for the right hand's trajectory). In summary, there are 7 parameters that describe each state.

Initialization. The sample set is initialized with N samples distributed over possible starting states and each assigned a weight of $\frac{1}{N}$. Specifically, the initial parameters are picked uniformly according to:

$$\begin{aligned} \mu &\in [1, \mu_{max}] \\ \phi^i &= \frac{1-\sqrt{y}}{\sqrt{y}}, y \in [0, 1] \\ \alpha^i &= [\alpha_{min}, \alpha_{max}] \\ \rho^i &\in [\rho_{min}, \rho_{max}] \end{aligned} \tag{1}$$

Prediction. In the prediction step, each parameter of a randomly sampled s_t is used to s_{t+1}determine based on the parameters of that particular s_t . Each old state,s_t , is randomly chosen from the sample set, based on the weight of each sample. That is, the weight of each sample determines the probability of its being chosen. This is done efficiently by creating a cumulative probability table, choosing a uniform random number on $[0, 1]$, and then using binary search to pull out a sample (see Isard and Blake for details[1]). The following equations are used to choose the new state :

$$\begin{aligned} \mu_{t+1} &= \mu_t \\ \phi^i_{t+1} &= \phi^i_t + \rho^i_t + N(\sigma_\phi) \\ \alpha^i_{t+1} &= \alpha^i_t + N(\sigma_\alpha) \\ \rho_{t+1} &= \rho^i_t + N(\sigma_\rho) \end{aligned} \tag{2}$$

where $N(\sigma_*)$ refers to a number chosen randomly according to the normal distribution with standard deviation σ_* . This adds an element of uncertainty to each prediction, which keeps the sample set diffuse enough to deal with noisy data. For a given drawn sample, predictions are generated until all of the parameters are within the accepted range. If, after, a set number of attempts it is still impossible to generate a valid prediction, a new sample is created according to the initialization procedure above.

Updating. After the Prediction step above, there exists a new set of N predicted samples which need to be assigned weights. The weight of each sample is a measure of its likelihood given the observed data $Z_t = (z_t, z_{t_1}, \cdots)$. We define $Z_{t,i} = (z_{t,i}, z_{(t-1),i}, \cdots)$ as a sequence of observations for the ith coefficient over time; specifically, let $Z_{(t,1)}, Z_{(t,2)}, Z_{(t,3)}.Z_{(t,4)}$ be the sequence of observations of the horizontal velocity of the left hand, the vertical velocity of the left hand, the horizontal velocity of the right hand, and the vertical velocity of the right hand respectively. Extending Black and Jepson [2], we then calculate the weight by the following equation:

$$p(z_t|s_t) = \prod_{i=1}^{4} p(Z_{t,i}|s_t) \tag{3}$$

where $p(z_{t,i}|s_t) = \frac{1}{\sqrt{2\pi}} exp \frac{-\sum_{j=0}^{\omega-1}(z_{(t-j),i} - \alpha^* m^{\mu}_{(\phi-\rho^* j),i})^2}{2(\omega-1)}$ and where ω is the size of a temporal window that spans back in time. Note that ϕ^*, α^* and ρ^* refer to the appropriate parameters of the model for the blob in question and that $\alpha^* m^{(\mu)}_{(\phi^*-\rho^* j),i}$ refers to the value given to the ith coefficient of the model μ interpolated at time $\phi^* - \rho^* j$ and scaled by α^* .

3 Gesture Model and Image Preprocessing

We adopt the two gesture model to verify the proposed particle filter. As shown in Figure 3, gesture 1 means leftover and gesture 2 means paddle.

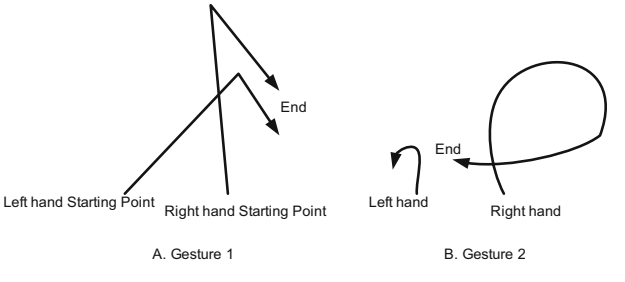


Fig. 3. Two gesture model

3.1 Raw Image Preprocessing

The image sequences were filmed using a Sony DCR Camcoder. They were manually aligned and then converted into sequences of TIFs to be processed in MATLAB. Each TIF was 320x240 pixels, 24bit color. The lighting and background in each sequence is held constant; the background is not cluttered. The focus of my project was not to solve the tracking problem, hence I wanted the hands to be relatively easy to track. I collected 7 film sequences of each sign (see Figure 4.)

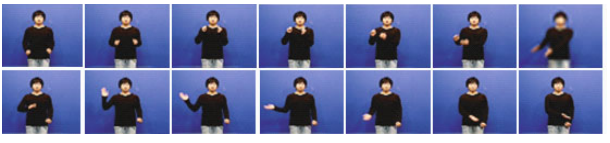


Fig. 4. Gesture Images of the Two Models

3.2 Skin Extraction

In order to segment out skin-colored pixels, we used the color segment routine we developed in MATLAB. Every image in every each sequence was divided into the following regions: skin, background, clothes, and outliers. First of all, we set up the mask using the gaussian distribution based on mean and covariance value which is stored in the database. Then we segment the images into four section above mentioned regions. So, we get the the segment of skin as shown in Figure 5.

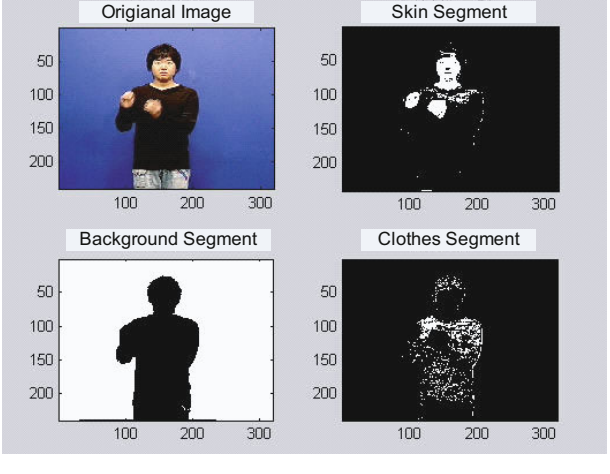


Fig. 5. Output of Segmentation

3.3 Finding Skin-Colored Blobs

We then calculated the centroid of the three largest skin colored 'blobs' in each image. Blobs were calculated by processing the skin pixel mask generated in the previous step. A blob is defined to be a connected region of 1's in the mask. Finding blobs turned out to be a bit more difficult than we had originally thought. Our first implementation was a straightforward recursive algorithm which scans the top down from left to right until it comes across a skin pixel which has yet to be assigned to a blob. It then recursively checks each of that pixel's neighbors to see if they too are skin pixels. If they are, it assigns them to the same blob and recurses. On such large images, this quickly led to stack overflow and huge inefficiency in MATLAB.

3.4 Calculating the Blobs' Motion Trajectories over Time

At this point, tracking the trajectories of the blobs over time was fairly simple. For a given video sequence, we made a list of the position of the centroid for each of the 3 largest blobs in each frame. Then, we examined the first frame in the sequence and determined which centroid was farthest to the left and which was farthest to the right. The one on the left corresponds to the right hand of signer, the one to the right corresponds to the left hand of the signer. Then, for each successive frame, we simply determined which centroid was closest to each of the previous left centroid and called this the new left centroid; we did the same for the blob on the right. Once the two blobs were labelled, we calculated the horizontal and vertical velocity of both blobs across the two frames using [(change in position)/time]. We recorded these values for each sequential frame pair in the sequence. The example of the tracking is shown in Figure 6

Fig. 6. Tracking result using centroid calculation

3.5 Creating the Motion Models

We then created models of the hand motions involved in each sign. Specifically, for each frame in the sign, we used 5 training instances to calculate the average horizontal and vertical velocities of both hands in that particular frame. The following graphs show the models derived for both signs (see Figure 7:)

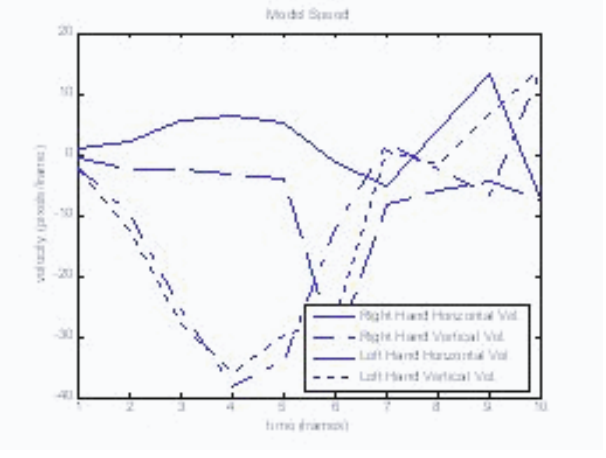


Fig. 7. Velocity of Model I

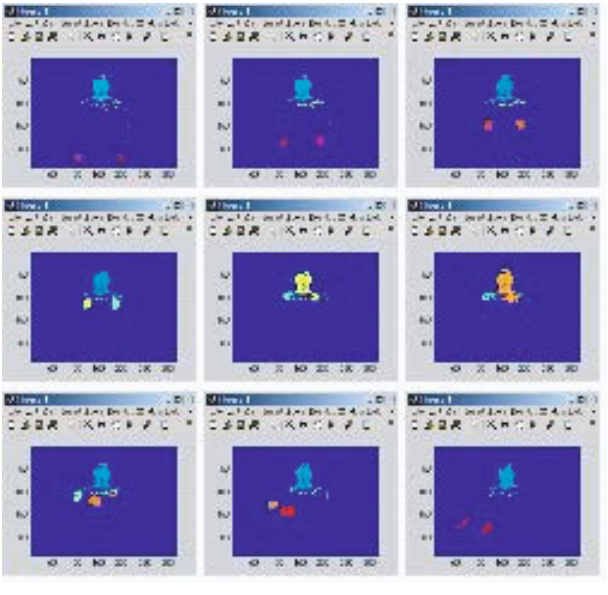


Fig. 8. The Tracking process of particle filter for the model 1(From left to right, top to down)

4 Experiment Result

To test the proposed particle filter scheme, we used two gesture model which is shown in Figure 3 in this paper. The coefficient of particle filter are $\mu_{max} = 2, \alpha_{min} = 0.5, \alpha_{max} = 1.5, \rho_{min} = 0.5, \rho_{max} = 1.5$ to maintain the 50Also, the other parameters are settled by $\sigma_\phi = \sigma_\alpha = \sigma_\rho = 0.1$. The variable of ω equation 3 is 10.

5 Conclusions

In this paper, we have developed the particle filter for the gesture recognition. This scheme is important in providing a computationally feasible alternative to classify the gesture in real time. We have proved that given an image, particle filter scheme classify the gesture in real time.

Acknowledgements

This work was supported by the Ministry of Knowledge and Economy Grant funded by the Korean Government(MKE) (KRF-2010-D00356).

References

1. Isard, M., Blake, A.: CONDENSATION-conditional density propagation for visual tracking. International Journal of Computer Vision 29(1), 5–28 (1998)
2. Black, M.J., Jepson, A.D.: A Probabilistic Framework for Matching Temporal Trajectories: Condensation-based Recognition of Gestures and Expressions. In: Burkhardt, H.-J., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1406, pp. 909–924. Springer, Heidelberg (1998)
3. Isard, M., Blake, A.: A mixed-state condensation tracker with automatic model-switching. In: Proceedings 6th International Conference of Computer Vision, pp. 107–112 (1998)
4. Lee, Y.-W.: Adaptive Data Association for Multi-target Tracking using relaxation. In: Huang, D.-S., Zhang, X.-P., Huang, G.-B. (eds.) ICIC 2005. LNCS, vol. 3644, pp. 552–561. Springer, Heidelberg (2005)
5. Lee, Y.W., Seo, J.H., Lee, J.G.: A Study on the TWS Tracking Filter for Multi-Target Tracking. Journal of KIEE 41(4), 411–421 (2004)

Performance Analysis of Improved Affinity Propagation Algorithm for Image Semantic Annotation

Dong Yang and Ping Guo*

Image Processing and Pattern Recognition Laboratory,
Beijing Normal University, Beijing 100875, China
{d.yang,pguo}@ieee.org

Abstract. In an image semantic annotation system, it often encounters the large-scale and high dimensional feature datasets problem, which leads to a slow learning process and degrading image semantic annotation accuracy. In order to reduce the high time complexity caused by redundancy information of image feature dataset, we adopt an improved affinity propagation (AP) algorithm to improve annotation by extracting and re-grouping the repeated feature points. The time consumption is reduced by square of repetition factor. The experiments results illustrate that the proposed annotation method has excellent time complexity and better annotation precision compared with original AP algorithms.

Keywords: Information retrieval, Image semantic annotation, Clustering, Affinity propagation.

1 Introduction

For semantic annotation of natural images or human-activity images, in order to describe complex and elaborated image semantics, as many as possible images are required in most existing annotation systems. Because of both the image redundancy and image features redundancy, such as overlapping sampling and repeated emergence of similar image regions, it is very common there are a lot of repeated or near same image feature samples before applying vector quantization (VQ) techniques [15] to optimize training data in large datasets.

For unsupervised annotation, each semantic label is considered as a variable. The annotation is treated as a joint probability modeling problem, with clustering representations of image features or words, such as N-cut based method [10] and cross media relevance models [11]. For supervised annotation, each semantic label is considered as a class. The annotation is regarded as a classification problem, which adopts clustering to get sparse representation or posterior probabilistic distribution of image features for each class, and some supervised learning techniques such as the divide and conquer strategy-based scene classifier [12], supervised multi-class labeling algorithm [13], and multi-class annotation using optimized training data [4] are applied to annotate the images.

* Corresponding author.

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 420–428, 2011.

In both unsupervised and supervised annotation systems, clustering has become an important process to handle the huge number feature samples. Affinity propagation (AP) clustering algorithm has been validated powerful for image categorization and annotation [1][2][4], because of its excellent performance, such as automatically determining cluster number, and using similarity of data pairs instead of data values. Recently, weighted AP (WAP) [5][7] and AP with repeated points (APRP) [8] are developed to process large dataset with repeated points.

Time consumption problem for large dataset has been studied, including following three aspects: (1) introducing prior knowledge such as sparse similarity matrix [6]; (2) divide-and-conquer strategy such as hierarchical method [7], partition method[9], and sampling techniques [3]; (3) vector quantization techniques [14], which is powerful especially for large and high-dimensioned data.

In this paper, we study the problem to reduce time complexity based on vector quantization and APRP algorithm and analyze the performance of the algorithm for supervised image annotation.

2 Image Annotation Using APRP

Image semantic annotation can be regarded as a multi-class classification problem, which maps image features to semantic class labels, through the procedures of image modeling and image-semantic classification [4]. For image modeling, APRP is adopted instead of original AP (OAP) and WAP to process image feature dataset with repeated points. The algorithm flowchart is illustrated as in Fig. 1.

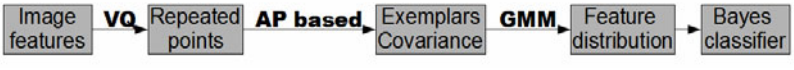


Fig. 1. The algorithm flowchart

2.1 OAP, WAP and APRP

Brief reviews of OAP, WAP and APRP are given according to the literatures [1], [7] and [8], respectively. Repeated point is usually represented as $(\mathbf{x}_i, n_i)$, with datum $\mathbf{x}_i$ and its repetition factor n_i [7].

AP algorithm is a graph-based message-passing clustering algorithm. Each data vector is viewed as a point in the graph, and real-value messages are recursively transmitted along edges of the graph until a relatively small number of exemplars and corresponding clusters emerge. The similarity $s(i, k)$ is the negative distance square between datum i and j, and the self similarity $s(k, k)$ is called preference.

The responsibility $r(i, k)$, which is sent from datum i to potential exemplar k, reflects the accumulated evidence for how appropriate datum k is the exemplar of datum i, considering other potential exemplars of datum i. The availability $a(i, k)$, which is sent from datum i to potential exemplar k, reflects the accumulated evidence for how appropriate it would be for datum i to choose datum k as its exemplar, considering the support from other data that datum k should be an exemplar.

The exemplar is determined by combing the availability and responsibility.

$$e = \max_{k} \{a(i,k) + s(i,k)\}. \tag{1}$$

The main differences of OAP, WAP and APRP are at the message-passing, similarity and preference setting. These differences lead to that WAP and APRP treats repeated points as one point thus to reduce the time complexity, while OAP does not consider the repeated point problem.

Frey and Dueck [1] set the initial $a(i, k)$ as 0, then the $r(i, k)$ and a(i, k) are iteratively updated:

$$r(i,k) \leftarrow s(i,k) - \max_{k' \text{ s.t. } k' \neq k} \{a(i,k') + s(i,k')\}. \tag{2}$$

$$a(i,k) \leftarrow \min \left\{ 0, r(k,k) + \sum_{i' \text{ s.t. } i' \notin \{i,k\}} \max \{0, r(i',k)\} \right\}. \tag{3}$$

$$a(k,k) \leftarrow \sum_{i' \text{ s.t. } i' \neq k} \max \{0, r(i',k)\}. \tag{4}$$

OAP does not refer to the repetition factor n_i of the datum i, therefore there should be n_i copies of $\mathbf{x}_i$ in the dataset for OAP.

Zhang *et al*. [7] changed similarity and preference as:

$$s'(i,j) = n_i s(i,j), \quad i \neq j. \tag{5}$$

$$s'(i,i) = s(i,i) + (n_i - 1)\varepsilon_i, \quad \varepsilon_i \geq 0. \tag{6}$$

The formula (5) expresses the repetition factor n_i of the datum i by changing the similarities from datum $\mathbf{x}_i$ to all other data. However, if n_i grows large, i tends to be dissimilar with all other data, which makes $\mathbf{x}_i$ tends to be an isolated point.

Yang and Guo [8] set the preference similar to formula (6) and change the objective function as:

$$S(c) = \sum_{i=1}^{N} n_i s(i, c_i) + \sum_{k=1}^{N} \delta_k(c). \tag{7}$$

Where c is the clustering result that maps datum i to its exemplar, $\delta(c)$ is a function that will produce a large value if an exemplar does not choose itself as its exemplar, and $S(c)$ is the summation of the similarity of all data and their exemplars, adding the penalty factor $\delta(c)$.

The update rule of availability is changed as:

$$a(i,k) \leftarrow \min \left\{ 0, r(k,k) + \sum_{i' \text{ s.t. } i' \notin \{i,k\}} \max \{0, n_{i'} r(i',k)\} \right\}. \tag{8}$$

$$a(k,k) \leftarrow \sum_{i' \text{ s.t. } i' \notin \{i,k\}} \max \{0, n_{i'} r(i',k)\}. \tag{9}$$

The formula (8) and (9) considers the repetition factor n_i because of the summation operation. APRP does not change the similarity because the similarity is a distance-like measure, which is not influenced by repetition factor.

2.2 Applying APRP to Large Dataset

Each image is divided into sub-blocks of 8×8 pixels with overlapping margin. For each sub-block, discrete cosine transform is applied on each color channel, the coefficient values are quantized and zigzag scanned from left top to right down. The first 16 values are selected as the feature vector of this color channel. Then the three vectors from three color channels are concatenated as a 48-dimensional feature vector of the sub-block. Vector quantization algorithm [15] is adopted on the vector group, while the codebooks are trained once in the whole dataset and are fixed in each quantization process.

The training set of each class label is composed with feature points from training images that has this label. APRP algorithm is adopted to get exemplars with automatically determined cluster number for each training set. The image feature distribution of each class label is estimated in the form of Gaussian mixture model (GMM) [4].

$$\{\mathbf{e}_k\} = \text{aprp}(\{\mathbf{x}\}),\ P(\mathbf{x}|\boldsymbol{\mu},\Sigma,\pi) = \sum_{k=1}^{K} \pi_k N(\mathbf{x}|\boldsymbol{\mu}_k,\Sigma_k), \tag{10}$$

where $\{\mathbf{x}\}$ is the set of 48-dimensional feature vectors. Each feature vector is assigned an exemplar by APRP algorithm, which means all the feature vectors are clustered into K clusters. Each cluster $\{\mathbf{x}_k\}$ has one exemplar $\boldsymbol{\mu}_k=\mathbf{e}_k$ as the mean vector. The covariance matrix is $\boldsymbol{\Sigma}_k=\text{cov}(\{\mathbf{x}_k\})$ and cluster weight is $\pi_k=\text{num}(\{\mathbf{x}_k\})$.

2.3 Image Semantic Annotation

Bayesian classifier is adopted for image semantic annotation. The classifier is trained using training images which are manually pre-annotated with single class label.

For each class label c, the re-grouping process collects repeated points from every image of this label c and assign them into groups. Then APRP algorithm is applied on these groups, and the class distribution $P(\mathrm{x}|c,\boldsymbol{\mu}_c,\boldsymbol{\Sigma}_c,\pi_c)$ is computed hierarchically using the GMM image modeling method.

For annotating a test image $\mathbf{I}$, the Bayesian decision rule is adopted. For a given class c, the probability that the test image belongs to this class is the product of the probabilities that the image feature samples $\mathbf{x}$ belong to this class.

$$P(\mathbf{I} \mid c,\boldsymbol{\mu}_c,\Sigma_c,\pi_c) = \prod_{\mathrm{x}\in \mathrm{I}} P(\mathbf{x} \mid c,\boldsymbol{\mu}_c,\Sigma_c,\pi_c)\cdot \tag{11}$$

By computing all class-conditional distributions $P(\mathbf{I}|c)$, the semantic annotation results for this image $\mathbf{I}$ can be obtained with the labels whose posterior probabilities $P(c|\mathbf{I})$ are the top several large values [4].

2.4 Performance Analysis of APRP

Image annotation system usually processes many similar image regions, which increase the probability of forming repeated points after vector quantization of a large number of feature points. On consider this fact, it is expected that APRP-based annotation algorithm works well with the dataset with repeated points.

Assuming the number of feature points is M and the number of repeated points is N, we count the repeated points as one point. N is less than the codebook length of vector quantization. The relationship between M and N is:

$$M = \sum_{i=1}^{N} n_i \cdot \tag{12}$$

Where n_i is the repetition factor of point i, M/N is the repetition factor of the whole group of feature points.

In figure 2, the influence of repetition factor on clustering is illustrated. The distance criterion is the sum of similarities from data to their exemplars, and the sum of similarities from exemplars to their center, which is used to prevents that the cluster number becoming too large.

$$f_{\text{distance}} = \sum_{i=1}^{N} \left| \mathbf{x}_i - \mathbf{m}_i \right| + \sum_{i=1}^{S} \left| \mathbf{m}_i - \overline{\mathbf{m}} \right| \cdot \tag{13}$$

When the M/N increases from 1.67 times to 3.00 times, the increase of similarity sum of WAP increase almost one time. The APRP performs the best among three algorithms, and when repetition factor is more than 2.00, the performance of APRP is stable compared with WAP and OAP.

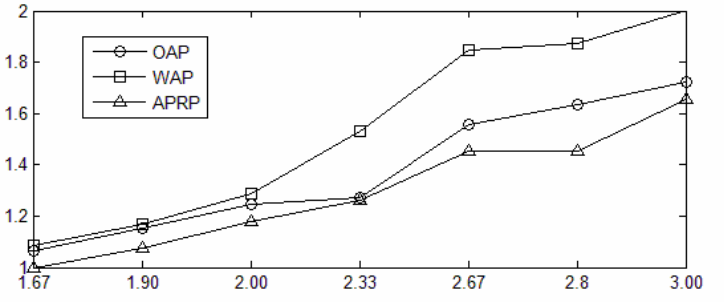


Fig. 2. The repetition factor vs. corresponding distance criterion of clustering

APRP algorithm greatly reduces the time consumption. AP algorithm repeatedly processes every copy of the repeated points. AP algorithm has the loop of comparing each point with other points, and its time consumption is M^2/N^2 times of APRP's time consumption.

3 Experiments

The annotation of natural images has become a difficult problem but a valuable benchmark to validate the image annotation algorithm. The performance of the proposed algorithm is evaluated in both the image modeling and annotation stages. The different image annotation method is compared with adopting the following clustering algorithms separately: OAP[1], WAP[7], APRP[8].

3.1 Dataset and Criteria

The images are selected from databases [18] and [19]. We select a subset of 2000 images which contain seven labels: building, car, crossroad, grass, plant, road and sky. There are 628 single-label images (we name it DBS), and 1372 multiple-label images (we name it DBM). One multiple-label image usually contains two or three labels. Using these images, we set up two training and testing scenarios, as in table 1.

Table 1. Two training and testing scenarios

	Training	Testing
Scenario1	2/3 DBS	1/3 DBS
Scenario2	DBS	DBM

Each image is divided into some sub-blocks with size of 8×8 pixels, and the adjacent blocks overlap 2 pixels. We resize the image with the resolution 300~500×300~500. A fixed code-vector number of 1000 is set up for vector quantization of each image.

The image modeling criteria are sum of point-to-centroid distance [17], and the logarithm of likelihood [16], which are shown in formula (13) and (14), respectively.

$$f_{\text{likelihood}} = \log P\left(\mathbf{I}|\boldsymbol{\mu}, \Sigma, \pi\right) = \sum_{\mathbf{x}\in \mathbf{I}}^{N} \log P\left(\mathbf{x}|\boldsymbol{\mu}, \Sigma, \pi\right). \tag{14}$$

The image annotation criteria are average recall and precise. For a given semantic class, we assume that there are w_h human annotated images w_{auto} computer annotated images in the test set, of which w_c are correct, the recall and precision are defined as following:

$$recall = \frac{w_c}{w_h}, \quad precise = \frac{w_c}{w_{auto}}. \tag{15}$$

3.2 Experiment Result Analysis

In table 2, we compare the time consumption of annotating one image, OAP-based method is several times of that of WAP or APRP. The total annotation time in table 2 contains clustering time. The time consumption is reduced by M^2/N^2 times. For example, if the repetition factor M/N is 3, the time consumption of OAP is nearly 9 times of that of WAP or APRP.

Table 2. Time consumption of annotating one image

Time (s)	OAP	WAP	APRP
DBS	331.0	54.3	56.3
DBM	487.9	77.2	78.6

In table 3 and table 4, we compare the image modeling results of the three annotation systems, where the APRP-based method performs similarly even a little bit better than OAP, and much better than WAP.

Table 3. Sum of point-to-centroid distance of image modelling in DBS and DBM

f_{distance}	OAP	WAP	APRP
DBS	24872	33515	24362
DBM	42238	48993	42626

Table 4. Logorithm of likelihood of image modelling in DBS and DBM

$f_{\text{likelihood}}$	OAP	WAP	APRP
DBS	-48.4	-56.6	-47.7
DBM	-52.1	-73.6	-56.1

In table 5, we compare the image annotation result of the three methods. In scenario 1, we use single-label testing images to simulate image categorization tasks; in scenario 2, we use multiple-labels testing images to simulate image annotation tasks.

Table 5. Image semantic annotation result

Annotation	Scenario 1			Scenario 2		
	OAP	WAP	APRP	OAP	WAP	APRP
Recall	85.7	75.3	88.5	74.5	67.8	75.2
Precise	73.5	69.4	75.0	61.2	59.1	60.9

The proposed APRP algorithm improves the accuracy of image annotation than WAP does, and it performs close to even higher than OAP. Considering the time consumption of OAP is several times of that of APRP, the proposed algorithm performs the best among the three algorithms for image annotation on large dataset.

4 Conclusion

On considering there exists redundancy information in image feature dataset, the performance of improved AP algorithm for image semantic annotation is analyzed in this paper. The redundancy information usually appears to be repeated points in the large feature datasets after vector quantization, we propose to adopt improved AP

algorithm to solve this problem. The image modeling accuracy is improved and time consumption in annotation is greatly reduced with APRP algorithm. The annotation precision approaches or even outperforms than the original AP algorithm, and is much better than that of WAP algorithm. For the case of repetition factor increases because of vector quantization, the performance of APRP algorithm is more stable compared with OAP and WAP. The proposed algorithm is promising on the effectiveness and response speed of the image semantic annotation.

Acknowledgments. The research work described in this paper was fully supported by the grants from the National Natural Science Foundation of China, (Project No. 90820010, 60911130513)

References

1. Frey, B.J., Dueck, D.: Clustering by Passing Messages between Data Points. Science 315, 972–976 (2007)
2. Dueck, D., Frey, B.J.: Non-metric Affinity Propagation for Unsupervised Image Categorization. In: IEEE International Conf. on Computer Vision, pp. 1–8. IEEE Press, New York (2007)
3. Sun, C.Y., Wang, C.H., Song, S., Wang, Y.F.: A Local Approach of Adaptive Affinity Propagation Clustering for Large Scale Data. In: IEEE International Joint Conf. on Neural Networks, pp. 161–165. IEEE Press, New York (2009)
4. Yang, D., Guo, P.: Image Modeling with Combined Optimization Techniques for Image Semantic Annotation. Neural Comput. Appl. (2011) (in press)
5. Furtlehner, C., Sebag, M., Zhang, X.L.: Scaling Analysis of Affinity Propagation. Phys. Rev. E 81(6), 006102 (2010)
6. Xiao, J.X., Wang, J.D., Tan, P., Quan, L.: Joint Affinity Propagation for Multiple View Segmentation. In: IEEE International Conf. on Computer Vision, pp. 1–7. IEEE Press, New York (2007)
7. Zhang, X., Furtlehner, C., Sebag, M.: Data Streaming with Affinity Propagation. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part II. LNCS (LNAI), vol. 5212, pp. 628–643. Springer, Heidelberg (2008)
8. Yang D., Guo P.: Improvement of Affinity Propagation Algorithm for Large Dataset. In: Workshop of the Cognitive Computing of Human Visual and Auditory Information (2010) (in Chinese)
9. Zhang, X.Q., Wu, F., Zhuang, Y.T.: Clustering by Evidence Accumulation on Affinity Propagation. In: IEEE International Conf. on Pattern Recognition, pp. 1–4. IEEE Press, New York (2008)
10. Barnard, K., Duygulu, P., Forsyth, D., de Freitas, N., Blei, D.M., Jordan, M.I.: Matching Words and Pictures. J. Mach. Learn. Res. 3, 1107–1135 (2003)
11. Jeon, J., Lavrenko, V., Manmatha, R.: Automatic Image Annotation and Retrieval using Cross-Media Relevance Models. In: ACM International Conf. on Research and Development in Information Retrieval, pp. 119–126. ACM Press, New York (2003)
12. Luo, J., Savakis, A.: Indoor VS Outdoor Classification of Consumer Photographs using Low-Level and Semantic Features. In: IEEE International Conf. on Image Processing, pp. 745–748. IEEE Press, New York (2001)

13. Carneiro, G., Chan, A.B., Moreno, P.J., Vasconcelos, N.: Supervised Learning of Semantic Classes for Image Annotation and Retrieval. IEEE Trans. on Pattern Anal Mach Intell. 29, 394–410 (2007)
14. Lin, S., Yao, Y., Guo, P.: Speed Up Image Annotation Based on LVQ Technique with Affinity Propagation Algorithm. In: Wong, K.W., Mendis, B.S.U., Bouzerdoum, A. (eds.) ICONIP 2010. LNCS, vol. 6444, pp. 533–540. Springer, Heidelberg (2010)
15. Linde, Y., Buzo, A., Gray, R.M.: An Algorithm for Vector Quantizer Design. IEEE Trans. on Commun. 28(1), 84–95 (1980)
16. Bishop, C.M.: Pattern Recognition and Machine Learning, ch. 9, sec. 3. Springer, Heidelberg (2006)
17. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, 3rd edn., ch.11, sec. 2. Academic Press, Salt Lake City (2006)
18. Visual Object Classes, `http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2010/`
19. IGPR Images, `http://igpr.bnu.edu.cn/~dyang/imageset/`

Learning Variance Statistics of Natural Images

Libo Ma, Malte J. Rasch, and Si Wu

Institute of Neuroscience, Chinese Academy of Sciences,
320 Yue Yang Road Shanghai 200031 China
{malibo,malte,siwu}@ion.ac.cn

Abstract. In this paper, we show how a nonlinear transformation can be applied to model the variance statistics of natural images resulting in a sparse distributed representation of image structures. A variance representation of the input is learned from raw natural image patches using likelihood maximization. The simulation results demonstrate that the model can not only learn new families of basis functions, including multi-scale blobs, Gabor-like gratings and ridge-like basis functions, but also captures more abstract properties of the image, such as statistical similarity of regions within natural images. Moreover, in contrast to traditional linear model, such as sparse coding and ICA, responses show only very little residual dependencies.

Keywords: Natural Image Statistics, Variance Dependencies, Generalized Gaussian Distribution, Receptive Fields.

1 Introduction

As humans, we can effortlessly differentiate between natural images and man-made pictures or random noise images. When one naively tries to construct an image by randomly assigning luminance values for each pixel, it is highly unlikely that the constructed random image appears to be a picture taken of a natural scene. This is because the statistical structure of natural images differs profoundly from simple noise images in that they show complex dependencies between luminance values at relative pixel locations. At the present stage, the true statistics of natural images is far from being well characterized, and, due to its complexity, finding a compact statistical description of natural images is difficult.

It is widely hypothesized that neurons in early sensory systems represent sensory information efficiently [1,2]. Over the past twenty years, this theory has been applied to derive efficient codes for the processing of natural signals. Independent component analysis (ICA) [3,4] and sparse coding [5] have been developed to linearly transform signals to a new representation in which individual components are as statistically independent as possible. These methods derived basis functions that resemble the localized receptive fields of simple cells in primary visual cortex when applied to natural images [5,6,7].

Most of these algorithms presume that the input signals are generated by a linear mixture of independent source signals. For natural signals, however, this linearity assumption is mostly not valid. Indeed, in case of natural signals the responses of such

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 429–436, 2011.

linear filters still exhibit striking dependencies [8,9]. Apparently, these linear models are not sufficient to capture the complex statistical structures of natural images. In particular, it remains an open question how to represent or model higher-order image structure. A number of more recent works attempt to model higher-order dependency by the pooled magnitudes or exact values of the linear filters [10,11,12].

In this paper, we take an approach to encode the statistics of the input images that is similar to hierarchical covariance model recently proposed by Karklin and Lewicki [13]. In contrast to [13], we directly learn the variance representation of the natural images. We assume that the joint probability of all pixel locations in natural images factorize and that the luminance value distribution at each pixel location obeys the generalized Gaussian distribution (GGD). Using a maximum likelihood method, we show that a distributed sparse representation can be learned directly from the raw natural signals, showing that the first linear ICA procedure in the hierarchical network of [12] can be avoided. Our main finding is that when the variance representation is directly employed on natural images, new families of basis functions emerge, including multi-scale blobs, Gabor-like gratings, and ridge-like basis functions. Our model thus naturally reproduces the high diversity of receptive fields also found in visual cortex[14], which were recently acquired by Rehn and Sommer [15] only if carefully fitting their sparseset coding model to biological data.

2 Model

In the learning methods we closely follow the procedure in [12]. However, in contrast to their model which is designed to account for the residual dependencies of the responses of an ICA model, we here assume that the intensity values x_i of each pixel location i of a natural image patch directly obey a generalized Gaussian distribution (GGD). The GGD describes a family of probability distributions under the control of two parameters, σ and β. The probability density function of the continuous random variable x of GGD takes the form:

$$P(x;\sigma,\beta) = \frac{\beta}{2\sigma\Gamma(1/\beta)} \exp\left(-\left|\frac{x}{\sigma}\right|^{\beta}\right) \tag{1}$$

where $\sigma > 0$ is the scale parameter, and $\beta > 0$ is the shape parameter. Note that for $\beta = 2$, the GGD equals a Gaussian distribution, and for $\beta = 1$ a Laplacian distribution.

We here assume that the joint distribution of all input signals (pixel locations) $\mathbf{x}$ factorizes, i.e. $P(\mathbf{x}|\boldsymbol{\sigma},\boldsymbol{\beta}) = \prod_i P(x_i|\boldsymbol{\sigma},\boldsymbol{\beta})$. However, there are striking dependencies between pixel locations of natural images, especially in their variance. Thus, similar to [12], we include these dependencies by assuming that $\boldsymbol{\sigma}$, the scale of the variance for the GGD, is modeled as a distributed non-linear transformation of N latent higher-order variables $\mathbf{s}$, which mediate these dependencies using basis functions $\mathbf{A}$. In the neural interpretation, $\mathbf{s}$ would be related to the neural responses and $\mathbf{A}$ to the receptive fields of the neurons. Thus the variances $\boldsymbol{\sigma}$ of all M input channels (pixel locations) can be written as

$$\boldsymbol{\sigma} = c\exp(\mathbf{As}) \tag{2}$$

where $c = \sqrt{\Gamma(1/\beta)/\Gamma(3/\beta)}$ and $\Gamma(\cdot)$ is the Gamma-function.

We assume that the joint distribution is factorizable. Then the input channels are independent when conditioned on the variance basis function coefficients, $P(\mathbf{x}|\boldsymbol{\sigma}, \boldsymbol{\beta}) = P(\mathbf{x}|\mathbf{A}, \mathbf{s}, \boldsymbol{\beta}) = \prod_i P(x_i|\mathbf{A}, \mathbf{s}, \boldsymbol{\beta})$. We can therefore write for the joint probability of the data $\mathbf{x}$ given the model parameters

$$P(\mathbf{x}|\mathbf{A}, \mathbf{s}, \boldsymbol{\beta}) = \prod_i \frac{\beta_i}{2c\exp([\mathbf{As}]_i)\Gamma(1/\beta_i)} \exp\Big(-\Big|\frac{x_i}{\exp([\mathbf{As}]_i)}\Big|^{\beta_i}\Big) \tag{3}$$

where $[\mathbf{As}]_i$ represents the ith element of the vector $\mathbf{As}$.

We further assume the neural activities $\mathbf{s}$ to be sparse and independent and hence model their distribution with a Laplacian distribution, $P(\mathbf{s}) = \prod_j P(s_j) \propto \prod_j \frac{1}{2c}\exp(-\big|\frac{s_j}{c}\big|)$ (obtained from generalized Gaussian distribution with $\beta_j = 1$, and variance $\sigma_j = c$).

Having established the model we would like to learn the model parameters $\mathbf{A}$ and $\mathbf{s}$ from the input statistics by maximizing the mean log likelihood of the input data. The goal is to find a set of basis functions, $\hat{\mathbf{A}}$, such that $\hat{\mathbf{A}} = \arg\max_{\mathbf{A}} \langle \log P(\mathbf{x}|\mathbf{A})\rangle$.

To compute $P(\mathbf{x}|\mathbf{A})$, we approximate the integral over all possible neural states

$$P(\mathbf{x}|\mathbf{A}) = \int P(\mathbf{x}|\mathbf{A}, \mathbf{s})P(\mathbf{s})d\mathbf{s} \approx P(\mathbf{x}|\mathbf{A}, \hat{\mathbf{s}})P(\hat{\mathbf{s}}) \tag{4}$$

with a single estimation at the maximum a posterior value $\hat{\mathbf{s}}$, i.e. $\hat{\mathbf{s}} = \arg\max_{\mathbf{s}} P(\mathbf{s}|\mathbf{x}, \mathbf{A})$.

Now, the objective function $\mathcal{L}$ is given by

$$\begin{aligned}\mathcal{L} &= \log P(\mathbf{x}|\mathbf{A}) \approx \log P(\mathbf{x}|\mathbf{A}, \hat{\mathbf{s}})P(\hat{\mathbf{s}}) \\ &\propto -\sum_{i=1}^{M} [\mathbf{As}]_i - \sum_{i=1}^{M} \Big|\frac{x_i}{c\exp([\mathbf{As}]_i)}\Big|^{\beta_i} - \sum_{j=1}^{N} \Big|\frac{s_j}{c}\Big|\end{aligned} \tag{5}$$

where M is the dimension of the input data and N is the number of basis functions. The basis functions can be learned by gradient ascent. The optimization procedure is divided into two stages: (1) adapt the basis functions $\mathbf{A}$ (2) determine the coefficients $\mathbf{s}$ given the input data $\mathbf{x}$, while holding the basis functions fixed. Taking the partial derivative of $\mathcal{L}$, one can obtain the learning rules for the gradient ascent.

3 Learning and Results

We build the training set by randomly extracting 16×16 image patches from a standard set of ten 512×512 natural images as in [5]. DC components were removed, and the shape parameters β_i were set to 1 for all i. The number of basis functions were set to 1024 achieving a four times over-completeness. The basis functions $\mathbf{A}$ are initialized to Gaussian random values and the coefficients $\mathbf{s}$ are initialized to small random values. We use a batch of 100 image patches to infer the MAP estimate $\hat{\mathbf{s}}$ using gradient ascent with fixed learning rate of 0.01 (depending somewhat on the type of data). After convergence of the maximization of $\hat{\mathbf{s}}$ (about 50 steps), we update the basis functions $\mathbf{A}$ once. After each update the length of all basis functions $\mathbf{a}_i$ (the column of matrix $\mathbf{A}$) are re-normalized to unity.

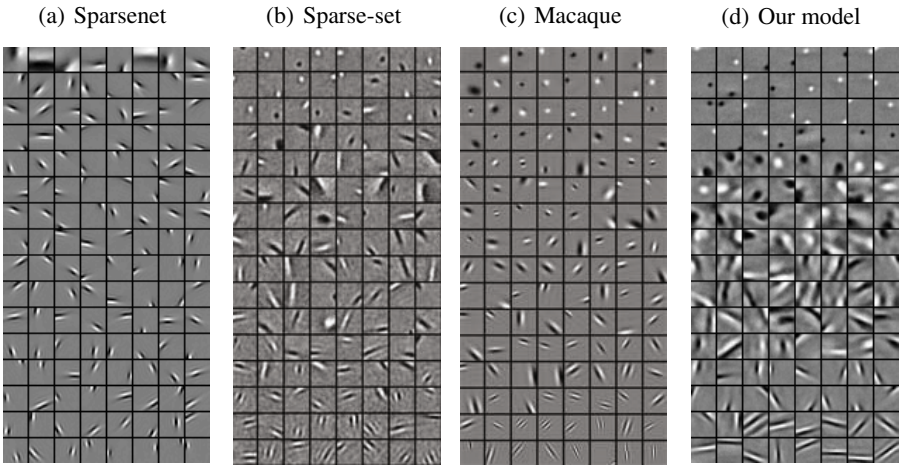


Fig. 1. Comparison of receptive fields (RFs) from three models and from recordings in monkey primary visual cortex. Each panel displays 128 randomly selected cells, sorted according to the shape. (a) RFs of the Sparsenet model [5]. (b) RFs of the sparse-set model [15]. (c) A Gabor fit to experimental measured RFs in macaque primary visual cortex. (d) RFs structures learned by our model. One notes the high diversity in RF shapes, including multi-scale-blobs, oriented Gabor patches of different frequencies and ridge-like elongated Gabor patches. For better visualization we have subtracted the background level of each RF (the median) and scaled the gray values. Figures (b) and (c) are re-drawn from [15] with permission of the authors.

Fig. 1 shows the basis functions **A** learned on natural images (Fig. 1d). Our model produces diverse families of basis functions that match the diversity found in nature very well. We found multi-scale blob-like and non-oriented RFs (top several rows in Fig. 1d), as well as Gabor-like RFs and elongated ridge-like functions (bottom rows). In Fig. 1, our results are contrasted with the results of two other efficient coding models, as well as compared to recordings from primary visual cortex in monkey [14]. The displayed RFs were randomly selected from the models and from the experimental data. One notes that the Sparsenet model of [5] only converges to localized Gabor-like RFs with rather stereotypical form. In contrast, in the visual cortex RF structures are very diverse: not only Gabor-like RFs of different spatial frequencies and elongations can be observed, but also non-oriented blob-like RF appearances. This natural occurring diversity is very similar to the learned basis function of our model. A recent study by Rehn and Sommer [15] exhibit a similar diverse family of basis functions, when adjusting a sparseness parameter in order to fit the RF shapes of the biological data closer. However, note that we do not fit parameters to the RF shapes of biological data. The structure of the RFs are directly inferred from the statistics of natural images.

3.1 Image Patches Classified by the Model Response

One feature of the proposed model is the emergence of a sparse representation of similar statistical patterns in natural image patches. To show whether the model can represent

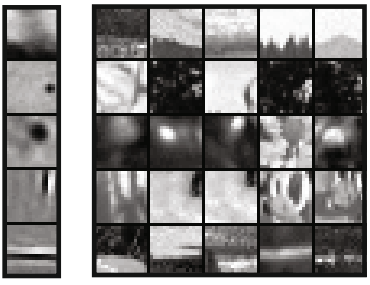


Fig. 2. Selected image patches which highly activate a neuron corresponding to a particular basis function. Five types of basis functions are selected according to their typical shapes: low-frequency, small blob, big blob, localized Gabor-patch, and ridge like appearance (left column). Highly activating image patches are shown on the right. Each row on the right corresponds to the basis function in the same row on the left. One notes that the structure of the image patches correspond to the shape of the basis functions.

similarity of patterns in natural images, we look for image patches from natural images that produce the greatest responses: for a given basis function $\mathbf{a}_i$, we search for image patches that yield large values of the basis coefficient s_i (Fig. 2).

We selected five typical shapes of basis functions, namely low frequency structure, small blob-like, large blob-like, Gabor-like and ridge-like basis functions. One can see that the image patches which mostly activate a particular basis function display a high luminance fluctuation (high contrast) in locations, where the absolute value of the basis function is high and a rather uniform luminance (low contrast), where the values of the basis functions are near zero. Note that the basis functions (together with the coefficients) code for the estimated variance in the luminance values per pixel location but not for the luminance value itself (as in traditional ICA models). Therefore image patches similarly activating one basis function can show very different luminance values (see right panel of Fig. 2 row-wise, some highly activating image patches are almost black, others almost white).

3.2 Segmentation of Statistical Similar Regions in Natural Images

Since we approximate the luminance distributions of natural image patches, similar variance structure in image patches will results in responses of similar neurons. If one estimates the responses of the neurons for all local regions (16×16 pixels) within a much larger natural image, objects (defined by a similar variance distribution of their surface texture) therefore tend to segregate in the neural responses. In Fig. 3b we show the identity of the maximal responsive neuron within randomly selected pool of 50 neurons in color code. One notes that similar textures tend to be represented by similar neurons, for instance regions containing the texture of tree bark (blue regions). In contrast, in the Sparsenet model edges of particular orientation are detected regardless of the local statistical structure.

Because the proposed model approximates the variances of each pixel location, we can infer the variance structure given a particular image patch: after learning the basis

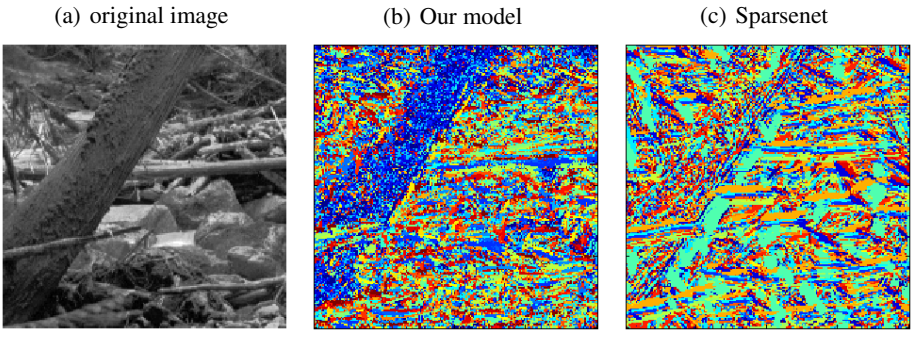


Fig. 3. Segmentation of locally similar statistical structures in natural images. The responses of our model are tuned to the structure of local luminance variations. Neurons therefore naturally tend to segregate similar textures. (a) Original gray scale image (b) Identity of maximally responsive neurons in our model plotted in color code. One notes that the tree bark is very well represented by only few neurons (blue region). (c) As in plot (b) but for the Sparsenet model [5], many different neurons code for the tree. Edges but not textures are pronounced.

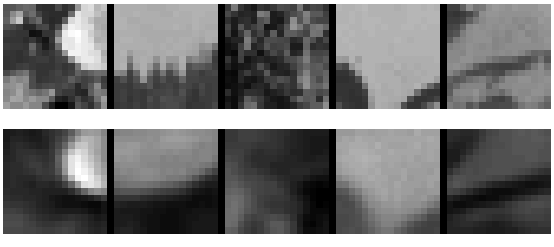


Fig. 4. Variance images. Upper row shows five example natural image patches (luminance images). The lower row displays the corresponding "variance images", i.e. the estimated variances of the generalized Gaussian distribution for each pixel location. Note that structure seen in the luminance images are also present in the variance images albeit in a smoothed fashion.

functions $\mathbf{A}$ and calculating the MAP estimate of the responses $\hat{\mathbf{s}}$ to one particular image patch, we can calculate $\boldsymbol{\sigma}$ according to equation 2. We call the variance estimates for each pixel location the "variance image". We found that the relative value of the variances between pixel locations (as mediated by the basis functions) contain similar structure as in the luminance values of the raw image patches (see Figure 4). However, the variance image appears spatially smoothed. This smoothness has the advantage to allow the model to generalize over image patches having roughly similar luminance structure but differ in their fine details. This property is important for the ability to segregate regions of similar local image structure from the background.

3.3 Conditional Distribution of Responses

Traditional ICA and sparse coding methods residual dependencies typically remain between filter responses after learning the basis functions on natural images, indicating

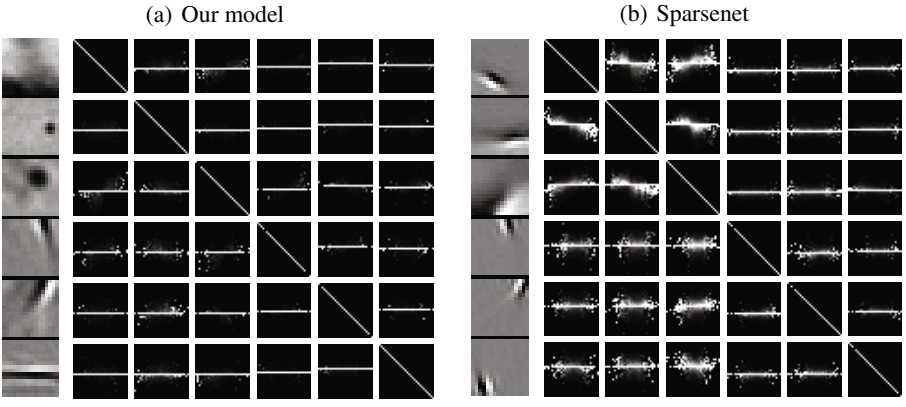


Fig. 5. Conditional histograms of the coefficients s_i. (a) The conditional histograms between six selected basis functions having typical RF shapes (shown in the left column) for all pairwise combinations. The residual dependencies of the responses are very low. (b) The conditional histograms in case of the Sparsenet model [5] calculated on the same set of images. The residual dependencies are much higher.

that a linear model is not able to extract all present structure from natural images. The conditional dependencies typically have a "bowtie" shape [9] (see also Fig. 5). Fig. 5 shows the conditional histograms of typical basis functions in case of the Sparsenet model [5] and for the proposed model. We have chosen six basis function having characteristic appearance, however, conditional histograms are very similar for nearly all learned basis functions. Note that we have selected two Gabor-like RFs in both models that have almost identical structure, to allow a direct comparison between the two models (the forth and fifth row in Fig. 5a and Fig. 5b). It is apparent that the responses in the proposed model are typically much less dependent, suggesting that more dependency structure is extracted.

4 Discussion

We have demonstrated that modeling the variance statistics of natural images yields nonlinear representations that are different from those obtained using linear generative models. The resulting basis functions show a high diversity of RF shapes, including Gabor-patches, multi-scale blobs and ridges. Moreover, in contrast to traditional ICA or sparse coding methods, we found that neuronal responses exhibit much less residual dependencies. This low residual dependency is in accordance with recent findings in the visual cortex that single cortical neurons are mostly decorrelated [16].

Furthermore, our approach is quite general and not restricted to natural images. It could also be applied to other types of structured signals such as auditory signals, gene-sequences or texts, to learn a representation capturing abstract properties in the local statistical structure of these signals. Presently, the shape parameter β is set to one, because we empirically found it suitable for DC removed natural images. For other types

of data the shape parameter β might have to be changed. One possible extension of our model would be to simultaneously infer the shape parameter β from the data.

Acknowledgements. This work was partly supported by a scholarship of the German Academic Exchange Service (MJR).

References

1. Attneave, F.: Some informational aspects of visual perception. Psychological review 61(3), 183–193 (1954)
2. Barlow, H.B.: Possible principles underlying the transformations of sensory messages. In: Rosenblith, W. (ed.) Sensory Communication, pp. 217–234. MIT Press, Cambridge (1961)
3. Comon, P.: Independent component analysis, a new concept? Signal processing 36(3), 287–314 (1994)
4. Hyvarinen, A., Karhunen, J., Oja, E.: Independent component analysis. J. Wiley, New York (2001)
5. Olshausen, B., Field, D.: Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature 381(6583), 607–609 (1996)
6. Bell, A., Sejnowski, T.: The 'independent components' of natural scenes are edge filters. Vision Research 37(23), 3327–3338 (1997)
7. Van Hateren, J., Van der Schaaf, A.: Independent component filters of natural images compared with simple cells in primary visual cortex. Proceedings of the Royal Society B: Biological Sciences 265(1394), 359–366 (1998)
8. Wegmann, B., Zetzsche, C.: Statistical dependence between orientation filter outputs used in a human-vision-based image code. In: Proceedings of SPIE, vol. 1360, p. 909 (1990)
9. Schwartz, O., Simoncelli, E.: Natural signal statistics and sensory gain control. Nature Neuroscience 4(8), 819–825 (2001)
10. Hyvärinen, A., Hoyer, P.O.: A two-layer sparse coding model learns simple and complex cell receptive fields and topography from natural images. Vision Research 41(18), 2413–2423 (2001)
11. Welling, M., Hinton, G., Osindero, S.: Learning sparse topographic representations with products of student-t distributions. Advances in Neural Information Processing Systems 15, 1359–1366 (2003)
12. Karklin, Y., Lewicki, M.S.: Learning higher-order structures in natural images. Network: Computation in Neural Systems 14, 483–499 (2003)
13. Karklin, Y., Lewicki, M.: Emergence of complex cell properties by learning to generalize in natural scenes. Nature 457, 83–86 (2009)
14. Ringach, D.: Spatial structure and symmetry of simple-cell receptive fields in macaque primary visual cortex. Journal of Neurophysiology 88(1), 455 (2002)
15. Rehn, M., Sommer, F.: A network that uses few active neurones to code visual input predicts the diverse shapes of cortical receptive fields. Journal of Computational Neuroscience 22(2), 135–146 (2007)
16. Ecker, A.S., Berens, P., Keliris, G.A., Bethge, M., Logothetis, N.K., Tolias, A.S.: Decorrelated Neuronal Firing in Cortical Microcircuits. Science 327(5965), 584 (2010)

Real-Time Joint Blind Speech Separation and Dereverberation in Presence of Overlapping Speakers

Rudy Rotili, Emanuele Principi, Stefano Squartini, and Francesco Piazza

A3LAB, Department of Biomedics, Electronics and Telecommunications, Università Politecnica delle Marche, Via Brecce Bianche 1, 60131 Ancona, Italy
{r.rotili,e.principi,s.squartini,f.piazza}@univpm.it
http://www.a3lab.dibet.univpm.it

Abstract. Blind source separation and speech dereverberation are two important and common issues in the field of audio processing especially in the context of real meetings. In this paper a real time framework implementing a sequential source separation and speech dereverberation algorithm based on blind channel identification is taken as starting point. The major drawback of this approach consists in the inability of the BCI stage of estimating the room impulse responses when two or more sources are concurrently active. To overcome the aforementioned disadvantage a speaker diarization system have been successfully inserted in the reference framework to pilot the BCI stage. In such a way the identification task can be accomplished by using directly the microphone mixture making the overall structure well suited for real-time applications. The proposed solution works in frequency domain and the NU-Tech software platform has been used on purpose for real-time simulations.

Keywords: Blind Source Separation, Speech Dereverberation, Speaker Diarization, Real-time Signal Processing, NU-Tech.

1 Introduction

The meeting scenario is one of the hardest situation to handle in the context of audio signal processing. In such a situation the extraction of a desired speech signal from mixtures picked up by microphones placed inside an enclosure can be a difficult task. In a multiple input multiple output (MIMO) acoustic system, the speech mixtures consist of a speech signal corrupted by the interference from other co-existing sources and the echoes due to the reverberation produced by multiple acoustic paths. Blind source separation (BSS) and speech dereverberation techniques are then required in order to retrieve the clean source signals.

In [1] a two stage approach leading to sequential source separation and speech dereverberation based on blind channel identification (BCI) is proposed. A real-time implementation of this approach has been presented in [2] and it is taken as starting point in this paper. The major drawback of such implementation is the inability of the BCI stage of estimating the room impulse responses (IRs) in

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 437–446, 2011.

presence of overlapping speakers. To overcome the aforementioned disadvantage a speaker diarization system is inserted in the reference framework to steer the BCI stage. Thus, the identification task can be accomplished by using directly the microphone mixture making the overall structure well suited for real-time applications in real world scenarios.

The proposed framework has been developed on a freeware software platform, namely NU-Tech [3], which allows to efficiently manage the audio stream by means of the ASIO interface with the PC sound-card and provides a useful plug-in architecture which has been exploited for the C++ implementation. Experiments performed over synthetic conditions at 8 kHz sampling rate confirm the effectiveness and real-time capabilities of the aforementioned architecture implemented on a common PC.

2 Problem Formulation

Let us assume having M independent speech sources and N microphones with $M < N$; the relationship between them is described by an $M \times N$ MIMO FIR system. According to such a model and denoting $(\cdot)^T$ as the transpose operator, we can write the following equation for the n-th microphone signal:

$$x_n(k) = \sum_{n=1}^{N} \mathbf{h}_{nm}^T \mathbf{s}_m(k, L_h) + b_n(k), \qquad k = 1, 2, ..., K, \qquad n = 1, 2, ..., N \quad (1)$$

where $\mathbf{h}_{nm} = [h_{nm,0}\ h_{nm,1}\ \dots\ h_{nm,L_h-1}]^T$ is the L_h-taps impulse response between the m-th source and the n-th microphone ($m = 1, 2, ..., M$, $n = 1, 2, ..., N$) and $\mathbf{s}_m(k, L_h) = [s_m(k)\ s_m(k-1)\ \dots\ s_m(k - L_h + 1)]^T$. The signal $b_n(k)$ is a zero-mean gaussian noise with variance $\sigma_b^2, \forall n$. By applying the Z-transform, the MIMO system can be expressed as $\left(H_{nm}(z) = \sum_{l=1}^{L_h-1} h_{nm,l} z^{-l} \right)$:

$$X_n(z) = \sum_{m=1}^{N} H_{nm}(z) S_m(z) + B_n(z), \qquad n = 1, 2, ..., N. \quad (2)$$

Our objective consists in recovering the original clean speech sources by means of a proper source separation and speech dereverberation algorithms considering in addition the presence of overlapping speakers.

3 Algorithm Description

The framework proposed in [2] consists of three main stage: source separation, speech dereverberation and BCI. Firstly source separation is accomplished by transforming the original MIMO system in a certain number of single input multiple output (SIMO) systems and secondly the separated sources (but still reverberated) pass through the dereverberation process yielding the final cleaned-up

speech signals. In order to make the two procedures properly working, it is necessary to know the MIMO IRs of the audio channels between the speech sources and the microphones by the usage of the BCI stage.

As stated in the introductory section, the major drawback of this approach leads on the inability of the BCI stage of estimating the IRs when two or more sources are concurrently active. To overcome this disadvantage we propose to include a speaker diarization system to steer the BCI stage. In such a way, the new framework is able to detects which one of the source is actually speaking. Using the information carried out by te speaker diarization stage the BCI will perform the estimation of the IRs if the correspondent source is the only active source. The block diagram of the proposed framework is reported in Fig.1 where $N = 3$ and $M = 2$ have been considered.

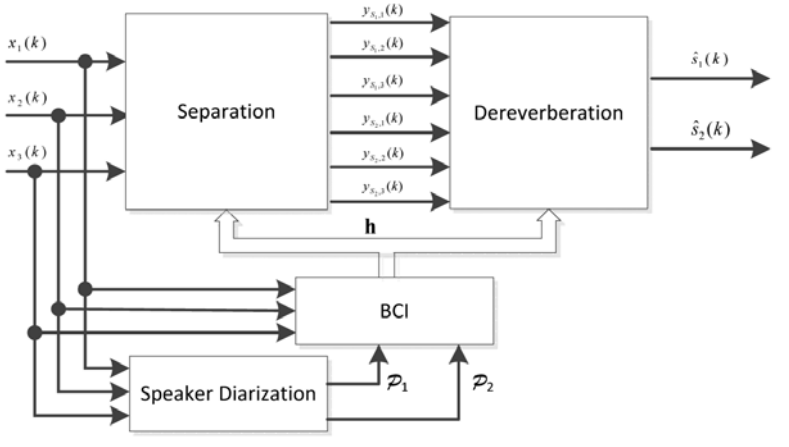


Fig. 1. Block diagram of the proposed framework

3.1 Speaker Diarization Stage

In this section we report the description of a speaker diarization system designed for real meeting, recently proposed in [4] and used in this contribution to steer the BCI stage.

First of all a voice activity detector (VAD) is applied to each channel independently in order to define the speech and non-speech frame. Let us denote $\mathrm{x}_n(f,\tau)$ as the short-time Fourier transform (STFT) of $x_n(k)$ where f is the frequency and τ is the frame index. The direction of arrival (DOA) is performed using the generalized cross correlation method with the phase transform (GCC-PHAT):

$$q'_{nn'}(\tau) = \mathrm{argmax}_{q'} \sum_f \frac{\mathrm{x}_n(f,\tau)\mathrm{x}^*_{n'}(f,\tau)}{|\mathrm{x}_n(f,\tau)\mathrm{x}^*_{n'}(f,\tau)|} e^{j2\pi f q'} \tag{3}$$

where $q'_{nn'}(\tau)$ is the time differences of arrival (TDOA) in between a microphone pair $n - n'$. The DOA vector $\mathbf{q}(\tau)$ is calculated by the TDOA information $\mathbf{q}'(\tau)$,

which consists of the $q'_{nn'}(\tau)$ of all microphones pairs, and the given microphone coordinate information $\mathbf{D}$:

$$\mathbf{q}(\tau) = c\mathbf{D}^{\dagger}\mathbf{q}'(\tau) \tag{4}$$

where c is the propagation velocity of the signals, $\mathbf{D}$ is the microphone coordinate information matrix and † denotes the Moore-Penrose pseudo-inverse. The DOA vector can be written as

$$\mathbf{q}(\tau) = [\cos\theta(\tau)\cos\phi(\tau), \sin\theta(\tau)\cos\phi(\tau), \sin\phi(\tau)]^T \tag{5}$$

where $\theta(\tau)$ and $\phi(\tau)$ are the source azimuth and the elevation respectively. The speaker diarization output, i.e. the individual speaker periods $\mathcal{P}_k$ are determined by clustering the estimated DOA at all speech frames τ in each speech period.

The previous description represents the basic speaker diarization method. In order to overcome some problems with this approach in [4] were also proposed three refined methods. Since the system cannot estimate multiple DOAs even if there are some speakers in a frame, the GCC-PATH have been substituted with a DOA estimation at each time-frequency slot (TFDOA). The second refinement concerns the suppression of the noise influence using amplitude weights for the DOA clustering procedure, while the third refinement employ a probabilistic VAD to make more robust the speech/non-speech discrimination. The complete description of the overall system with the implementation details and related references can be found in [4].

3.2 Blind Channel Identification Stage

MIMO blind system identification is typically obtained by decomposing the MIMO system in a certain number of SIMO subsystems in order to make the problem tractable and use powerful algorithms to properly estimate involved IRs. The solution can be achieved using different techniques among them we can cite the subspace methods [5] and adaptive filters [6]. Considering a real-time scenario the adaptive filter techniques are the most suitable. In particular the so-called unconstrained normalized multichannel frequency-domain LMS (UNMCFLMS) [6] algorithm, which has been employed here, represents an appropriate choice in terms of estimation quality and computational cost.

3.3 Source Separation Stage

In this section we brief review the procedure already described in [1] according to which it is possible to transform an $M \times N$ MIMO system (with $M < N$) in M $1 \times N$ SIMO systems free of interferences, as described by the following relation:

$$Y_{s_m,p}(z) = F_{s_m,p}(z)S_m(z) + B_{s_m,p}(z), \quad m = 1, 2, \ldots, M, \quad p = 1, 2, \ldots, P \tag{6}$$

where $P = C_N^M$ is the number of combinations. It must be noted that the SIMO systems outputs are reverberated, likely more than the microphone signals due to the long IR of equivalent channels $F_{s_m,p}(z)$. Related formula and the detailed description of the algorithm can be found in [1].

3.4 Speech Dereverberation Stage

Given the SIMO system corresponding to source s_m, let us consider the polynomials $G_{s_m,p}(z), p = 1, 2, \ldots, P$ as the dereverberation filters to be applied to the SIMO outputs to provide the final estimation of the clean speech source s_m, according to the following:

$$\hat{S}_m(z) = \sum_{p=1}^{P} G_{s_m,p}(z) Y'_{s_m,p}(z). \tag{7}$$

Typically optimal filtering is considered, as done in [1], but also adaptive solutions can be employed. The same iterative solution presented in [7] and adopted in [2] is still used here to achieve a real-time implementation of the overall algorithm.

4 Real Time Implementation

This section is devoted to show how the entire framework has been implemented in real-time within the Nu-tech platform [3]. NU-Tech allows the developer to concentrate on the algorithm implementation without worrying about the interface with the sound card. The ASIO protocol is also supported to guarantee low latency times. NU-Tech architecture is plug-in based: an algorithm can be implemented in C++ language to create a NUTS (NU-Tech Satellite) that can be plugged in the graphical user interface. Inputs and outputs can be defined and connected to the sound card inputs/outputs or other NUTSs. To achieve a more optimized and efficient code, all the NUTSs are written by using the Intel® Integrated Performance Primitives (Intel® IPP). In Fig.2 is shown the global scheme of the various plug-in and their interconnection used for testing sessions. Five main NUTSs have been developed on purpose, four corresponding to the main stages of the algorithmic architecture (i.e. *SPEAKER_DIARIZARION*, *BCI*, *SEPARATION*, *DEREVERBERATION*) and one devoted to performance evaluation (*EVALUATION*).

The speech signals loaded in *FileRead(0)* are mixtures captured by the microphones and are the inputs for *SEPARATION*, *SPEAKER_DIARIZARION* and the two *BCI* NUTSs. The latter take as input the speaker diarization result also in order to provide the IR estimates, the corresponding NPM values (see section 5.1 for proper definition) and value of cost function for each frame. IR estimates are then used by *SEPARATION*, *DEREVERBERATION* and *EVALUATION*. Signals delivered by *SEPARATION* feed *DEREVERBERATION* which provides the clean speech estimates. At each stage, the output signals are used by *EVALUATION* together with the original sources (loaded in *FileRead(1-2)*) for performance evaluation. The block *ADelay(0-1)* has been inserted to properly align original and estimated speech signals and therefore guarantee the correct performance calculation. Nu-Tech built viewers are used to visualize all the quality indexes described in section 5.1. Through the *Switch* blocks the user can decide which audio signal send to the PC loudspeakers.

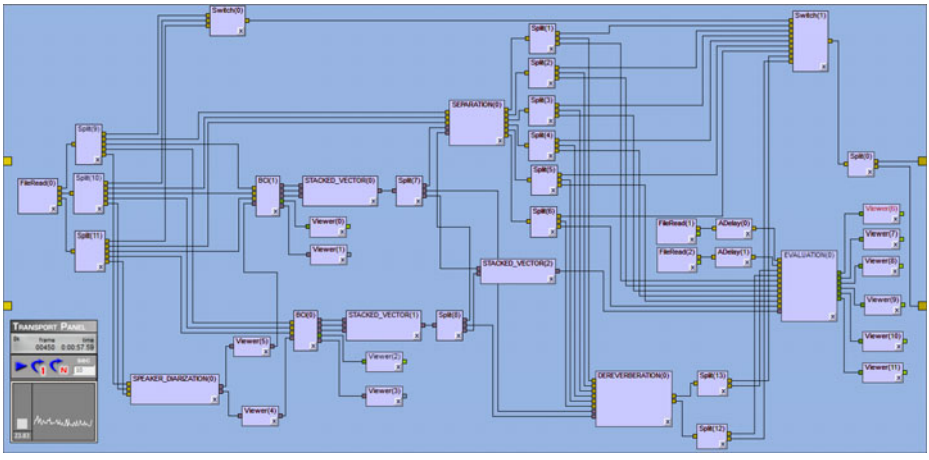


Fig. 2. Nu-Tech setup

5 Computer Simulations

Some computer simulations have been performed to prove the effectiveness of the proposed solution, both in terms of separation/dereverberation performances and also real-time processing capabilities.

5.1 Experimental Setup and Performance Indexes

The setup depicted in Fig 3 has been considered in our test sessions. It consists of two independent speech sources (1 male and 1 female) sampled at 8 kHz and three microphones arranged in a triangle configuration. MIMO IRs have been generated using the tool RIR Generator [8] with all FIR filters of 256-sample long and reverberation time set to $T_{60} = 310$ ms. It is important to note that the separation angle between the sources is set to $\alpha = 120°$. This angle was chosen according to the experiment conducted in [4] and related works. To better understand ad illustrate the behaviour of the proposed framework two proper source signal have been constructed. In our simulations no additive noise has been considered. The mixture are constructed in such a way all the possible situations are considered.

As in [1] some quality indexes have been used to evaluate the algorithm performances. They have been calculated in real-time through the dedicated NUTS. First we have the signal to interference ratio (SIR) related to the n-th microphone, defined as:

$$SIR_n^{in} \doteq \frac{1}{M} \sum_{m=1}^{M} \frac{E\{[h_{nm} * s_m(k)]^2\}}{\sum_{i=1, i\neq m}^{M} E\{[h_{ni} * s_i(k)]^2\}}. \qquad n = 1, 2, \ldots, N \qquad (8)$$

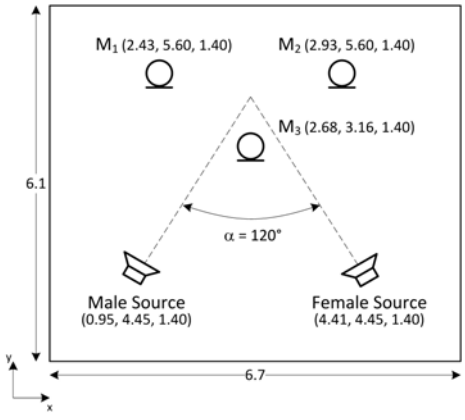


Fig. 3. Room setup (coordinate values measured in meters)

The overall SIR is the average over all microphones. The SIR after the separation stage is defined analogously. Let us note $\phi_{p,ji}(p = 1, 2, \ldots, P, i, j = 1, 2, \ldots, M)$ as the IR of the equivalent channel between the i-th input and j-th output of the p-th subsystem. The output SIR for the p-th subsystem is therefore defined as:

$$SIR_p^{out} \doteq \frac{\sum_{i=1}^{M} E\{[\phi_{p,ii} * s_i(k)]^2\}}{\sum_{j=1}^{M} \sum_{i=1, i\neq j}^{M} E\{[\phi_{p,ji} * s_i(k)]^2\}}. \qquad p = 1, 2, \ldots, P \tag{9}$$

Global output SIR is the average over all P subsystems. Comparing the global input and output SIR allows us evaluating the separation stage effectiveness. Then the well-known Itakura-Saito distance (d_{IS}) has been used to evaluate the speech distortion after the separation and dereverberation stages. It is is calculated on a frame-by-frame basis and the global d_{IS} value will be the average over all frames. In our real-time implementation two global d_{IS} values are considered, at the output of the separation and dereverberation stages. Finally, to evaluate the BCI algorithm performances the normalized projection misalignment (NPM) has been used: $\mathrm{NPM}(k) = 20 \log_{10} (\|\epsilon(k)\|/\|\mathbf{h}\|)$, where $\epsilon(k) = \mathbf{h} - \frac{\mathbf{h}^T \mathbf{h}_t(k)}{\mathbf{h}_t^T(k)\mathbf{h}_t(k)} \mathbf{h}_t(k)$ is the projection misalignment, $\mathbf{h}$ is the real IRs vector whereas $\mathbf{h}_t(k)$ is the estimated one at k-th iteration (i.e. the frame index).

5.2 Real Time Simulations

Three different configurations of the framework have been considered in order to view the benefit arising from the introduction of a speaker diarization system. In the fist configuration no speaker diarization system have been included and the BCI stage performs the IRs estimation using the reverberated version of the source signals and not the microphone mixtures. This configuration has

been exhaustively studied in [2] and reported here, with a different setup, for comparison purpose. The second configuration considers a system where the BCI is performed directly using as input the microphone mixtures without the usage of a speaker diarizazion system, while the third configuration implement the proposed framework whose block diagram and Nu-Tech configuration are depicted in Fig.1 and Fig.2 respectively.

In Fig.4 and Fig.5 are showed the obtained NPM values for the male and female source signal considering the first configuration. It can be notice that if the IRs estimation is performed using the reverberated version of the source signals and not the microphone mixtures, no problem are encountered during the estimation procedure and final NPM values are low enough to guarantee that successive stages can work properly.

For the second configuration the NPM curves are depicted in Fig.6 and Fig.7. When the BCI stage takes as input the microphone mixture the estimation is very poor since there are speech period where the two sources are concurrently active. The initial trend visible in Fig.7 is achieved since at the beginning of input signal only the female source is active but when the male source is also present, the performance rapidly decay. This means that the separation and dereverberation stages perform the required operation in a wrong way.

Fig.8 and Fig.9 show the NPM curves for the proposed architecture. In this case, the BCI stage is steered by the speaker diarization system and the IRs estimation is performed if the correspondent source is the only active source. It is possible to see that the NPM curve show the right trend and since the identification operation is not performed if an overlap period is detected the UNCFMLS do not suffer of misconvergence problem. On the other hand, comparing this results with the one obtained for the first configuration it can be notice that the convergence speed of the identification algorithm decrease if the speaker diarization stage is used.

In table Tab.1 is reported the comparison between the indices of performance evaluation averaged over all processed frames for the three different configurations[1]. The obtained results show that the introduction of the speaker diarization system decrease the performance with regard to configuration 1 but allow the system to deal with the overlapped speaker. The small decrease in performance is attributable to the error occurring in the speaker diarization result. In fact if a speaker error is commited or a false speaker time is detected the IRs are wrongly estimated. In addition in the case of missed speaker time, the BCI stage do not perform the identification even if only a source signal is active thus decreasing the convergence rate. Finally, it is important to note that, the real-time simulation has been conducted on a common end-user PC (Intel®Core2 Duo 1.83GHz, 2GB RAM) with Windows 7 32-bit operating system. The percentage of time the threads of the process used the processor is 52% including the Nutech application overload, showing a not heavy computational load.

[1] The values of SIR are computed only in the frame where the sources are concurrently active.

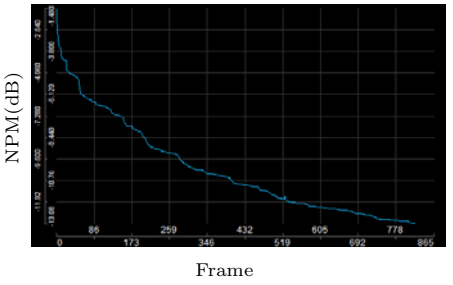


Fig. 4. NPM for source 1 (configuration 1)

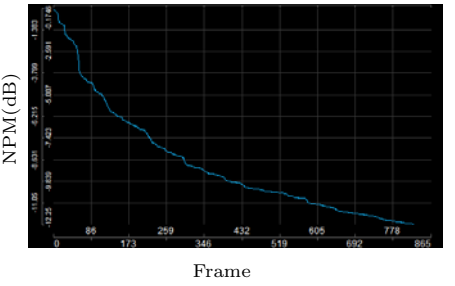


Fig. 5. NPM for source 2 (configuration 1)

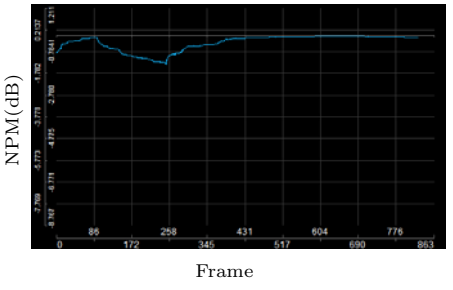


Fig. 6. NPM for source 1 (configuration 2)

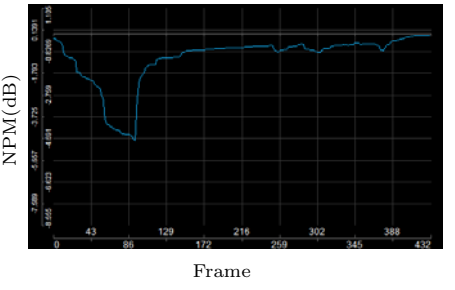


Fig. 7. NPM for source 2 (configuration 2)

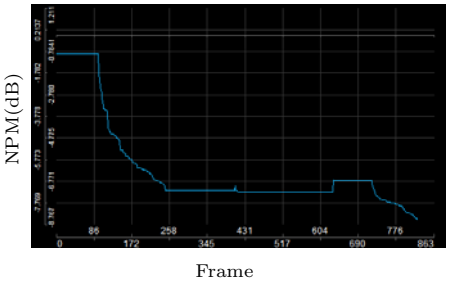


Fig. 8. NPM for source 1 (configuration 3)

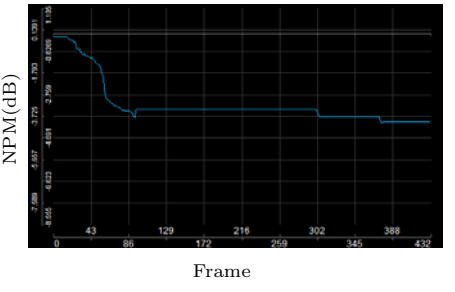


Fig. 9. NPM for source 2 (configuration 3)

Table 1. Comparison of performance of the source separation (SS) and speech dereverberation (SD) for the different configuration

	SIR^{in} (dB)	SIR^{out} (dB)	d_{IS,s_1}^{SS}	d_{IS,s_2}^{SS}	d_{IS,s_1}^{SD}	d_{IS,s_2}^{SD}
Configuration 1	10.27	47.22	1.96	2.23	0.24	0.27
Configuration 2	10.27	30.10	7.22	7.02	1.10	1.10
Configuration 3	10.27	40.82	3.12	2.85	0.31	0.37

6 Conclusions

In this paper a real-time implementation (within the PC-based NU-Tech software) of a framework for the multichannel joint speech separation and dereverberation is proposed. A speaker diarization have been included in order to make the overall structure able to deal with overlapping speakers. Simulation results show the effectiveness of the proposed approach both in terms of quality of IRs estimation and separation/dereverberation capabilities. The insertion of such a speaker diarization system makes the overall solution well-suited for a real meeting scenario. As future works, the case of additive noise will be investigated, together with the possibility to use different identification algorithm in order to make the framework more robust to the speaker diarization error improving the overall quality of the audio outputs. Different room setup with smaller separation angle will also be taken into account.

References

1. Huang, Y., Benesty, J., Chen, J.: A blind channel identification-based two-stage approach to separation and dereverberation of speech signals in a reverberant environment. IEEE Transactions on Speech and Audio Processing 13(5), 882–895 (2005)
2. Rotili, R., De Simone, C., Perelli, A., Cifani, S., Squartini, S.: Joint Multichannel Blind Speech Separation and Dereverberation: A Real-Time Algorithmic Implementation. In: Huang, D.-S., McGinnity, M., Heutte, L., Zhang, X.-P. (eds.) ICIC 2010. Communications in Computer and Information Science, vol. 93, pp. 85–93. Springer, Heidelberg (2010)
3. Squartini, S., Ciavattini, E., Lattanzi, A., Zallocco, D., Bettarelli, F., Piazza, F.: Nutech:implementing dsp algorithms in a plug-in based software platform for real time audio applications. In: Proceedings of 118th Convention of the Audio Engineering Society (2005)
4. Araki, S., Fujimoto, M., Ishizuka, K., Sawada, H., Makino, S.: A doa based speaker diarization system for real meetings. In: Hands-Free Speech Communication and Microphone Arrays, HSCMA 2008, pp. 29–32 (May 2008)
5. Moulines, E., Duhamel, P., Cardoso, J., Mayrargue, S.: Subspace methods for the blind identification of multichannel FIR filters. Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 1994, pp. IV/573–IV/576 (1994)
6. Huang, Y., Benesty, J.: A class of frequency-domain adaptive approaches to blind multichannel identification. IEEE Transactions on Signal Processing 51(1), 11–24 (2003)
7. Rotili, R., Cifani, S., Principi, E., Squartini, S., Piazza, F.: A robust iterative inverse filtering approach for speech dereverberation in presence of disturbances. In: APCCAS 2008 - 2008 IEEE Asia Pacific Conference on Circuits and Systems, pp. 434–437 (2008)
8. Habets, E.A.P.: Room impulse response (RIR) generator (May 2008), http://home.tiscali.nl/ehabets/rirgenerator.html

Audio Segmentation and Classification Using a Temporally Weighted Fuzzy C-Means Algorithm

Ngoc Thi Thu Nguyen[1], Mohammad A. Haque[1], Cheol-Hong Kim[2], and Jong-Myon Kim[1,*]

[1] School of Computer Engineering and Information Technology, University of Ulsan, Ulsan, South Korea
{nguyenthungoc.dt2,iamahsanul,jongmyon.kim}@gmail.com
[2] School of Electronics and Computer Engineering, Chonnam National University, Kwangju, South Korea
cheolhong@gmail.com

Abstract. In this paper, we present a noble method to segment and classify audio stream using a temporally weighted fuzzy c-means algorithm (TWFCM). The proposed algorithm is utilized to determine the boundaries between different kinds of sounds in an audio stream; and then classify the audio segments into five classes of sound such as music, speech, speech with music background, speech with noise background, and silence. This is an enhancement on conventional fuzzy c-means algorithm, applied in audio segmentation and classification domain, by addressing and reflecting the matter of temporal correlations between the audio signals in the current and previous time. A *3*-elements feature vector is utilized in segmentation and a *5*-elements feature vector is utilized in classification by using TWFCM. The audio-cuts can be detected accurately by this method, and mistakes caused by audio effects can be eliminated in segmentation. Improved classification performance is also achieved. The application of this method is demonstrated in segmenting and classifying real-world audio data such as television news, radio signals, etc. Experimental results indicate that the proposed method outperforms the conventional FCM.

Keywords: Audio segmentation and classification, fuzzy c-means algorithm, database retrieval.

1 Introduction

Recently, the spread of high speed access of multimedia signals and applications have created demands for databases and summarizing system as well as classify signals [1]. For these purposes, the audiovisual materials must be segmented and indexed with labels to represent its contents. A number of methods have been proposed for audio segmentation and classification [2]. Conventional methods utilize threshold processing to audio features, such as zero-crossing rate and energy of signal, to detect

* Corresponding author.

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 447–456, 2011.

the abrupt changes in the audio stream, which are called the boundaries of audio segment (audio-cuts) [1]. These methods could cause misclassification of audio that contains sound effects such as fade-in, fade-out, cross-fade, etc. A robust data clustering method is the fuzzy c-means (FCM), which can detect audio-cuts accurately even if the audio signal contains fade-in, fade-out and cross-fade [3]. By applying FCM clustering, the possibility of the existence of audio-cut can be represented as a real value between 0 and 1, thus addressing the issues of sound effect in audio cuts. All possible audio-cuts can be detected efficiently in this way. Further, FCM clustering method subdivides the audio-segments into five audio classes: silence, speech, music, speech with music background, and speech with noise background. These classification results are utilized for the performance evaluation of audio signal segmentation and classification. However, the conventional fuzzy c-means algorithm considers each point of data as an independent object without any correlation [4]. In addition, this algorithm ignores the impact of neighboring data points to its membership value and center value of each cluster, though the data elements in an audio stream are temporally correlated with each other. When the data elements are correlated then, the membership of each element for segmentation or classification is caused by its membership and the memberships of neighboring elements which depend on their distances to the considered data element [5].

In this paper we propose a temporally weighted fuzzy c-means algorithm (TWFCM) algorithm to detect audio-cuts and segment audio signals. The proposed TWFCM algorithm utilizes important correlation information between the neighboring segments and the center segment. For example, if all the neighbors around a data point in an audio signal are in the same cluster, then the center point has higher possibility to belong to this cluster. To improve the performance of audio classification, we also utilize the TWFCM. To evaluate the performance of audio classification using the proposed TWFCM algorithm, we employ the state-of-the-art cluster validity functions. Experimental results indicate that the proposed algorithm outperforms the conventional FCM in terms of audio segmentation and classification.

The remainder of this paper is divided into three sections. Section 2 describes the background information regarding to the conventional FCM and cluster validity functions. Section 3 introduces audio segmentation and classification using the proposed TWFCM algorithm, and Section 4 presents experimental results of the proposed TWFCM in comparison with the traditional FCM. Finally, Section 5 concludes this paper.

2 Background Information

2.1 Fuzzy c-Means Algorithm for Audio Segmentation and Classification

Fuzzy c-Means (FCM) is an iterative method of clustering which produces optimal partitions [6]. It allows one piece of data to belong to two or more clusters. Let an unlabelled data set $X = (x_1, x_2, x_3,\ldots, x_n)$ represents the intensity of the audio stream, where n is the number of frames. The FCM algorithm tries to sort the data set X into c clusters. The standard FCM objective function is defined as follows:

$$J_m(U,V) = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^m d^2(x_k, v_i), \quad (1)$$

where $d^2(x_k, v_i)$ represents the Euclidian distance between the data point x_k and the center v_i of i-th cluster, u_{ik} is the degree of membership of the data x_k in the k-th cluster, along with the constrain $\sum_{i=1}^{c} u_{ik} = 1$. The parameter m control the fuzziness of the resulting partition with $m \geq 1$, and c is the total number of clusters. Local minimization of the objective function $J_m(U,V)$ is accomplished by repeatedly adjusting the values of u_{ik} and v_i according to the following equations:

$$u_{ik} = \left[\sum_{j=1}^{c} \left(\frac{d^2(x_k, v_i)}{d^2(x_k, v_j)} \right)^{\frac{1}{m-1}} \right]^{-1} \quad (2)$$

$$v_i = \frac{\sum_{k=1}^{n} u_{ik}^m x_k}{\sum_{k=1}^{n} u_{ik}^m}, \quad 1 \leq i \leq c. \quad (3)$$

As J_m is iteratively minimized, v_i becomes more stable. The iteration of the FCM algorithm is terminated when the ending condition $\max_{i \leq i \leq c} \left\{ \left\| v_i^t - v_i^{t-1} \right\| \right\} < \varepsilon$ is satisfied, where $v^{(t-1)}$ is the center of the previous iteration, and ε is the predefined termination threshold. Finally, all data points are distributed into clusters according to the maximum membership u_{ik}. In addition, the fuzzy partition matrix U is congregated for further operations to evaluate the efficiency of clustering.

2.2 Cluster Validity Functions

To evaluate the classification performance quantitatively, two important types of cluster validity functions are used: the fuzzy partition and the feature structure of data set. For the fuzzy partition, partition with less fuzziness provides better performance. Fuzzy partitions include two parameters: Bezdek's partition coefficient V_{pc} and partition entropy V_{pe} [7]. These two parameters are defined as follows:

$$V_{pc}(U) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^2 \quad (4)$$

$$V_{pe}(U) = -\frac{1}{n} \left\{ \sum_{i=1}^{c} \sum_{j=1}^{n} \left(u_{ij} \log u_{ij} \right) \right\}. \quad (5)$$

When V_{pc} is maximal or V_{pe} is minimal, the optimal clustering is achieved. However, these two parameters are only depending upon the membership value of data in the clusters, not the data precisely. To overcome this shortcoming, other validity functions based on the feature structure are proposed in [8][9]. Using the feature structure of data set, a robust clustering result can be generated in which samples are compact within one cluster and separated among different clusters.

To evaluate the performance of clustering with the feature structure, two parameters are defined as follows:

$$V_{fs}(U,V,X)=\sum_{i=1}^{c}\sum_{j=1}^{n}u_{ij}^{m}\left(\left\|x_j-v_i\right\|^2-\left\|v_i-\bar{v}\right\|^2\right) \tag{6}$$

$$V_{xb}(U)=\frac{\sum_{i=1}^{c}\sum_{j=1}^{n}u_{ij}^{m}\left(\left\|x_j-v_i\right\|^2-\left\|v_i-\bar{v}\right\|^2\right)}{n\left\{\min_{i=k}\left(\left\|v_i-v_k\right\|^2\right)\right\}}, \tag{7}$$

where $\bar{v}=\frac{1}{c}\sum_{i=1}^{c}v_i$, V_{fs} is the Fukuyama-Sugeno function [8], and V_{xb} is the Xie-Beni function [9]. The smaller the values of V_{fs} or V_{xb}, the better the clustering results.

The aforementioned four cluster validity functions are the basis of comparison of the proposed TWFCM and the conventional FCM algorithm proposed in [3] for audio segmentation and classification.

3 Audio Segmentation and Classification Using a Temporally Weighted Fuzzy c-Means Algorithm

3.1 The Proposed Temporally Weighted Fuzzy c-Means Algorithm

The traditional FCM for audio segmentation and classification classifies each segment using only the attributes of that segment [3]. However, the general aspects of an audio segment are highly correlated with the aspects of its neighboring segments. Therefore, the traditional FCM algorithm leads to accuracy degradation in segmentation. This aspect of performance degradation of FCM is explored by Lung et al in the image segmentation domain [5]. To solve this problem inherent in the audio segmentation and classification domain, we propose a temporally weighted fuzzy c-means (TWFCM) algorithm that not only utilizes the current segment attributes but also considers the memberships of its neighboring segments by modifying the membership functions, in which the membership of each segment is calculated with a weighted sum of the current segment membership and the memberships of the previous neighboring segments in the window length of W_l along with the center segment x_k.

TWFCM utilizes a neighboring impact factor, called p_{ik}, to take into account the temporal information of neighbors, which is defined as the following function:

$$p_{ik}=\sum_{j=k-\frac{W_l}{2}}^{k+\frac{W_l}{2}}h(x_k,x_j)u_{ij}, \tag{8}$$

where

$$h(x_k,x_j)=\left(\sum_{j=k-\frac{W_l}{2}}^{k+\frac{W_l}{2}}\frac{d^2(x_k,x_j)}{d^2(x_k,x_i)}\right)^{-1}. \tag{9}$$

The function $h(x_k,x_j)$ is the distance coefficient between the center segment x_j and the neighbor x_j, and u_{ij} is the membership value of the neighbor x_j in the cluster i. The smaller the distance between center segment with its neighbor, the higher probability that this segment and its neighbor are at the same cluster.

To assign appropriate function of $h(x_k,x_j)$ in (9), we define some hypotheses. The neighbor impact factor p_{ik} is ranged in [0,1] with j in the range of $\left[k-\frac{W_l}{2}, k+\frac{W_l}{2}\right]$ to indicate the importance of neighbor segments. If all segments in the range of W_l completely belong to cluster i, then the impact factor value $p_{ik}=1$. This implies that this segment is mostly impacted by its neighbors. To determine the function $h(x_k,x_j)$, we assume that $u_{ik}=1$, as a result, $\sum_{j=k-\frac{W_l}{2}}^{k+\frac{W_l}{2}} h\left(x_k, x_j\right)=1$ when the neighbor impact factor $p_{ik}=1$. Therefore, the function $h(x_k,x_j)$ defined in (9) satisfies that the longer distance between x_k and x_j, the smaller value of $h(x_k,x_j)$. So, we can re-write the function p_{ik} as follows:

$$p_{ik}=\left(\sum_{j=k-\frac{W_l}{2}}^{k+\frac{W_l}{2}} \frac{1}{d^2\left(x_k, x_i\right)}\right)^{-1}\left(\sum_{j=k-\frac{W_l}{2}}^{k+\frac{W_l}{2}} \frac{u_{ij}}{d^2\left(x_k, x_i\right)}\right). \tag{10}$$

In (10), the function p_{ik} incorporates the fuzzy partition matrix U_{cxN}. We generate the distance function regarding to the impact factor p_{ik} as follows:

$$d_{new}^2\left(x_k, v_i\right)=d^2\left(x_k, v_i\right)\times p_{ik}^{-1}. \tag{11}$$

So, the new membership function is calculated as:

$$w_{ik}=\left(\sum_{j=1}^{c}\left(\frac{d_{new}^2\left(x_k, v_i\right)}{d_{new}^2\left(x_k, v_j\right)}\right)^{\frac{1}{m-1}}\right)^{-1}. \tag{12}$$

By simplifying (12), finally we get the membership function for TWFCM in (13) and the center of the clusters in (14):

$$w_{ik}=\frac{u_{ik}\times p_{ik}^{\frac{1}{m-1}}}{\sum_{j=1}^{c} u_{jk}\times p_{jk}^{\frac{1}{m-1}}} \tag{13}$$

$$v_i=\frac{\sum_{k=1}^{n} w_{ik}^m x_k}{\sum_{k=1}^{n} w_{ik}^m}, \quad 1\le i\le c. \tag{14}$$

The steps of TWFCM for audio segmentation and classification are summarized as follows:

1. Distribute the segments of audio stream into data set *X* and initiate center values $V^0 = \left(v_1^0, v_2^0, ..., v_c^0\right)$.
2. Compute the membership values u_{ik} from (2).
3. Compute new membership values w_{ik} from (13) by calculating p_{ik} from (10).
4. Calculate the new center values using (14).
5. Evaluate the termination condition $\max_{i \le i \le c}\left\{\left\|v_i^t - v_i^{t-1}\right\|\right\} < \varepsilon$. Finish if it is satisfied, otherwise go back to step 2.
6. Assign each segment according to its maximum membership to the clusters.

3.2 Audio Segmentation and Classification Using TWFCM

To segment and classify audio signals, we calculate the following feature parameters of audio signals:

1. Power of audio signal with sample steps *wl* is defined as

$$E(n) = \frac{1}{wl}\sum_{k=0}^{wl}\left(\frac{sig(k)}{\max\{abs(sig)\}}\right)^2, \tag{15}$$

where $sig(k)$ indicates the intensity of the signal in the sample k, sig indicates all the samples within the window of length *wl*, and $abs(\ )$ indicates the absolute value.

2. Parameter sequence *C(n)* is defined as

$$C(n) = \frac{\sum_{k=0}^{w_x-1} E(n+k) \times E(n-w_x+k)}{\sqrt{\sum_{k=0}^{w_x-1} E(n+k)^2 \times \sum_{k=0}^{w_x-1} E(n-w_x+k)^2}}, \tag{16}$$

where w_x is a predefined window, described in [3].

3. The mean μ_E and variance σ_E^2 of the power sequence *E(n),* where *E(n)* is in decibel value calculated by (15).
4. The mean μ_G and variance σ_G^2 of the center of gravity *G(n)*. *G(n)* is a parameter that observes alteration of a low frequency domain, and it is computed as follows:

$$G(n) = \frac{\sum_{j=0}^{17} j \times \{F_n(0,j)\}^2}{\sum_{j=0}^{17} j \times \{F_n(0,j)\}^2}, \tag{17}$$

where n, i, and j represent the granule number, sub-band number, and sample number, respectively. Thus, $F_n(i, j)$ represents the modified discrete cosine transform (MDCT) coefficient of the n-th granule, i-th subband, and j-th sample [3].

5. The zero ratio Z_R, between the number of zeros and the total number in an index sequence such as [10]

$$Z(n) = \frac{1}{2}\sum_m \left|\mathrm{sgn}[x(m)] - \mathrm{sgn}[x(m-1)]\right| w(n-m). \tag{18}$$

The feature vector for audio classification includes five parameters from the aforementioned features such as

$$V_f = \left[\mu_E, \sigma_E^2, \mu_G, \sigma_G^2, Z_R\right]. \tag{19}$$

Audio-cut detection: The proposed TWFCM utilizes parameter sequence $C(n)$ to detect the audio-cuts. In segmentation, three vectors defined in (19), (20) and (21) are grouped into two clusters by applying the proposed TWFCM algorithm:

$$P_n = \left[C(n),..., C\left(n + W_2 - 1\right)\right]^T, \tag{20}$$

$$P_{n-\Delta} = \left[C(n-\Delta),..., C\left(n - \Delta + W_2 - 1\right)\right]^T, \text{ and} \tag{21}$$

$$Z = \left[0,..., 0\right]^T. \tag{22}$$

When the distance between P_n and Z is smaller than the distance between $P_{n-\Delta}$ and P_n, then audio-cut can be obtained [3].

Audio-segment classification: the audio-segments are classified into the following five audio classes:

- Silence: An audio signal which only contains quasi-stationary background noise.
- Speech: An audio signal which contains the voices of human beings, such as the sound of conversations.
- Music: An audio signal which contains sounds made by musical instruments.
- Speech with music background: An audio signal which contains speech in an environment in which music exists in a background.
- Speech with noise background: An audio signal which contains speech in an environment in which noise exists in a background

Audio-segment classification using TWFCM utilizes the feature vector defined in (19) to classify each segment into the respective classes.

4 Experimental Results

This section evaluates the performance of the proposed TWFCM algorithm and compares it with the conventional FCM in [3]. The clustering performance is measured with several audio streams in terms of four cluster validity functions

described in Section 2.2. For all cases, we use the same empirical values such as: weighting exponent m=2.0, ε=0.001, and wl=5. We implemented and simulated the proposed TWFCM algorithm for audio segmentation and classification with Matlab7.6 on a PC platform.

To evaluate the performance of the proposed TWFCM, we use two representative audio streams, containing TV program, drama, and music. Figure 1 and Figure 2 show the original audio signals, namely *Demo1* and *Demo2*, for evaluating the performance of the TWFCM algorithm and their corresponding membership function values of audio-cuts.

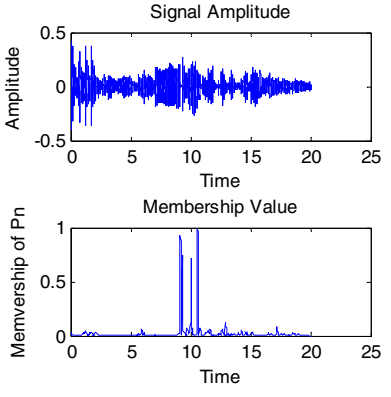


Fig. 1. *Demo1* signal amplitude and membership function values of the signal with respect to time

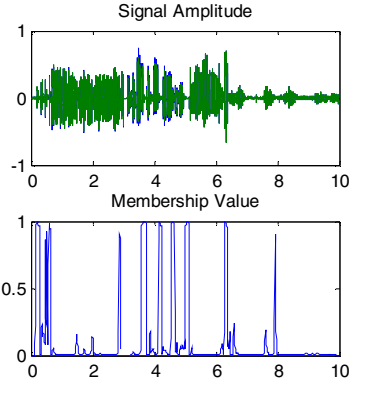


Fig. 2. *Demo2* signal amplitude and membership function values of the signal with respect to time

Audio-cut detection results with the two audio stream *Demo1* and *Demo2* are illustrated in Table 1, where the recall rate and precision rate are defined as follows:

$$\text{Recall rate} = \frac{\text{Number of correctly detected audio-cuts}}{\text{Number of manually detected audio-cuts}} \tag{23}$$

$$\text{Precison rate} = \frac{\text{Number of correctly detected audio-cuts}}{\text{Number of all detected audio-cuts}} \tag{24}$$

Table 1. Audio-cut detection results

	TWFCM		FCM	
	Demo1	*Demo2*	*Demo1*	*Demo2*
Number of all audio-cuts	14	27	14	27
Number of correct detection	13	26	11	23
Number of over detection	0	3	1	5
Number of misdetection	1	2	3	6
Recall rate	**0.929**	**0.963**	0.786	0.852
Precision rate	**1.000**	**0.929**	1.000	0.885

According to these definitions, there are a few misdetections if the recall rate is high, and there are a few over detections if the precision rate is high [3]. The proposed TWFCM outperforms the conventional FCM in both the recall rate and the precision rate in the audio-cuts detection, as shown in Table 1.

In addition, a quantitative evaluation using the cluster validity functions is necessary to analyze the classification performance. As described in Section 2.2, the better performance of the clustering is achieved if V_{pc} is maximal and V_{pe}, V_{fs}, V_{xb} are minimal. Table 2 shows performance comparison between the proposed TWFCM and the traditional FCM with the two sample audio streams *Demo1* and *Demo2*, in terms of these validity functions. The results indicate that the TWFCM algorithm outperforms the conventional FCM by all means of the cluster validity functions. We also tested the TWFCM with several audio streams, and had similar results in audio segmentation and classification.

Table 2. Performance comparison between the TWFCM and the conventional FCM, in terms of cluster validity functions

Audio files	Technique	V_{pc}	V_{pe}	V_{xb}	V_{fs}
Demo1.wav	FCM	0.7530	0.2154	0.3245	-6.6759
	TWFCM	0.8472	0.1177	0.1489	-9.1918
Demo2.wav	FCM	0.8542	0.1359	0.1844	-0.5844
	TWFCM	0.9612	0.0350	0.0626	-0.7826

5 Conclusion

In this paper, we proposed a robust audio segmentation and classification approach using a temporally weighted fuzzy c-means (TWFCM) algorithm. Unlike the conventional FCM in clustering audio signals, the proposed TWFCM utilizes the impact of previous audio signals. This results in noticeably achieving high performance than the conventional FCM algorithm. Experimental results showed that the proposed TWFCM outperforms the conventional FCM in audio segmentation and classification. These results demonstrate that the proposed TWFCM algorithm is a suitable candidate to apply in real-world applications of audio content analysis, segmentation, and retrieval.

Acknowledgement

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST) (No. 2010-0010863) and (No. R01-2008-000-20493-0).

References

1. Lu, L., Zhang, H.J., Jiang, H.: Content Analysis for Audio Classification and Segmentation. IEEE Trans. Speech Audio Processing 10(7), 504–516 (2002)
2. Brezeale, D., Cook, D.J.: Automatic Video Classification: A Survey of the Literature. IEEE Trans. on Syst, Man, and Cybernetics-Part C: Applications and Reviews 38(3), 416–430 (2008)
3. Naoki, N., Miki, H., Hideo, K.: Audio Signal Segmentation and Classification Using Fuzzy c-Means Clustering. Journal of Systems and Computers in Japan 37(4), 23–34 (2006)
4. Huang, J., Liu, Z., Wang, Y.: Integration of Audio and Visual Information for Content-based Video Segmentation. In: Proc. Int. Conf. on Image Processing, pp. 526–3530 (1998)
5. Luong, H.V., Kim, J.-M.: A Generalized Spatial Fuzzy C-Means algorithm for Medical Image Segmentation. In: Proc. of IEEE Int. Conf. on Fuzzy Systems, pp. 409–414 (2009)
6. Bezdek, J.C.: Pattern recognition with Fuzzy Objective Function algorithm. Plenum Press, New York (1981)
7. Bezdek, J.C., Nikhil, R.: On Cluster Validity for the Fuzzy c-Means Model. IEEE Trans. Fuzzy Systems 3(3), 370–379 (1995)
8. Fukuyama, Y., Sugeno, M.: A New Method for Fuzzy Clustering. In: Proc. 5th Fuzzy Syst. Symp., pp. 247–250 (1989)
9. Xie, X.L., Beni, G.A.: Validity Measure for Fuzzy Clustering. IEEE Trans. Pattern Anal. Machine Intell. 3(8), 357–363 (1982)
10. Zhang, T., Kuo, C.–C.J.: Audio Content Analysis for Online Audiovisual Data Segmentation and Classification. IEEE Transactions on Speech and Audio Processing 9(4), 441–457 (2001)

Extracting Specific Signal from Post-nonlinear Mixture Based on Maximum Negentropy

Dongxiao Ren*, Mao Ye, and Yuanxiang Zhu

School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, P.R. China and
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, P.R. China
rendx29@163.com, yem_mei29@hotmail.com

Abstract. To address the problem of extracting specific signal from the post-nonlinear (PNL) mixture, we propose a novel algorithm based on maximum negentropy. Assume that the prior knowledge of the desired source, such as its rough template referred as the references signal, is available. The closeness measurement between the corresponding estimated output and given reference signal is treated as a constraint and incorporated into the negentropy objective function. Therefore, a constrained optimization problem is formed, which is solved by the augmented Lagrange function method with standard gradient descent learning. The inverse of the unknown nonlinear function in the post-nonlinear (PNL) mixture model is approximated by the multilayer perceptions (MLP) network. Experiments on the synthesis dataset demonstrate the validity of our proposed algorithm.

Keywords: blind source extraction (BSE), the augmented Lagrange function, multilayer perceptions (MLP) network, reference.

1 Introduction

Blind source extraction (BSE) is a powerful technique which can obtain the desired sources from the mixed signals with some prior knowledge available. Compared to blind source separation (BSS) which is an important technique to recover the sources from all kinds of their mixtures, BSE has many advantages. The most obvious advantage is that only extracting the sources of interest can save a large time and reduce unnecessary computation, specially when the number of sensors is large and the number of the desired sources is small. In many practical situations, such as biomedical signal processing [1] and speech signal processing [2], only a single source or a subset of sources is subject of interest

* This work was supported in part by the National Natural Science Foundation of China (60702071), Program for New Century Excellent Talents in University (NCET-06-0811), 973 National Basic Research Program of China (2010CB732501), Fundation of Sichuan Excellent Young Talents (09ZQ026-035) and Open Project of State Key Lab. for Novel Software Technology of Nanjing University.

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 457–465, 2011.

and their prior knowledge, e.g. statistical properties or rough templates, are usually available. Therefore, in these scenarios, BSE is more appropriate and more competitive than BSS.

There are many BSE algorithms proposed in the open literature. These algorithms can be divided into two categories. One is sequential BSE [3,4], which combines extraction stage and deflation stage. By this technique, the source is recovered one by one until the desired sources have been extracted. Another kind is constrained BSE [1,5,6] which provides a general framework to incorporate some prior information, e.g. autocorrelation, kurtosis and rough templates, into the contrast function. With these prior knowledge, the constrained BSE algorithm usually has a better performance, whereas the performance of sequential BSE algorithms is easily affected by accumulation of error during deflation. Besides, due to the fact that the desired sources obtained by sequential BSE algorithms are not the first outputs, most of these algorithms are not suitable to extract the desired source at a time. However, most of these existing constrained BSE algorithms have been specially designed for the linear instantaneous mixtures which are not realistic and accurate in many practical applications [7].

To address these problems above, in this paper, we propose a novel algorithm for extracting the desired source as the first output from a specific nonlinear mixture model known as post-nonlinear (PNL) mixture. The PNL mixture is a realistic and accurate model in many situations [7] and the sources can be estimated in such a nonlinear mixture, subject to the ambiguities of permutation and scaling. In our approach, assume that some prior information of the desired source, such as its rough template, is available. A constrained optimization problem is then constructed by incorporating the prior knowledge as a constraint into the contrast function. The unknown nonlinear function in the PNL mixture is approximated by the multilayer perceptions (MLP) network, because the neural network can uniformly approximate any continuous function if there is sufficient number of neurons in the hidden layers. Finally, the desired source is extracted by the augmented Lagrange function method with standard gradient descent learning for the constrained optimization problem. Experimental results demonstrate that the proposed algorithm can successfully extract the desired source from the PNL mixture as the first output.

The manuscript is organized as follows. The mathematical model of BSE from the PNL mixture is briefly introduced in Section 2. A novel algorithm for extracting the desired source from the PNL mixture based on maximum negentropy is described in detail in Section 3. Experiments on synthetic data are performed in Section 4. Finally, discussions and conclusions are drawn in Section 5.

2 Problem Formulation

The PNL mixture consists of a linear instantaneous mixture followed by an unknown and invertible memoryless nonlinear distortion. Assume that the source vector is denoted by $S(k) = [s_1(k), s_2(k), ..., s_n(k)]^T$ and the observed signal

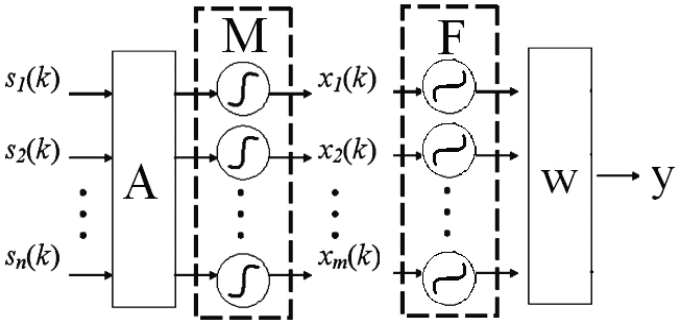


Fig. 1. The PNL mixture model and its extracting system

vector by $X(k) = [x_1(k), x_2(k), ..., x_m(k)]^T$, where $k = 1, 2, ...$ is the discrete time or sample index, n and m are the numbers of the sources and the observed signals, respectively. The PNL mixture model and its extracting system can be shown in Fig.1 where A is an unknown and nonsingular mixing matrix and $M(.) = [M_1(.), M_2(.), ..., M_m(.)]^T$ is an invertible nonlinear function vector that operates componentwise.

From Fig. 1, we can see that the extracting structure of the PNL mixture is a two-stage system, namely, a nonlinear stage followed by a linear stage. Therefore, if we just want to extract the desired source from the PNL mixture, the corresponding estimated output y can be given by

$$y(k) = w^T F(X(k)), \tag{1}$$

where $X(k) = M(AS(k)$, F is the inverse of M and w is a demixing vector.

Our goal is to extract the desired source from the PNL mixture with some prior knowledge of its rough template. The rough template, generally referred to as the reference signal, is the trace of the desired source and carries some information to distinguish the desired source, but is not identical to the corresponding source [8]. So, the desired source is both an independent component and the one closest to the given reference signal.

According to the analysis of the literature [7], when blind source separating from the PNL mixture, the output independence can be obtained if and only if $\forall j = 1, 2, ..., m, F_j(.) * M_j(.)$ are linear. Similarly, to extract the desired source from the PNL mixture, under the above restrictions, due to the central limit theory, we can use the following classical negentropy contrast function:

$$J(y) = -\rho[E\{G(y)\} - E\{G(v)\}]^2, \tag{2}$$

where ρ is a positive constant, $G(.)$ can be any non-quadratic function and v is a Gaussian variable with zero-mean and unit-variance.

For simplicity, in the following, we assume that the sources $S(k)$ are statistically independent components with zero-mean and unit-variance, the observed signals $X(k)$ are removed the correlation by whitening and $m = n$.

3 Proposed Method

To extract the desired source from the PNL mixture with the reference signal, we need to minimize the contrast function $J(y)$ and take into account the prior knowledge of its reference signal. Due to the desired source closest to the given reference signal, the closeness measurement between the corresponding estimated output and the reference signal can be treated as a constraint and incorporated into the contrast function. So, a constrained optimization problem is formed, which can be solved by the augmented Lagrange function method with standard gradient descent learning. Detailed derivation of our algorithm is as follows. For simplicity, in the following, the time index k is omitted.

Let us denote the closeness measurement between an estimated output y corresponding to the desired source and the given reference signal r by $\varepsilon(y, r)$. The demixing vector for the desired source is represented by w^* and $w_i, i = 1, 2, ..m-1$ denote the demixing vectors for other undesired independent components. Assume that the minimum value of $\varepsilon(y, r)$ is the desired source closest to r, so we have

$$\varepsilon(w^{*T}F(X), r) < \varepsilon(w_1^T F(X), r) \leq \varepsilon(w_{m-1}^T F(X), r). \tag{3}$$

The following inequality relationship is matched if and only if $w = w^*$,

$$\varepsilon(w^T F(X), r) - \tau \leq 0, \tag{4}$$

where $\tau \in [\varepsilon(w^{*T}F(X), r), \varepsilon(w_1^T F(X), r))$ is a threshold parameter.

Therefore, a constrained optimization problem is constructed by incorporating the above inequality constraint into the objective function as following:

$$\begin{cases} min\ J(y) = -\rho[E\{G(y)\} - E\{G(v)\}]^2, \\ s.t.\ \ \varepsilon(y, r) - \tau \leq 0, \end{cases} \tag{5}$$

where $y = w^T F(X)$.

The inequality constraint in Eq.(5) can be transformed into an equality constraint $\varepsilon(y, r) - \tau + z^2 = 0$ by introducing a slack variables z. To search for the optimal solution for Eq.(5), we adopt the augmented Lagrange multipliers method. The corresponding Lagrange function is given by

$$\begin{aligned} L(w, F, \lambda, \mu) = &- \rho[E\{G(y)\} - E\{G(v)\}]^2 \\ &+ \lambda[\varepsilon(y, r) - \tau + z^2] + \frac{\mu}{2}[\varepsilon(y, r) - \tau + z^2]^2, \end{aligned} \tag{6}$$

where λ is the non-negative Lagrange multiplier and μ is the scalar penalty parameter. The inequality constraint is further translated to eliminate the slack variable z as:

$$\begin{aligned} L(w, F, \lambda, \mu) = &- \rho[E\{G(y)\} - E\{G(v)\}]^2 \\ &+ \frac{1}{2\mu}\{[max(0, \lambda + \mu(\varepsilon(y, r) - \tau))]^2 - \lambda^2\}. \end{aligned} \tag{7}$$

The constrained optimization problem has been transformed into an optimization problem without constraints by the augmented Lagrange multiplier method and the new objective function is formulated in Eq.(7). Now, the PHR algorithm is used to search for the optimal solution for this new objective function.

First, a positive and non-decreasing sequence of the penalty factor $\{\mu^q, q = 1, 2, ...\}$, is selected, the initial point y^0, the initial Lagrange multiplier λ^1 and the required precision ϵ are chosen and set iteration number $q = 1$.

Second, with y^0 as the initial point, the new objective function in Eq.(7) is solved by the standard gradient descent learning. Since the demixing vector w and the nonlinear function F in $y = w^T F(X)$ are unknown, the partial derivative of the objective function with respect to w is

$$\frac{\partial L}{\partial w} = -\hat{\rho} E\{G'(y) * F(X)^T\} + [max(0, \lambda + \mu(\varepsilon(y, r) - \tau))] \frac{\partial \varepsilon(y, r)}{\partial w}, \tag{8}$$

where $\hat{\rho} = 2\rho[E\{G(y)\} - E\{G(v)\}]$.

The nonlinear function vector F can be expressed as $F = (F_1, F_2, ...F_n)^T$, where $F_i = F(\theta_i, X)$ and θ_i are the unknown parameters of F_i. The partial derivative of the new objective function with respect to θ_i is given by

$$\begin{aligned} \frac{\partial L}{\partial \theta_i} = & -\hat{\rho} E\{G'(y) * w(i) * \frac{\partial F(\theta_i, X)}{\partial \theta_i}\} \\ & + [max(0, \lambda + \mu(\varepsilon(y, r) - \tau))] \frac{\partial \varepsilon(y, r)}{\partial \theta_i}. \end{aligned} \tag{9}$$

Following the universal approximation theorem for a nonlinear input-output mapping [9], $F_i(i = 1, 2, ...n)$ can be approximated by a group of MLP networks as follows:

$$F_i(X) = \sum_{j=1}^{P} \alpha_j \sigma(\omega_j X + b_j), \tag{10}$$

where P denotes the number of hidden neurons of the j-th perceptions, ω and α are the weights of the input and the output layers, respectively, b are bias and σ represents the activation function.

The partial derivatives of Eq.(10)) with respected to α_j, ω_j and b_j can be derived as

$$\begin{cases} \frac{\partial F_i(X)}{\partial \alpha_j} = \sigma(\omega_j X + b_j), \\ \frac{\partial F_i(X)}{\partial \omega_j} = \alpha_j * X * \sigma'(\omega_j X + b_j), \\ \frac{\partial F_i(X)}{\partial b_j} = \alpha_j * \sigma'(\omega_j X + b_j). \end{cases} \tag{11}$$

The unknown parameters w and F in the extracting system of the PNL mixture can be updated by Eq.(8)-Eq.(11). To avoid the critical case where the norm of w becomes too small, after each update, w should be normalized to unit length.

Finally, when the minimum value of Eq.(7) has been obtained after the above second step, whether the algorithm meets the predetermined required precision

ϵ is judged. If the condition is satisfied, the algorithm can be terminated and the approximate values y^q is obtained; otherwise, the Lagrange multiplier is updated by

$$\lambda^{q+1} = max(0, \lambda^q + \varepsilon(y^q, r)), \tag{12}$$

and the second step of the algorithm is repeated until the maximum iteration number is reached.

4 Experiments and Discussions

To investigate the validity of our proposed algorithm, experiments are performed on a synthetic dataset obtained from [10]. Four synthetic source in this dataset are depicted in Fig.2. For simplicity, we just use the first 600 samples.

The reference signals are obtained from the sign of the corresponding source signals, drawn in Fig.3. Therefore, the reference signals have the same frequency and phase as the true ones and are used to extract the desired source with precise morphology [11]. The closeness measurement $\varepsilon(y, r)$ between an estimated output y and the reference signal r can take any form, such as the correlation coefficient $E\{yr\}$, the mean square error (MSE) $E\{(y-r)^2\}$, or any other suitable closeness measurement [12]. The correlation coefficient $E\{yr\}$ is widely used closeness measurement in BSS or BSE. The correlation coefficient $E\{yr\}$ is higher means that these two signals y and r are closer. In our experiments, the correlation coefficient $E\{yr\}$ is selected as the closeness measurement and performance index.

Assume that we want extract the desired source from the PNL mixture of the same type signals. Certainly, the desired source can also be extracted from the

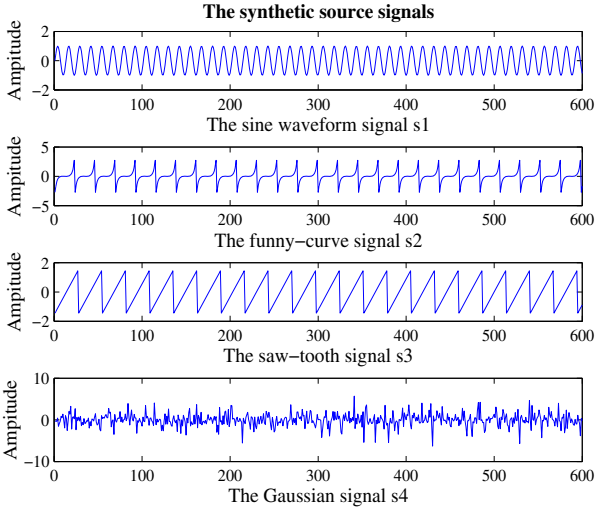


Fig. 2. The four source signals

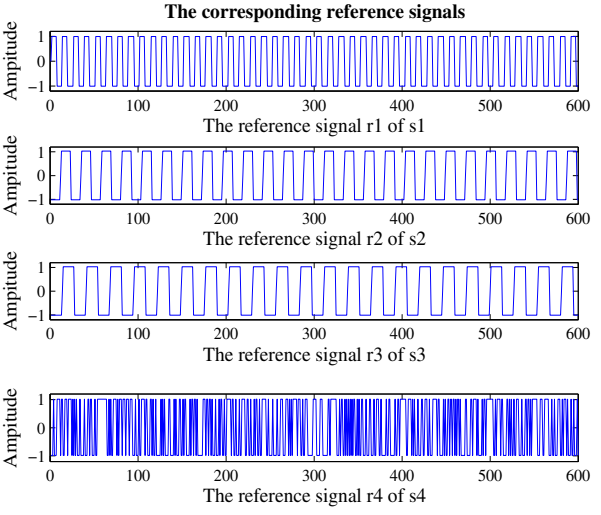


Fig. 3. The corresponding reference signals

PNL mixture of the different type. To save space, we don't provide the latter experiment results. From the Fig.2, we select s_1 and s_2 (sub-Gaussian) as the sources to mix together. The mixing matrix A is randomly selected as

$$A = \begin{pmatrix} 0.3412 & 0.7271 \\ 0.5341 & 0.3093 \end{pmatrix},$$

and the nonlinear function M is chosen as the hyperbolic tangent function $tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$ for simplicity.

After whitening the mixed signals $X(k)$, we run our proposed algorithm. If we want to extract the first source s_1, we use r_1 as the reference signal and set $\tau = 0.5$. We choose $G_1(y) = \frac{1}{a_1} logcosh(a_1 y), where\ 1 \leq a_1 \leq 2$ which is a good general purpose function for G. The experiment result is shown in Fig.4. To clear contrast, the source s_1 and the extracted signal y are drawn together. From Fig.4, it is clear to see that the waveforms of the extracted signal y are similar to those of the source s_1 except the scale of these amplitudes. Since the main information of signals are included in the waveforms, how to recover the waveforms of the desired source from the mixed signals is needed to care about and the real magnitude of the amplitude is minor. That is to say, the desired source s_1 has been extracted successfully by our proposed algorithms.

To further check the performance of our algorithm, the correlation coefficient between the source and the corresponding estimated output as the performance index. Our algorithm has been independently executed 100 times and the correlation coefficient of every trial is shown in Fig.5. The average value of these correlation coefficients is 0.937897 which means that the performance of our proposed algorithm is good.

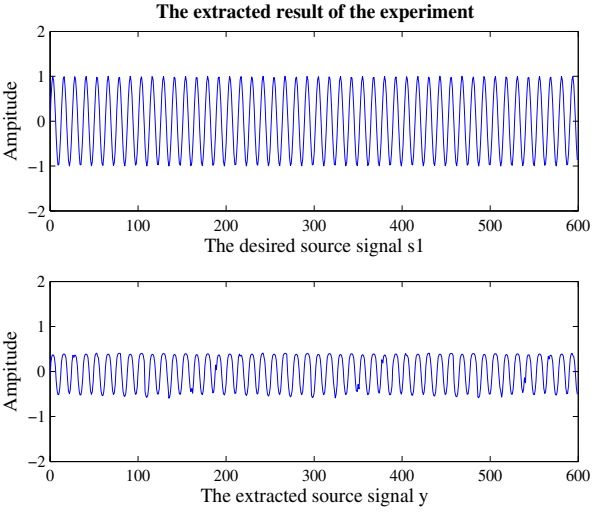


Fig. 4. The extraction result of the experiment

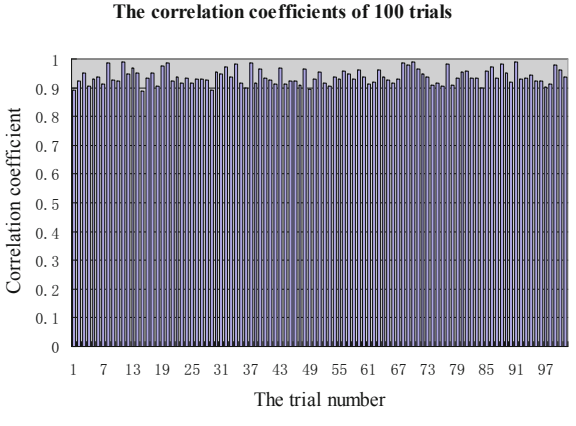


Fig. 5. The extraction result of the experiment

5 Conclusion

To extract the desired source from the PNL mixture with a given reference signal, based on maximum negentropy, we propose a novel algorithm which treats the prior knowledge as a constraint to the contrast function. A constrained optimization problem is then formed, which is solved by the augmented Lagrange function method with standard gradient descent learning. The inverse of the nonlinear function in the PNL mixture is approximated by the MLP network. The validity of our algorithm is demonstrate by simulation results.

References

1. Ye, Y.L., Sheu, P.C., Zeng, J.Z., Gang, W., Ke, L.: An efficient semi_blind source extraction algorithm and its applications to biomedical signal extraction. Science in China Series F: Information Sciences 52, 1863–1874 (2009)
2. Lin, Q.H., Zheng, Y.R., Yin, F.L., Liang, H.L.: Speech segregation using constrained ICA. In: Yin, F.-L., Wang, J., Guo, C. (eds.) ISNN 2004. LNCS, vol. 3173, pp. 755–760. Springer, Heidelberg (2004)
3. Leong, W.Y., Liu, W., Mandic, D.P.: Blind source extraction: Standard approaches and extensions to noisy and post_nonlinear mixing. Neurocomputing 71, 2344–2355 (2008)
4. Leong, W.Y., Mandic, D.P.: Post-nonlinear blind extraction in the presence of ill-conditioned mixing. IEEE Transaction on Circuits and Systems I: Regular Papers 55, 2631–2638 (2008)
5. Barros, A.K., Cichocki, A.: Extraction of specific signals with temporal structure. Neural Computation 13, 1995–2003 (2001)
6. Zhang, Z.L., Zhang, Y.: Extraction of a source signal whose kurtosis vaule lies in a specific range. Neurocomputing 69, 900–904 (2006)
7. Taleb, A., Jutten, C.: Source separation in post-nonlinear mixture. IEEE Transaction on Signal Processing 47, 2807–2822 (1999)
8. Liu, W., Rajapakse, J.C.: Approach and applications of constrained ICA. IEEE Transactions on Neural Networks 16, 203–212 (2005)
9. Kaykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edn. China Machine Press, Beijing (2004)
10. FastICA matlab package. The MathWorks, Inc., Nattick, MA, http://cis.hut.fi/projects/ica/fastica
11. Li, C.L., Liao, G.S., Shen, Y.L.: An improved method for independent component analysis with reference. Digital Signal Processiing 20, 575–580 (2010)
12. James, C.J., Gibson, O.J.: Temporally constrained ICA: an application to artifact rejection in electromagnetic brain signal analysis. IEEE Transactions on Biomedical Engineering 50, 1108–1116 (2003)

A Method to Detect JPEG-Based Double Compression

Qingzhong Liu[1], Andrew H. Sung[2,3], and Mengyu Qiao[2,3]

[1] Department of Computer Science, Sam Houston State University
Huntsville, TX 77341, U.S.A.
liu@shsu.edu
[2] Department of Computer Science and
[3] Institute for Complex Additive Systems Analysis, New Mexico Tech,
Socorro, NM 87801, U.S.A.
{sung,myuqiao}@cs.nmt.edu

Abstract. Digital multimedia forensics is an emerging field that has important applications in law enforcement, the protection of public safety, and notational security. As a popular image compression standard, the JPEG format is widely adopted; however, the tampering of JPEG images can be easily performed without leaving visible clues, and it is increasingly necessary to develop reliable methods to detect forgery in JPEG images. JPEG double compression is frequently used during image forgery, and it leaves a clue to the manipulation. To detect JPEG double compression, we propose in this paper to extract the neighboring joint density features and marginal density features on the DCT coefficients, and then to apply learning classifiers to the features for detection. Experimental results indicate that the proposed method delivers promising performance in uncovering JPEG-based double compression. In addition, we analyze the relationship among compression quality factor, image complexity, and the performance of our double compression detection algorithm, and demonstrate that a complete evaluation of the detection performance of different algorithms should necessarily include both the image complexity and double compression quality factor.

Keywords: Feature mining, SVM, digital forensics, double JPEG compression, forgery, image complexity, marginal density, neighboring joint density, DCT coefficient, quality factor.

1 Introduction

Today's digital technology allows digital media to be easily altered and manipulated. As a conspicuous example, a state-run newspaper in Egypt published in late 2010 a doctored picture in an apparent attempt to create the impression that its country's president was leading the Middle East peace talks in Washington [1, 2, 3].

JPEG images are one of the most popular media. Generally, tampering manipulation on a JPEG image involves several different basic operations, such as image resize, rotation, splicing, double compression, etc. As one decodes the bit stream of a JPEG image and implements the manipulation in spatial domain, and then compresses the modified image back into JPEG format, if the quantization matrices are different between the original image and the modified one, the latter is said to have undergone

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 466–476, 2011.

a *double JPEG compression*. Although double compression does not by itself prove malicious or unlawful tampering, it is evidence of image manipulation.

For image forgery detection, researchers have proposed several different methods, and most of them have been included in a survey [6]. For specifically detecting double JPEG compression, due to the fact that double JPEG compression changes the compressed DCT coefficients and hence modifies the histogram at a certain frequency in DCT 2-D array, Pevny and Fridrich designed a feature set comprising low-frequency DCT coefficients [15], and Chen et al. [5] designed a 324-feature set consisting of Markov transition probability on the difference 2-D array.

In studying detection of JPEG-based double compression, we explore the statistical property of DCT coefficients and find that the operations in double compression actually modify some DCT coefficients, and hence modify the marginal density at each specific frequency band or change the correlation of neighboring DCT coefficients; accordingly, we design a method to detect JPEG-based double compression. In what follows, we briefly describe related statistical models of DCT coefficients and the modification of the statistical property caused by the manipulations and design marginal density and neighboring joint density features in section 2. Experiments are given in section 3, followed by conclusions in section 4.

2 Statistical Models and Feature Mining

2.1 Characteristics and Modification

The Generalized Gaussian distribution (GGD), given below in (1), is widely used in modeling probability density function (PDF) of a multimedia signal, and is very often applied to transform coefficients such as discrete cosine transform (DCT) or wavelet ones. Experiments show that adaptively varying two parameters of the generalized Gaussian distribution (GGD) [14, 16] can achieve a good probability distribution function (PDF) approximation, for the marginal density of transform coefficients.

$$\rho(x;\alpha,\beta)=\frac{\beta}{2\alpha\Gamma(1/\beta)}\exp\left\{-\left(|x|/\alpha\right)^{\beta}\right\} \tag{1}$$

Where $\Gamma(\cdot)$ is the Gamma function, scale parameter α models the width of the PDF peak, and shape parameter β models the shape of the distribution.

An 8×8 DCT block has 64 frequency coefficients, our study shows that the marginal density of DCT coefficients at each specific frequency approximately follows the GGD distribution and certain manipulations, e.g. double JPEG compression, changes the density. Fig.1 demonstrates a singly compressed JPEG image with quality factor '75' (a); doubly compressed JPEG images with the first compression quality factor '55' (b), and '90' (c), respectively, followed by the second compression quality factor '75'; and the marginal densities at frequency coordinates (2,1), (2,2), and (1,3). Compared to the marginal density of the single compression, Fig.1(d), Fig.1(g), and Fig.1(j), the modification caused by the double compression from the low quality factor '55', shown in Fig.1(e), Fig.1(h), and Fig.1(k), is noticeable. However, the modification caused by the double compression from the high quality factor '90', Fig.1(f),(i), and (l), is not as noticeable.

Although there does not appear to exist a generally agreed upon multivariate extension of the univariate GGD, some researchers define a parametric multivariate generalized Gaussian distribution (MGGD) model that closely fits the actual distribution distribution of wavelet coefficients in clean natural images, exploit the dependency between the estimated wavelet coefficients and their neighbors or other coefficients in different subbands based on the extended GGD model, and achieve good image denoising [6]. The MDDG model is shown as follows:

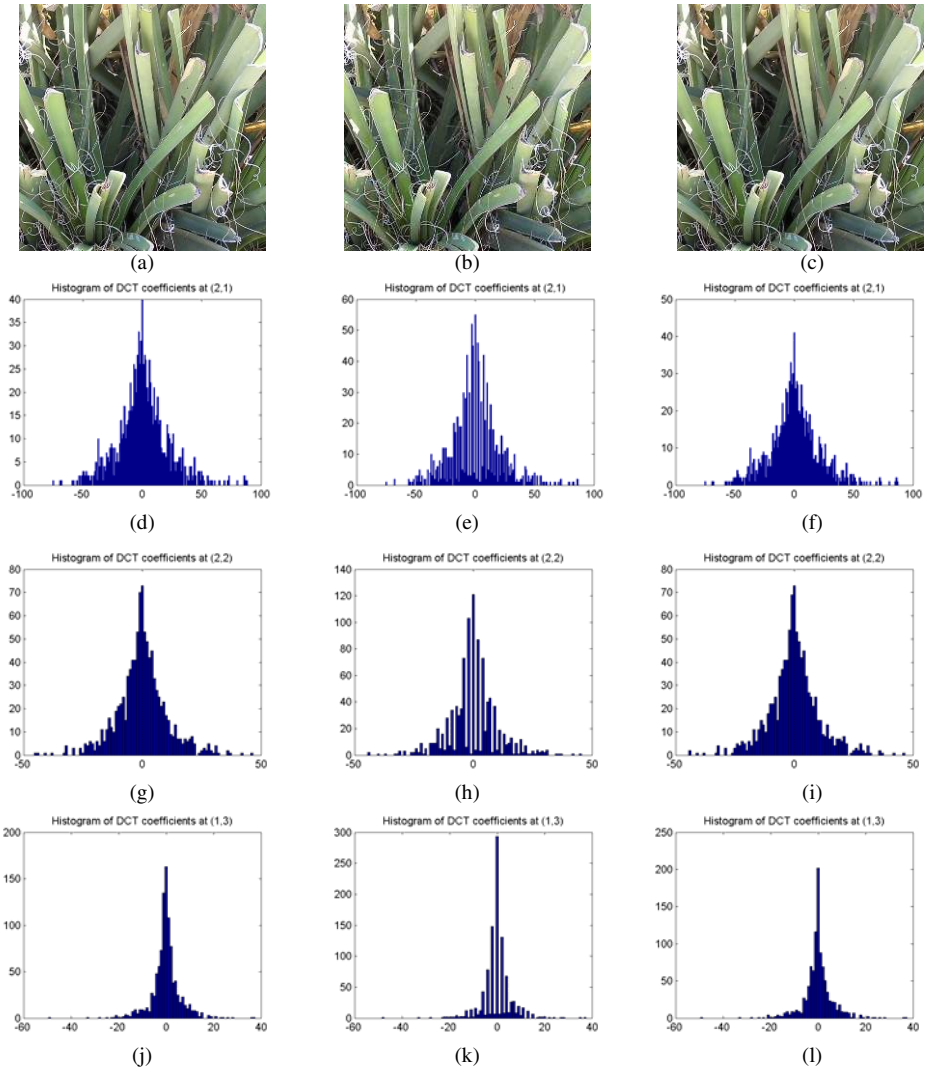


Fig. 1. Marginal densities of the singly compressed JPEG image (left) and the double compressions (middle and right). X-axis shows the values of the DCT coefficients and y-axis shows the occurrences.

$$p(\mathrm{x}) = \gamma \exp\left\{-\left(\frac{(\mathrm{x}-\mu)^t \sum_{\mathrm{x}}^{-1} (\mathrm{x}-\mu)}{\alpha}\right)^{\beta}\right\} \tag{2}$$

Where γ indicates a normalized constant defined by α and β, $\sum_{\mathrm{X}}$ is the covariance matrix and μ is the expectation vector.

To exploit the dependency between the compressed DCT coefficients and their neighbors, we study the neighboring joint density of the DCT coefficients, and postulate that some manipulation such as JPEG double compression will modify the neighboring joint density, shown by Fig. 2. Let the left (or upper) adjacent DCT coefficient be denoted by random vector X1 and the right (or lower) adjacent DCT coefficient be denoted by random vector X2; let X = (X1, X2). The DCT neighboring joint density will be modified by the manipulation, and the change hence leaves a trail of the manipulation. Fig.2(a), (b), and (c) show the neighboring joint density of the singly compressed JPEG image of Fig.1(a), of the doubly compressed JPEG image of Fig.1(b), and of the doubly compressed JPEG image of Fig.1(c). The differences of the neighboring joint density between the double compression and the single compression are given by Fig.1(d) and (e). It verifies our postulation that the neighboring joint density has been modified by the double compression.

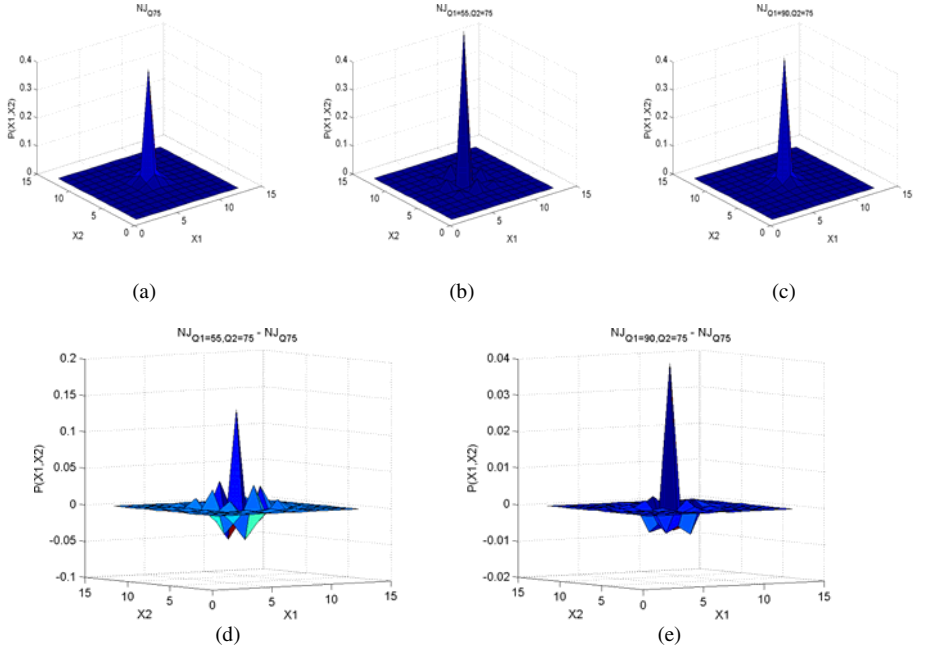


Fig. 2. Neighboring joint densities of the DCT arrays of the singly compressed JPEG image (Fig.1(a)), and the doubly compressed images (Fig.1(b) and Fig.1(c)) and the differences

2.2 Feature Mining

2.2.1 Marginal Density Features

Since manipulations such as double JPEG compression modify the marginal density of DCT coefficients at each specific frequency coordinate, it will also modify the marginal density of the absolute DCT coefficients. To reduce the number of features and speed up the detection process, we design the following marginal density features at the low frequency of the absolute DCT coefficients.

An 8×8 DCT block has 64 frequency coefficients, the frequency coordinates are paired from (1, 1) to (8, 8), corresponding to upper-left low frequency to right-bottom high frequency. Let F denote the DCT coefficient array of a JPEG image, which consists of $M \times N$ blocks, F_{ij} (i =1, 2, …, M; j = 1, 2, …, N). We extract the histogram at each location of the following coordinate pair:

$$S = \{(2, 1), (1, 2), (1, 3), (2, 2), (3, 1), (1, 4), (2, 3), (3, 2), (4, 1)\} \tag{3}$$

The feature set consists of the following probability values

$$X = \left\{ \frac{1}{MN} \left(h_{kl}(0), h_{kl}(1), h_{kl}(2), h_{kl}(3), h_{kl}(4), h_{kl}(5), \right) \middle| (k,l) \in S \right\} \tag{4}$$

Where $h_{kl}(m)$ denotes the histogram of the absolute DCT coefficient at frequency coordinate (k,l) with the value m. So there are 54 features in the marginal density set.

2.2.2 Neighboring Joint Density Features

In our algorithm, the neighboring joint features are extracted on intra-block from the DCT coefficient array and the absolute array, respectively, described as follows.

DCT Coefficient Array Based Feature Extraction

Let F denote the compressed DCT coefficient array of a JPEG image, consisting of $M \times N$ blocks F_{ij} (i = 1, 2, …, M; j = 1, 2, …, N). Each block has a size of 8×8. The intra-block neighboring joint density matrix on horizontal direction NJ_{1h} and the matrix on vertical direction NJ_{1v} are constructed as follows:

$$NJ_{1h}(x, y) = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{m=1}^{8} \sum_{n=1}^{7} \delta(c_{ijmn} = x, c_{ijm(n+1)} = y)}{56MN} \tag{5}$$

$$NJ_{1v}(x, y) = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{m=1}^{7} \sum_{n=1}^{8} \delta(c_{ijmn} = x, c_{ij(m+1)n} = y)}{56MN} \tag{6}$$

Where c_{ijmn} stands for the compressed DCT coefficient located at the m^{th} row and the n^{th} column in the block F_{ij}; $\delta = 1$ if its arguments are satisfied, otherwise $\delta = 0$; x and y are integers. For computational efficiency, we define NJ_1 as the neighboring joint density features on intra-block, calculated as follows:

$$NJ_1(x, y) = \left\{ NJ_{1h}(x, y) + NJ_{1v}(x, y) \right\} / 2 \tag{7}$$

In our experiment, the values of x and y are in the range of [−6, +6], so NJ_1 has 169 features.

Absolute DCT Coefficient Array Based Feature Extraction

Let F denote the compressed DCT coefficient array as before. The intra-block neighboring joint density matrix on horizontal direction $absNJ_{1h}$ and the matrix on vertical direction $absNJ_{1v}$ are given by:

$$absNJ_{1h}(x, y) = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{m=1}^{8}\sum_{n=1}^{7}\delta(\left|c_{ijmn}\right| = x, \left|c_{ijm(n+1)}\right| = y)}{56MN} \tag{8}$$

$$absNJ_{1v}(x, y) = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{m=1}^{7}\sum_{n=1}^{8}\delta(\left|c_{ijmn}\right| = x, \left|c_{ij(m+1)n}\right| = y)}{56MN} \tag{9}$$

We define $absNJ_1$ as the neighboring joint density features on intra-block, calculated as follows:

$$absNJ_1(x, y) = \left\{ absNJ_{1h}(x, y) + absNJ_{1v}(x, y) \right\} / 2 \tag{10}$$

In our algorithm, the values of x and y are in the range of [0, 5], so $absNJ_1$ consists of 36 features.

3 Experiments

3.1 Detection of Double JPEG Compression

The original 5150 raw images are obtained in 24-bit lossless true color and never compressed format used in our previous study of steganalysis [7, 8, 9, 12]. The single and double compressed JPEG images are generated by applying JPEG compression to these images with different quality factors. The first and second compression quality factors in the double compression are denoted "Q1" and "Q2", respectively. Table 1 shows the detection accuracy by using support vector machines (SVM) [16] for the binary classification. The results at the first row are gained by using the 324 Markov transition probability features presented in reference [5], and the results in the second row are obtained by using the integration of Marginal density features, defined in equation (4), and the neighboring joint density features, defined in equation (10), for a total of 90 features. The results show that our approach achieves the higher detection performance with respect to detection accuracy, especially in the detection of the double compression in the following: (a) Q2=40, Q1=80/85/90; (b) Q2=45, Q1=90; (c) Q2=50, Q1=85/90; (d) Q2=55, Q1=85; (e) Q2=60, Q1=90; and (f) Q2=70, Q1=90, our method outperforms the Markov approach by 11.3% to 31.5%.

3.2 Compression Quality Factor, Image Complexity, and Detection Performance

Our work in steganalysis has demonstrated that image/signal complexity is a significant parameter for the performance evaluation [8, 9, 10, 11, 12, 13] To illustrate the relationship among compression quality factor, image complexity, and the detection performance, the shape parameter β of GGD of the DCT coefficients is used to measure the image complexity [8, 9, 10, 16]. All singly and doubly compressed JPEG images are classified as low image complexity and high image complexity:

1. $\beta < 0.3$, low image complexity
2. $0.6 \leq \beta$, high image complexity

We apply SVM to the feature sets extracted from these five groups for detecting double JPEG compression. Thirty experiments are run for testing each type of feature set in each group. Average testing accuracy is compared. Henceforth we simplify the feature set of the 36 marginal features, defined in equation (4), as **Marginal**, the 324 Markov transition probability features presented in reference [5] as **Markov**, and the 169 neighboring joint density, defined in equation (7) as **NJ**. Due to the page limit, Table 4 gives the average testing detection accuracy over 30 experiments in the low image complexity (A) and high image complexity(B). In each comparison, the highest average testing accuracy is highlighted in bold.

Table 1. Average accuracy over 100 testing using Markov approach [5] (first row), marginal & neighboring joint density feature set (second row), in binary classification

Q2 \ Q1	40	45	50	55	60	65	70	75	80	85	90
40		94.6% **96.4**	97.6 **97.8**	98.1 **98.5**	98.0 **98.5**	96.8 **96.9**	93.5 **97.8**	96.4 **97.2**	59.8 **91.3**	82.4 **95.4**	63.9 **82.5**
45	96.1 **96.9**		86.6 **92.8**	96.6 **97.3**	97.3 **98.3**	97.9 **98.5**	96.8 **97.2**	94.2 **98.2**	90.6 **96.0**	88.9 **94.5**	72.9 **89.9**
50	98.6 **98.6**	91.0 **95.3**		85.5 **92.4**	97.2 **97.6**	98.3 **98.6**	97.9 **98.3**	93.0 **95.3**	96.1 **97.2**	82.4 **95.4**	53.9 **85.0**
55	**99.1** **99.1**	98.3 **98.4**	90.2 **94.7**		91.2 **95.8**	97.6 **98.4**	98.4 **98.7**	97.6 **98.1**	95.2 **97.2**	66.3 **94.5**	83.8 **94.7**
60	99.2 **99.4**	99.1 **99.1**	**98.6** 98.5	94.8 **96.9**		94.7 **97.6**	97.7 **98.6**	98.3 **98.9**	92.8 **97.0**	94.0 **97.4**	81.3 **93.0**
65	99.3 **99.6**	99.4 **99.6**	99.2 **99.3**	98.9 **99.1**	97.1 **98.1**		94.7 **97.4**	97.9 **98.6**	98.2 **98.5**	95.5 **98.5**	88.6 **94.4**
70	99.4 **99.7**	99.4 **99.7**	99.4 **99.7**	99.3 **99.5**	**99.3** 99.2	97.2 **98.1**		96.3 **97.6**	98.5 **99.0**	95.1 **97.2**	72.5 **95.5**
75	99.5 **99.8**	99.4 **99.8**	99.4 **99.8**	99.4 **99.8**	99.5 **99.7**	**99.3** **99.3**	98.2 **98.3**		97.1 **98.9**	98.6 **99.1**	94.8 **96.8**
80	99.6 **99.8**	99.6 **99.9**	99.6 **99.8**	99.5 **99.9**	99.5 **99.8**	99.5 **99.8**	99.5 **99.7**	99.0 **99.6**		97.6 **99.0**	94.7 **97.2**
85	99.6 **100.0**	99.6 **100.0**	99.6 **100.0**	99.6 **99.9**	99.7 **99.9**	99.6 **100.0**	99.6 **99.9**	99.5 **99.9**	99.4 **99.5**		98.5 **99.4**
90	99.8 **100.0**	99.8 **100.0**	99.8 **100.0**	99.8 **100**	99.8 **100.0**	99.8 **100.0**	99.7 **100.0**	99.8 **100.0**	99.9 **100.0**	99.6 **99.9**	

The results show that Marginal and NJ feature sets generally outperform Markov feature set, although the feature numbers in Marginal and NJ are less than that in Markov. It can be seen that the compression quality factors during the double JPEG compression significantly impact the detection accuracy. The detection accuracy under the condition of Q1 > Q2 is generally lower than that under Q1 < Q2.

The comparison between Table 2(A) and Table 2(B) indicates that image complexity plays a critically important role for the evaluation of detection performance. The detection accuracy in high image complexity, shown in Table 2(B), is much less than the results in Table 2(A). The detection of the double compression (Q2=40, Q1=80/85/90; Q2=45, Q1=90; Q2=50, Q1=85/90; Q2=55, Q1 =85; Q2=60, Q1=90; and Q2=70, Q1=90) by using the marginal features is not well, neither by using Markov transition probability features, nor by using neighboring joint density features. These results indicate that at the same compression factors, while image complexity increases, the detection performance deteriorates.

Table 2. Average detection accuracy over 30 experiments using **Marginal** (first row), **Markov** (second row), and **NJ** (third row) feature sets under different image complexities

(A) Detection accuracy in low image complexity ($\beta < 0.3$)

Q2 \ Q1	40	45	50	55	60	65	70	75	80	85	90
		95.9%	98.7	99.1	98.9	98.3	94.2	97.7	52.1	81.0	60.4
40		98.2	99.1	99.5	99.3	96.7	**98.4**	98.9	**92.7**	**96.7**	**82.9**
		98.2	**99.6**	**99.8**	**99.5**	**99.2**	95.5	**99.0**	64.2	83.4	62.2
	97.1		87.8	97.8	98.4	99.0	98.2	95.2	90.5	90.5	69.6
45	98.1		88.3	98.6	**99.4**	99.1	97.2	**98.5**	**98.1**	**95.2**	**89.7**
	98.5		**91.1**	**99.1**	99.3	**99.6**	**98.9**	97.7	93.9	94.8	72.2
	99.5	91.8		85.5	98.3	99.1	98.9	93.6	97.3	79.2	43.4
50	99.5	93.8		86.0	98.8	99.5	98.9	92.0	98.6	**96.4**	**92.9**
	99.7	**94.5**		**88.7**	**99.5**	**99.7**	**99.5**	**94.0**	**98.7**	90.2	51.2
	99.7	99.1	90.4		92.3	98.4	99.1	98.4	96.0	54.3	81.6
55	99.7	99.4	89.9		**96.2**	99.1	99.4	97.9	97.4	**96.0**	**94.4**
	99.8	**99.7**	**91.0**		95.8	**99.5**	**99.6**	**99.3**	**98.2**	56.3	89.0
	99.7	99.7	99.4	96.0		94.5	98.2	98.7	91.7	93.6	73.1
60	**99.9**	**99.8**	99.4	97.8		**98.6**	**99.3**	99.4	**98.2**	**98.6**	**96.9**
	99.9	**99.8**	**99.6**	**97.9**		93.8	**99.3**	**99.6**	91.0	97.0	62.0
	99.8	99.8	99.7	99.5	97.3		95.0	98.3	98.5	95.8	87.7
65	**100.0**	99.9	**99.8**	99.7	**99.1**		**98.2**	99.3	98.9	**98.7**	94.2
	99.9	**100.0**	99.8	**99.8**	97.2		97.5	**99.5**	**99.4**	98.1	**94.5**
	99.9	99.8	99.8	99.7	99.7	98.3		96.7	98.9	93.4	57.9
70	**100**	**100**	**100.0**	**99.9**	99.8	**99.1**		98.5	99.5	**98.1**	**97.8**
	99.9	99.9	**100.0**	99.8	**99.9**	99.0		**98.9**	**99.6**	93.5	65.7
	99.9	99.8	99.8	99.8	99.8	99.8	98.9		97.2	98.1	94.3
75	**100**	**100**	**100**	**100**	**100.0**	99.8	99.1		**99.1**	99.3	97.7
	99.9	99.9	99.9	99.9	99.9	**99.9**	**99.5**		98.6	**99.5**	**97.9**
	99.9	99.9	99.9	99.8	99.8	99.8	99.6	99.3		96.0	89.4
80	**100**	**100**	**100**	**100**	**100**	**100**	**99.9**	**99.7**		**99.4**	**98.4**
	99.8	99.9	99.9	99.9	99.9	**100.0**	99.7	99.4		98.9	83.0

Table 2. *(continued)*

	99.9	99.9	99.9	99.9	99.9	99.8	99.9	99.9	99.6		97.3
85	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100.0**	**99.9**		99.3
	100.0	99.9	**100**	99.9	**100.0**	**100.0**	**100.0**	**100.0**	99.6		**99.6**
	100.0	99.9	99.9	99.9	99.9	99.8	99.8	99.8	99.9	99.5	
90	**100**	**100**	**100.0**	**100**	**100**	**100**	**100**	**100**	**100**	99.9	
	100.0	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100.0**	

(B) Detection accuracy in high image complexity ($0.6 \leq \beta$)

Q1 / Q2	**40**	**45**	**50**	**55**	**60**	**65**	**70**	**75**	**80**	**85**	**90**
		59.7%	75.3	80.4	81.5	79.0	57.8	**66.4**	38.1	43.7	38.2
40		58.0	70.3	**81.0**	81.9	74.2	**70.7**	63.3	**48.5**	**55.5**	**46.7**
		63.6	**77.6**	80.1	**82.0**	**80.3**	48.3	62.2	41.9	43.2	42.2
	69.0		44.8	68.0	**76.8**	**82.1**	80.2	57.7	50.7	47.9	41.7
45	68.9		49.8	67.3	76.3	81.7	77.0	**76.0**	**55.7**	**53.7**	**48.3**
	71.3		**51.8**	**69.5**	75.8	80.9	**80.2**	54.8	47.8	51.5	45.5
	84.1	47.4		44.5	75.7	83.2	83.3	**70.8**	**66.4**	43.8	38.1
50	81.2	56.6		48.6	72.2	82.8	81.9	67.6	64.4	**54.3**	**47.7**
	82.7	**61.4**		**53.9**	**76.4**	**83.6**	**84.3**	68.5	64.3	48.2	43.6
	90.6	**82.7**	48.6		50.0	79.5	84.0	81.7	64.3	40.7	42.6
55	85.4	79.2	51.0		57.8	**81.3**	**85.5**	80.4	63.8	**52.3**	**51.0**
	84.5	80.3	**56.1**		**63.0**	77.9	85.2	**84.6**	**66.7**	44.9	46.5
	93.2	**86.9**	**86.8**	62.4		61.9	79.4	83.6	65.9	58.4	41.1
60	90.6	84.8	80.4	66.1		**69.1**	**82.5**	**85.3**	**78.9**	**64.2**	**58.1**
	90.8	85.5	78.2	**69.5**		59.4	77.2	84.1	62.5	56.4	45.9
	91.7	**93.4**	**87.1**	**86.8**	72.7		54.0	79.3	81.2	61.3	45.3
65	89.8	90.7	86.7	84.5	77.9		66.2	**79.4**	82.3	**67.9**	51.5
	88.9	89.9	85.1	84.4	63.4		65.0	78.6	**84.1**	61.2	**52.2**
	92.3	**93.6**	**95.6**	**93.7**	**89.1**	68.6		66.5	81.0	68.7	42.4
70	91.0	91.9	93.2	89.7	85.3	73.8		67.1	**84.3**	77.8	56.9
	89.7	89.6	90.8	89.9	83.4	72.5		**72.6**	80.7	64.7	46.4
	96.3	92.3	92.1	**93.6**	**94.0**	**88.4**	**77.7**		67.7	81.8	50.5
75	91.6	**92.4**	**93.6**	92.7	93.4	86.8	74.0		83.0	84.7	55.6
	89.8	89.0	88.7	89.2	89.3	85.4	74.4		70.1	83.9	**58.6**
	88.6	91.3	91.6	93.2	89.5	91.0	91.4	85.5		68.3	65.8
80	**95.5**	**94.7**	**95.0**	**95.5**	**95.2**	**94.8**	**92.9**	**92.9**		**78.8**	**79.9**
	89.1	88.1	88.8	88.1	88.5	88.1	86.2	73.3		68.4	61.1
	89.9	91.0	90.0	90.4	94.4	91.4	92.4	91.7	**90.3**		72.1
85	**97.4**	**97.4**	**97.3**	**97.4**	97.3	**97.7**	**97.7**	**98.2**	87.8		**85.5**
	90.1	90.5	91.0	90.1	89.7	93.9	90.4	89.6	83.9		81.3
	92.2	89.6	92.7	93.1	92.7	92.5	93.3	93.2	98.6	97.8	
90	**99.9**	**99.8**	**99.9**	**100.0**	**99.9**	**99.9**	**99.9**	**99.9**	**100**	**99.5**	
	98.1	98.8	97.8	98.0	98.5	98.7	97.3	97.9	96.7	97.9	

4 Conclusions

We presented a method to detect double JPEG compression based on feature mining and pattern recognition techniques. The developed features include marginal density and the neighboring joint density features on the DCT coefficients. Compared to a

recently well-developed detection method, our method is superior with respect to either detection accuracy or computational cost. Our study also shows that the detection performance is related not only to the compression quality factors but also to image complexity, which is an important parameter that seems so far to have been overlooked by the research community in conducting performance evaluation. To formally study the performance evaluation issues, both the image complexity and compression quality should therefore be included.

Acknowledgments. This project was supported in part by Award No. 2010-DN-BX-K223 awarded by the National Institute of Justice, Office of Justice Programs, U.S. Department of Justice. The opinions, findings, and conclusions or recommendations expressed in this publication/program/exhibition are those of the authors and do not necessarily reflect those of the Department of Justice. Part support from the Institute for Complex Additive Systems Analysis of the New Mexico Tech, and from the Sam Houston State University is also greatly acknowledged.

References

1. CBS News, http://www.cbsnews.com/8301-503543_162-20016679-503543.html
2. CBS News, http://www.cbsnews.com/stories/2010/09/17/world/main6876519.shtml
3. National Public Radio, http://www.npr.org/blogs/thetwo-way/2010/09/17/129938169/doctored-photograph-hosni-mubarak-al-ahram-white-house-obama-mideast-peace-talks
4. Chen, C., Shi, Y., Su, W.: A Machine Learning Based Scheme for Double JPEG Compression Detection. In: Proc. of 19th ICPR, pp. 1–4 (2008)
5. Cho, D., Bui, T.: Multivariate Statistical Modeling for Image Denoising Using Wavelet Transforms. Signal Processing: Image Communication 20, 77–89 (2005)
6. Farid, H.: Image Forgery Detection, a Survey. IEEE Signal Processing Magazine, 16–25 (March 2009)
7. Liu, Q., Sung, A.H.: Feature Mining and Nuero-fuzzy Inference System for Steganalysis of LSB Matching Steganography in Grayscale Images. In: Proc. 20th IJCAI, pp. 2808–2813 (2007)
8. Liu, Q., Sung, A.H., Chen, H., Xu, J.: Feature Mining and Pattern Classification for Steganalysis of LSB Matching Steganography in Grayscale Images. Pattern Recognition 41(1), 56–66 (2008)
9. Liu, Q., Sung, A.H., Ribeiro, B.M., Wei, M., Chen, Z., Xu, J.: Image Complexity and Feature Mining for Steganalysis of Least Significant Bit Matching Steganography. Information Sciences 178(1), 21–36 (2008)
10. Liu, Q., Sung, A.H., Qiao, M.: Novel Stream Mining for Audio Steganalysis. In: Proc. 17th ACM Multimedia, pp. 95–104 (2009)
11. Liu, Q., Sung, A.H., Qiao, M.: Derivative Based Audio Steganalysis. ACM Trans. Multimedia Computing, Communications and Applications (in press)
12. Liu, Q., Sung, A.H., Qiao, M.: Neighboring Joint Density Based JPEG Steganalysis. ACM Trans. Intelligent Systems and Technology 2(2), article 16 (2011), doi:10.1145/1899412.1899420

13. Ohm, J.R.: Multimedia Communication Technology, Representation, Transmission and Identification of Multimedia Signals. Springer, Berlin (2004)
14. Pevny, T., Fridrich, J.: Detection of Double-compression in JPEG Images for Applications in Steganography. IEEE Trans. Information Forensics and Security 3(2), 247–258 (2008)
15. Sharifi, K., Leon-Garcia, A.: Estimation of Shape Parameter for Generalized Gaussian Distributions in Subband Decompositions of Video. IEEE Trans. Circuits Syst. Video Technol. 5, 52–56 (1995)
16. Vapnik, V.: Statistical Learning Theory. John Wiley, Chichester (1998)

Semi Supervised Learning for Prediction of Prosodic Phrase Boundaries in Chinese TTS Using Conditional Random Fields

Ziping Zhao, Xirong Ma, and Weidong Pei

College of Computer and Information Engineering,
Tianjin Normal University,
Tianjin, China
gign_2001@yahoo.com.cn, {maxirong@,pweidong2004}@eyou.com

Abstract. Hierarchical prosody structure generation is a key component for a speech synthesis system. One major feature of the prosody of Mandarin Chinese speech flow is prosodic phrase grouping. In this paper we proposed an approach for prediction of Chinese prosodic phrase boundaries from a limited amount of labeled training examples and some amount of unlabeled data using conditional random fields. Some useful unlabeled data are chosen based on the assigned labels and the prediction probabilities of the current learned model. The useful unlabeled data is then exploited to improve the learning. Experiments show that the approach improves overall performance. The precision and recall ratio are improved.

Keywords: Prosodic Phrase; Text-to-speech system(TTS); Semi Supervised learning; Conditional Random Fields(CRFs).

1 Introduction

In continuous speech, native speakers tend to group words into phrases whose boundaries are marked by duration and intonational cues, and many phonological rules are constrained to operate only within such phrases, usually termed prosodic phrases. Whether the prosodic phrase boundary is properly predicted will affect the naturalness and correctness of TTS directly.

At present, a variety of studies have been done on the subject and some effective methods are put forward. For Chinese prosodic phrasing, the traditional method is based on handcrafted rules[1]. The method is easily explicable and understandable, but it is quite time consuming to get lots of trivial rules. Recently, many researchers exploited statistically-based method for this and achieved good performance. For instance, CART[2] based method is experienced recently. An HMM based statistical method for prosodic structure prediction is used in[3]. Maximum entropy (ME) model is also reported[4].

However, automatically predicting prosodic phrase boundaries with high precision and recall ratio requires a large amount of hand-annotated data, which is expensive to obtain. Meanwhile unlabeled data may be relatively easy to collect, but there has been

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 477–485, 2011.

few ways to use them. Semi-supervised learning addresses this problem by using large amount of unlabeled data, together with the labeled data, to build better classifiers.

Self-training is a commonly used technique for semi-supervised learning. Initially, an underlying classifier is trained using a small number of labeled data with all the features. Then the classifier classifies unlabeled data, and a selection metric is used to rank these classified data and to select some data that have high rankings to update the labeled training set. The procedure iterates until all the unlabeled data have been included into the training set or the maximum number of iterations is reached.

Most relevantly, semi-supervised learning in the sense of self-training has been used in natural language processing. Yarowsky uses self-training for word sense disambiguation[5]. Riloff et al. uses it to identify subjective nouns[6]. Maeireizo et al. classify dialogues as 'emotional' or 'non-emotional' with a procedure involving two classifiers[7]. Self-training has also been applied to parsing and machine translation. Rosenberg et al. apply self-training to object detection systems from images, and show the semi-supervised technique compares favorably with a state-of-the-art detector[8]. As far as we know, semi-supervised learning has not been used for prosodic phrase prediction.

The self-training method requires the underlying classifier with high performance. Recently, a probabilistic approach called Conditional Random Fields(CRFs) is proposed by Lafferty et al[9]. CRFs is a framework of discriminative method developed based on undirected graphical models and produces very good results in sequence labeling learning task. In this paper, we study the performance of self-training using CRFs, as underlying classifier. The semi-supervised learning method for CRFs utilizes two sources of information: a small amount of manually-labeled data, and a large amount of data with derived labels obtained in an unsupervised fashion. Our goal, then, is to make use of these two data sources to learn a better CRFs model. We have conducted extensive experiments to demonstrate the effectiveness of our approach.

The paper unfolds as follows. Section 2 describes CRFs model. The principle and mathematical representation of CRFs are introduced. CRFs based method to predict prosodic phrase boundaries is presented in Section 3 in detail. Section 4 gives the description of semi-supervised learning algorithm. Section 5 gives the evaluations on each method. And the experiment results and discussion are made in Section 6. Section 7 presents the conclusion and the view of future work.

2 Conditional Random Fields(CRFs)

Conditional Random Fields are undirected graphical models used to calculate the conditional probability of values on designated output nodes given values assigned to other designated input nodes. CRFs are recently introduced from of conditional model that allow the strong independence assumptions of HMMs to be relaxed, as well as overcoming the label-bias problem exhibited by MEMM[10]. This allows the specification of a single joint probability distribution over the entire label sequence given the observation sequence, rather than defining per-state distributions over the next states given the current state.

Let $X = x_1 ... x_T$ be some observed input data sequence, such as a sequence of words in training data. Let $Y = y_1 ... y_T$ be a set of finite state machine(FSM) states, each of which is associated with a label. Linear-chain CRFs thus define the conditional probability of a state sequence given an input sequence to be

$$P_\Lambda(Y \mid X) = \frac{1}{Z_X} \exp(\sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(y_{t-1}, y_t, x, t)) \tag{1}$$

where Z_X is a normalization factor over all state sequences,

$$Z_X = \sum_{y \in Y} \exp(\sum_{i=1}^{T} \sum_{k} \lambda_k f_k(y_{t-1}, y_t, x, t)) \tag{2}$$

$f_k(y_{t-1}, y_t, x, t)$ is an arbitrary feature function over its arguments. The feature function can measure any aspect of a state transition $y_{t-1} \to y_t$, and the observation sequence X, centered at the current time step t. λ_k is a learned weight for each feature function. Large positive values for λ_k indicate a preference for such an event, while large negative values make the event unlikely.

Traditional maximum entropy learning algorithms, such as GIS and IIS[11] can be used to train CRFs.

Given such a model as defined in formula 1, the most probable labeling sequence for an input X is Y^* which maximizes a posterior probability.

$$Y^* = \arg\max_{y} P_\Lambda(Y \mid X) \tag{3}$$

It can be found with dynamic programming using the Viterbi algorithm.

3 CRFs Based Method for Prediction of Prosodic Phrase Boundaries

It has been shown that Chinese utterance is also structured in a prosodic hierarchy. As proposed by Cao[9], prosodic word(PW), prosodic phrase(PP) and intonation phrase(IP) are the three prosodic units utilized in the prosodic scheme for our Mandarin speech synthesis system. These three prosodic units are in a hierarchical relation. An utterance can contain several IPs, an IP can contain several PPs, and a PP can contain several PWs respectively. The paper mainly discussed the prediction of prosodic phrases.

For automatic prediction of prosodic phrase boundaries, the sentences in training corpus are dealt with follows:

XiFang/s/B GuoJia/n/E Zai/p/B Hen/d/I Da/a/I ChengDu/n/I Shang/m/E HuShi/v/B Le/u/I FeiZhou/ns/I De/u/I ZhaiWu/n/E。 /w

(Western Countries have ignored African's debt to a large degree)

Here 'B'(Beginning) represents the beginning of a PP(prosodic phrase), 'E'(End) is the end of a PP, 'I'(Inside) represents the middle of a PP.

Thus the problem of prosodic phrase prediction can be resolved by CRFs model where the observation sequence is $X = x_1 ... x_T$, and the state sequence is a tag sequence $Y = y_1 ... y_T$ ($y \in \{B, I, E\}$).

3.1 Feature Selection in Prosodic Phrase Prediction

Feature templates are established manually from context information. For our specific application, most commonly used features include the part-of-speech(POS), the length in syllables and the word itself of the words surrounding the boundary. The neighbor words are restricted to two words before the boundary and one word after the boundary.

Besides these commonly used features, two important features are also introduced into the templates. A prosodic phrase break depends on where the last break occurs[12]. The greater the distance from the previous break, the higher the probability of a break being inserted.

For this reason, we take into consideration length measures by adding 'DTLP' and 'DTNP' into our templates for prosodic phrase prediction, which means the distance(in syllables) from current boundary to the last and next nearest PP boundary.

The features used in the model are shown in Table 1.

Table 1. Feature used in CRFs model

Feature tag	Feature explanation
W-1	previous lexicon word
W0	current lexicon word
W+1	next lexicon word
W0W+1	current lexicon word and next lexicon word
W-1W0	previous lexicon word and current lexicon word
P-1	part-of-speech of the previous lexicon word
P0	part-of-speech of the current lexicon word
P+1	part-of-speech of the next lexicon word
P0P+1	part-of-speech of the current lexicon word and next lexicon word
P-1P0	part-of-speech of the current lexicon word and previous lexicon word
WL-1	The length of the previous lexicon word, in Chinese characters
WL0	The length of the current lexicon word, in Chinese characters
WL+1	The length of the next lexicon word, in Chinese characters
WL0WL+1	The length of the current lexicon word and next lexicon word, in Chinese characters
WL-1WL0	The length of the pervious lexicon word and current lexicon word, in Chinese characters
DTLP	Distance from current position to last PP boundary
DTNP	Distance from current position to next PP boundary

We use the software CRF++[1] as our Chinese prosodic phase boundaries prediction software.

4 Semi-supervised Learning Algorithm

4.1 General Algorithm of Self-training

In self-training, an underlying classifier is first trained with a small number of labeled data which is called the initial training set. The underlying classifier is used to classify the unlabeled data. The most confident unlabeled instances with their predicted labels are added to the training set. The underlying classifier is then re-trained and the procedure repeats. The following is the general procedure of self-training algorithm.

Input: L is labeled instance set, U is unlabeled instance set, C is underlying classifier, t is the number of times of iteration, θ is the number of selected unlabeled instances for next iteration, M is the selection metric, $S(U_t, \theta, C, M)$ is the selection function, and maxIteration is the maximum number of iterations.

Initial: $t = 0$, $L_t = L, U_t = U$, where L_t and U_t are the labeled and unlabeled instance set at the t th iteration

Repeat:

train C on L_t ;

$S_t = S(U_t, \theta, C, M)$,where S_t is the selected unlabeled instance set;

$U_{t+1} = U_t - S_t$; $L_{t+1} = L_t + S_t$;

$t = t + 1$

Until: (U_t is empty) $\vee$ (maxIterations reached)

The selection function is used to rank the unlabeled instances and select a certain number of unlabeled instances to update the training instance set for the next iteration.

CRFs is selected as underlying classifier in the paper. We make use of the prediction probabilities and the labels assigned by the current learned conditional random fields model[14].Once we obtain these useful unlabeled data, we can annotate their labels and add them to the labeled training examples so as to maximize the effectiveness of learning.

5 Experiment

5.1 The Experiment Corpus

In our experiments, a speech corpus for training and testing are used. 10000 sentences are randomly selected from the People's Daily corpus read by a radiobroadcaster. The sentences with three-level prosodic boundaries are labeled manually by listening to the record speech.

[1] http://crfpp.sourceforge.net/

To check consistency of annotation across different people, an exploratory experiment was carried out. Three annotators were first trained on the same 100 sentences. At this stage, they were required to discuss criteria for annotation so that they could achieve agreement on most of the annotations in the 100 sentences. Then they were asked to annotate a small subset of the corpus. All three annotators achieved agreement on 85%. That is to say pretty good consistency existed among the three annotators.

The sentences of the corpus are also processed with a text analyzer, where Chinese word segmentation and part-of-speech tagging are accomplished in one step using a statistical language model. The segmentation and tagging yields a gross accuracy rate over 96.5%.

We randomly hide the labels of 500 sentences to make the unlabeled set, and keep the remaining portion as the labeled set which is initially used to train the underlying classifier and evaluation.

5.2 The Evaluation Criteria

The precision, recall ratio and F-score are adopted as the evaluation criteria. The precision and recall are defined as: Pre= C_1 / C_2 , Rec= C_1 / C_3 . C_1 is the number of prosodic phrase boundaries correctly recognized, C_2 is the total number of prosodic phrase boundaries recognized, and C_3 represents the total number of real prosodic phrase boundaries in the test corpus.

The F-score is calculated as: $F = 2 \times \mathrm{Pr}e \times \mathrm{Re}c / (\mathrm{Pr}e + \mathrm{Re}c)$.

6 Results and Discussion

In the first experiment, we attempt to illustrate the effectiveness of our approach for choosing useful unlabeled data.

We first randomly selected these 5000 sentences to train a CRFs model and apply the trained model to automatically label the unlabeled data. The automatically labeled data is then sorted in ascending order of the probability and divided into 5 equal portions. The performance for each portion is measured. Table 2 shows the results in each portion and the overall performance. It shows that the performance increases according to the prediction probability. The useful unlabeled data refers to those data in which the model labels do not match with the actual labels. Hence it is likely that the 1st portion contains the highest number of useful unlabeled instances.

Next, we intend to show that the useful unlabeled data can effectively improve the performance if they are added to the training set with their actual labels. For each portion, we make use of the manual labels of the sequences and added these manually labeled sequences to the training set.

A CRFs model is then trained with this new training set and applied to automatically label the sequences in the other nine portions. Table 3 shows the performance for the original labeled training set after adding unlabeled data achieved on the same testing set. It shows that the portions with lower prediction probability

show the greater improvement. This suggests that the unlabeled data with low prediction probability are more useful than those with high prediction probability.

Two factors may influence the performance of self-training. One is the size of the initial labeled instance set, and the other one is the number of classified unlabeled data selected for the next iteration. A set of experiments is developed for self-training with different sizes of initial labeled instance sets. Then, we design the experiments of self-training with different numbers of selected unlabeled data for the next iteration. The experimental results of F-sore are shown in Table 4 and Table 5.

In Table 6, the results of F-score on the semi-supervised learning method for CRFs are compared with the results from baseline when adding the same labeled set. The initial labeled set of the two methods are both 1000. The results from baseline are obtained from the corresponding supervised classifiers. From the results, we can see that the performance of self-training is better than the baseline.

6.1 Results

Table 2. The result for using CRFs trained by original labeled data to label the unlabeled data

Portion	Precision(%)	Recall(%)	F-score(%)
1	82.8	68.4	74.9
2	83.1	70.1	75.9
3	85.1	71.2	77.5
4	87.0	75.0	80.6
5	89.5	81.2	85.1
Average	85.5	73.2	78.9

Table 3. The result for the original labeled training set after adding unlabeled data

Portion	Precision(%)	Recall(%)	F-score(%)
1	84.5	70.7	77.0
2	83.4	68.0	74.9
3	82.1	67.3	74.0
4	81.3	66.5	73.2
5	80.1	66.0	72.4

Table 4. The result of different sizes of initial labeled data

Initial labeled data	F-score(%)
1000	70.7
1200	71.2
1400	71.8
1600	71.0
1800	70.4
2000	72.0

Table 5. The result of different numbers of selected unlabeled data for the next iteration

Num of unlabeled data for next iteration	F-score(%)
50	71.7
100	71.8
200	71.4
300	71.9
400	72.2

Table 6. The result of self-training, baseline

	10	30	50	100	200
Self-training	71.7	72.0	71.5	72.4	73.7
baseline	68.7	69.4	70.0	70.5	70.3

7 Conclusion

In this paper, we introduce a semi-supervised learning method, self-training, to solve the task of prosodic phrase prediction. The results also show that self-training can achieve comparable performance to the supervised learning models for prosodic phrase prediction.

Our future work is to incorporate more contextual information into the models. We will extend our study of self-training to other applications of machine learning.

Acknowledgements

The work described in this paper was substantially supported from the National Science Foundation of China (Grant No: 60970060), Doctor Foundation of Tianjin Normal University of China(Grant No: 52X09012), the Open Project of Shanghai Key Laboratory of Trustworthy Computing of China under Grant No.53H10058 and the Technology Fund Planning Project of Higher Education, Tianjin(Grant No:20080801).

References

1. Niu, Z., Chai, P.: Segmentation of Prosodic Phrases for Improving the Naturalness of Synthesized Mandarin Chinese Speech. In: ICSLP 2000 Conference, Beijing, China, pp. 350–353 (2000)
2. Yao, Q., Chu, M., Hu, P.: Segmenting unrestricted Chinese text into prosodic words instead of lexical words. In: ICASSP 2001 Conference, Salt Lake City, pp. 825–828 (2001)
3. Veilleux, N.M., Ostendorf, M., Price, P.J., Shattuck-Hufnagel, S.: Markov Modeling of prosodic phrase structure. In: ICASSP 1990, New Mexico, USA, pp. 777–780 (1990)

4. Li, J., Hu, G., Wang, R.: Chinese prosody phrase prediction based on maximum entropy model. In: Interspeech 2004, Jeju Island, Korea, pp. 729–732 (2004)
5. Yarowsky, D.: Unsupervised word sense disambiguation rivaling supervised methods. In: 33rd Annual Meeting of the Association for Computational Linguistics, USA, pp. 189–196 (1995)
6. Riloff, E., Wiebe, J., Wilson, T.: Learning subjective nouns using extraction pattern bootstrapping. In: 7th Conference on Natural Language Learning (CoNLL 2003), Canada, pp. 25–32 (2003)
7. Maeireizo, B., Litman, D., Hwa, R.: Co-training for predicting emotions with spoken dialogue data. In: 42nd Annual Meeting of the Association for Computational Linguistics (ACL), Spain (2004)
8. Rosenberg, C., Hebert, M., Schneiderman, H.: Semi-supervised self-training of object detection models. In: 7th IEEE Workshop on Applications of Computer Vision 2005, USA, pp. 29–36 (2005)
9. Lafferty, J., McCallum, A., Pereiram, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: 18th International Conference on Machine Learning, USA, pp. 282–289 (2001)
10. McCallum, A., Freitag, D., Pereira, F.: Maximum Entropy Markov Models for Information Extraction and Segmentation. In: ICML 2000, USA, pp. 591–598 (2000)
11. della Pietra, S., della Pietra, V., Lafferty, J.: Inducing Features of Random Fields. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(4), 380–393 (1997)
12. Sanders, E., Taylor, P.: Using statistical models to predict phrase boundaries for speech synthesis. In: 4th European Conference on Speech Communication and Technology, Spain, pp.19–25 (1995)
13. Wong, T.-L., Lam, W.: Semi-Supervised learning for sequence labeling using conditional random fields. In: Proceeding of 4th International Conference on Machine Learning and Cybernetics, China, pp. 2832–2837 (2005)

Singer Identification Using Time-Frequency Audio Feature

Pafan Doungpaisan

Department of Information Technology, Faculty of Industrial Technology and Management, King Mongkut's University of Technology North Bangkok, 1518 Pibulsongkram Road, Bangsue, Bangkok 10800, Thailand
pafan@kmutnb.ac.th

Abstract. Singer identification is a difficult topic in music information Retriveal research area. Because the background instrumental accompaniment in audio music is regarded as noise source that has to reduce a performance.

This paper proposes a singer identification algorithm thai is able to automatically identify a singer in an audio music signal with background music by using Time-Frequency audio feature. The main idea is used a spectrogram to able effective Time-Frequency feature and used as the input for classification. The proposed technique is test with 20 different singer. Sereval classification technique are compared,such as Feed-Forward Neural Network, k-Nearest Neighbor (kNN) and Minimum least square linear classifier(Fisher). The experimental result on singer identification using a spectrogram with Feed-Forward Neural Networkand and k-Nearest Neighbor (kNN) can effectively identify the singer in music signal with background music more than 92%.

Keywords: Spectrogram, Time-Frequency audio feature, Singer identification, Feed-Forward Neural Network, k-Nearest Neighbor (kNN).

1 Introduction

With digital music becoming more popular such as music CDs and MP3 music downloadable from the internet, music databases, are growing rapidly. Technologies are demanded for efficient retrieval of these music collections, so that consumers can be provided with powerful functions for browsing and searching musical content. Among such technologies, is the automatic singer identification of a song, i.e. to recognize the singer of a song by analyzing audio features of the music signal. With this capability provided in a music system, the user can easily get to know the singer's information of an arbitrary song, or retrieve all songs performed by a particular singer in a distributed music database. Furthermore, this technology may be used to classify songs of similar voices of singers in a music collection, or search for songs which are similar to a query song in terms of the singer's voice.

Several techniques are proposed for the algorithm to solve the problem of audio classification [3] [4] [5] [6] [7]. Most of the proposed methods are divided into

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 486–495, 2011.

two processing steps: feature extraction and classification. In the first step, feature exaction, the redundant information contained in the signal are transformed into descriptors that are used to be the input of classifier. A variety of feature extraction techniques are applied such as power pattern and frequency pattern [8], short-time Fourier transform [3], continuous wavelet transform [9] and Mel-frequency cepstral coefficients (MFCCs) [10]. However, only few temporal-domain features have been developed to identification singer. In the second step, classification , the singing voice is recognized. Several classification techniques are applied such as HMM-base method [11][12], multilayered neural network [13], Limited receptive area (LIRA) neural classifier [3][9]. Support vector machine (SVM) [14][15].

Whitman [1] presents earliest work on artist identification for MIR. In this work, ANN and SVM classifiers are applied to spectral features computed from short clips of popular music by a variety of artists. The audio clips are one second long, and are analyzed using, alternately, the Discrete Fourier Transform (DFT) and MFCCs. For a five-artist set, the best-case testing-set classification accuracy is found to be 91%. Increasing to ten artists lowers this accuracy figure to 70%, while a twenty-one-artist set yields only 50% accuracy in the best case.

Another system, which is evaluated on the same data as the system in [1], is presented by Berenzweig et al. [2]. This system is used MFCC's as features input to another neural network classifier. On the 21-artist data set, the use of the vocal identification preprocessing improves classification accuracy to 65%. Overall, the results suggest that using the spectra of audio segments that contain vocals improves performance.

Kim [16] presents a system that is similar Berenzweig et al. [2], but it specifically claims to perform singer identification rather than artist identification. In this system, a armonicity estimate is compared against a threshold to identify segments of audio containing vocals. Two classifiers, one using Gaussian mixture models and another using SVMs, are tested using warped linear prediction coefficients. Best case performance on a set of 17 artists is found to be approximately 45%.

Hiromasa Fujihara [17], propose a method that can reduce the negative influence of accompaniment sounds directly from a given musical audio signal to solve this problem. This method consists of the following four parts: accompaniment sound reduction, feature extraction, reliable frame selection, and stochastic modeling. To reduce the negative influence of accompaniment sounds, the accompaniment sound reduction part first segregates used fundamental frequency (F0) and resynthesizes the singing voice from polyphonic audio signals on the basis of its harmonic structure. The feature extraction part then calculates the feature vectors from the segregated singing voice. The reliable frame selection part chooses reliable vocal regions from the feature vectors and removes unreliable regions that do not contain vocals or are greatly influenced by accompaniment sounds. The stochastic modeling part represents the selected features as parameters of the Gaussian mixture model (GMM).

An interested word proposed by Peerapol[19], propose a method that can recognize the word in a singing signal with background music by using the concept of spectrogram pattern matching. The main idea is to apply both the spectrogram for feature extraction to solve the problem of singing voice recognition. His technique, Each signal that accompanies music is analyzed and generated to its spectrogram that is used to train data for the classifier. Several classification functions are compared , such as Fisher classifier, Feed-Forward can effectively recognize the word in music with the accuracy rate more than 84%.

The object of this paper is to solve the problem singer identification in audio that accompanying music without using any method to separate a music instrumental in background. Especially, The instrumental interference is regard as a noise source degarding the performance of identification performance. We used the idea of recognize audio signal in [19] to solve the problem singer identification. Fig 1 show a diagram of our proposed algorithm. First the audio signal is divided into a short segment. Fast Fourier transform (FFT) is applied to each segment to generate a spectrogram. After that a classification technique was used such as Feed-Forward Neural Network, k-Nearest Neighbor (kNN) and Minimum least square linear classifier(Fisher). The K-fold cross-validation technique is used to evaluate the performance of classifier.

The rest of the paper is organized as follows. The detail of our proposed algorithm is described in Section 2. The experimental results are showed in Section 3. Section 4 concludes paper.

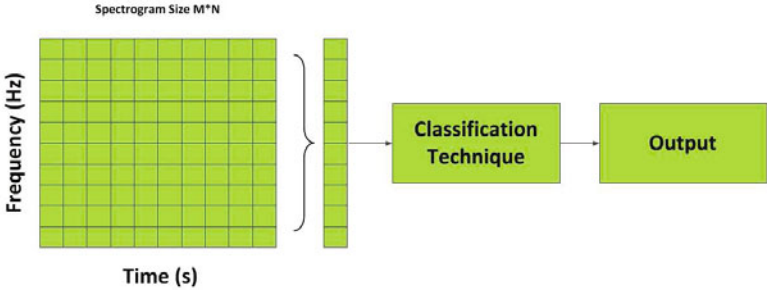


Fig. 1. Overview of the proposed algorithm

2 Methodology

2.1 Data Collection

The database used in this experiment contains sixteen Thai popular singers (14 men singer, 2 women singer), 10 music for each singer, total 160 songs All soungs collected from a commercial music covering five genres. The five genres are Rock, Soft-Rock, Pop, Acoustic, R&B. We capture by manual 20s vocal area that accompany music in each song to generate a dataset. All songs were coded in stereo of frequency 44.2 kHz with 128/s bit rate. The files were converted in mono and down sampling to 16 kHz.

2.2 Feature Extraction

There are many features that can be used to characterize audio signals. Feature extraction is the process of computing a compact numerical representation that can be used to characterize segments of audio signal. The present work uses spectrogram analysis based on Fast Fourier transform (FFT) for feature extraction. Figure 2 shows the block diagram of spectrograms generated by Fourier transformation.

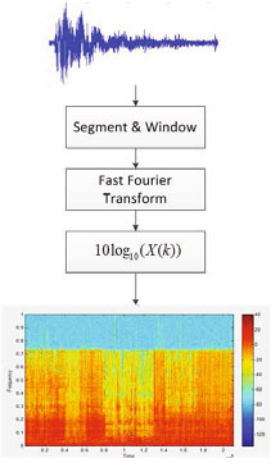


Fig. 2. Block diagram of spectrograms generated by Fourier transformation

First the audio signal is divided into short time windows. Fast Fourier transform (FFT) is applied to each time window for the discrete-time signal $x(n)$ with length N and N must be sized equal power of two, given by

$$X(k) = \sum_{n=0}^{N-1} w(n)x(n)exp(-\frac{j2\pi kn}{N}) \tag{1}$$

for $k = 0, 1, \ldots, N-1$, where k corresponds to the frequency $f(k) = (\frac{kf_s}{N})$, f_s is the sampling frequency in Hertz and $w(n)$ is a time-window. Here, we chose Hamming window as a time window, given by

$$w(n) = 0.54 - 0.46\cos\left(\frac{\pi n}{N}\right) \tag{2}$$

In this paper, each segment is transformed with DFT or FFT in (1). After that the magnitude frequency vectors are stacked and plotted with the vertical axis representing the frequency and the horizontal axis representing the time. In this paper, we used each column of the spectrogram as a feature vector for

classification the signal. The spectrogram displays just the energy and not the phase of the short-term Fourier transform, we compute the energy $p(n)$ as

$$p(k) = 10log_{10}(X(k)) \tag{3}$$

2.3 Data Classification

This section provides information regarding the classification methods and parameters used. Three different classifiers, namely Minimum least square linear classifier(Fisher), K-nearest neighbors (KNN), Feed-Forward neural network [17] [18] are used. Parameters for each classification technique are shown in Table 1.

Table 1. Parameters for Each Classification Technique

Classifiers	**Parameters**
Feed-Forward NN	Hidden neurons = 20, 30 and 40 No. of iterations = 1000
K-Nearest Neighbor	K=5, 10 and 5
Fisher	Default from PRTools.

K-Nearest Neighbor and Minimum least square linear classifier(Fisher) were implemented by using the PRTools Matlab package [19]. Feed-Forward Neural Networks were implemented by using the Neural Network Toolbox Matlab package [18]. The performances were evaluated by 5-fold cross-validation technique.

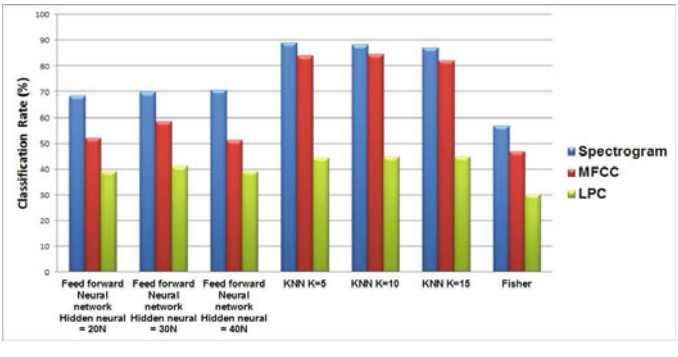


Fig. 3. Test classification performance of different classifiers (x axis) using a spectrogram, MFCC and LPC with a window of 512

3 Result

In each experiment, we performed 50 runs of the 5-fold cross-validation to obtain statistically reliable results. The mean recognition rate was calculated based on the error average for one run on test set.

First, We created a spectrogram used a window of 512. The experiment is compared with Mel-frequency cepstral coefficients (MFCCs) 13 coeffs and Linear predictive coding (LPC) 13th-order. MFCCs and LPC was used a window of 512 same a spectrogram. Figure 3 shows the test classification performance of different classifiers (x axis) . Experiments by using K-Nearest Neighbor with 5, 10 and 15 Neighbor with a spectrogram show the best performance of 89.13%, 88.47% and 87.44%.

A spectrogram can be obtained from different sizes of windowed segment. We wanted to find out the size of windowed segment that gives the best accuracy rate in classification for the data set. The following window sizes were experimented: 4096, 2048, 1024, 512, 256, 128 and 64 Figures 7, 8, 9, 10, 11, 12, and 13 show the

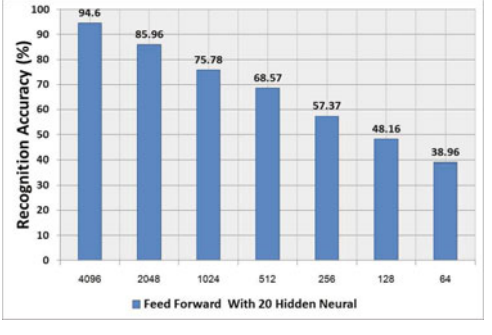


Fig. 4. Test classification performance of different sizes of windowed segment (x axis) using Feed forward Neuron network 20, hidden neuron

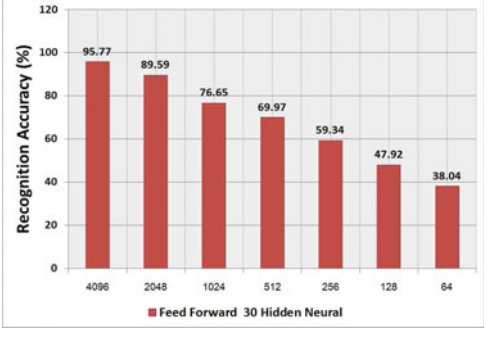


Fig. 5. Test classification performance of different sizes of windowed segment (x axis) using Feed forward Neuron network 30, hidden neuron

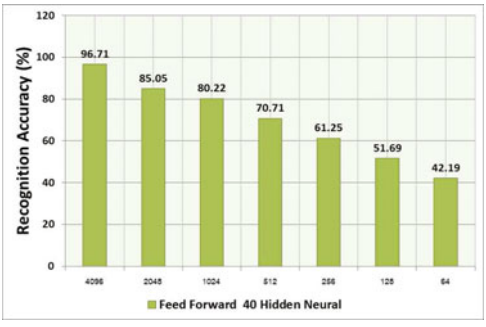


Fig. 6. Test classification performance of different sizes of windowed segment (x axis) using Feed forward Neuron network 40, hidden neuron

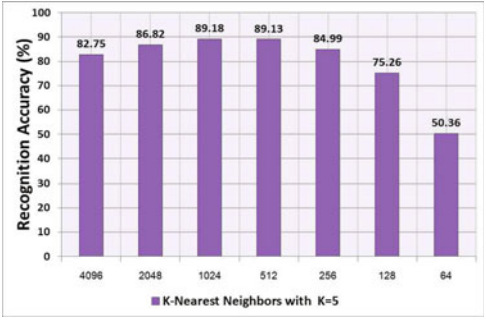


Fig. 7. Test classification performance of different sizes of windowed segment (x axis) using K-Nearest neighbor classifier with 5 neighbor

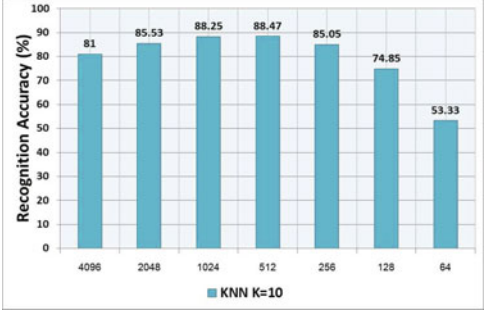


Fig. 8. Test classification performance of different sizes of windowed segment (x axis) using K-Nearest neighbor classifier with 10 neighbor

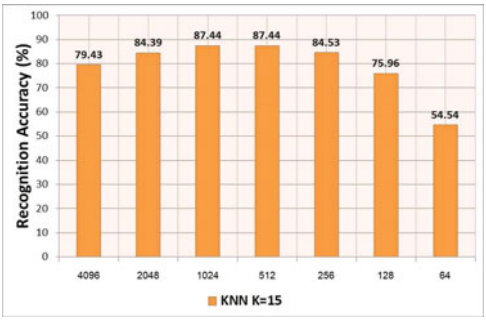


Fig. 9. Test classification performance of different sizes of windowed segment (x axis) using K-Nearest neighbor classifier with 15 neighbor

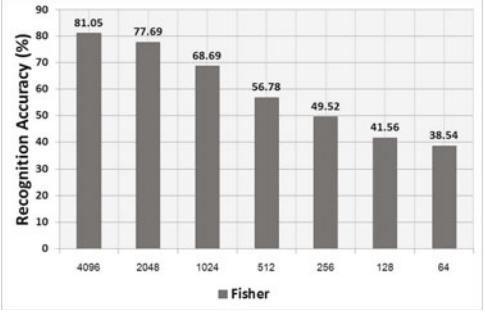


Fig. 10. Test classification performance of different sizes of windowed segment (x axis) using using Fisher classifier

Table 2. AVERAGE ACCURACY OF ALL CLASSES FROM EACH METHOD. KNN(5) , KNN(10) and KNN(15) refer to KNN with K =5, KNN with K =10 and KNN with K =15, respectively. FFNet(20), FFNet(30) and FFNet(40) refer to feed forward networks with 20, feed forward networks with 30 and feed forward networks with 40 hidden neuron, respectively.

Window Size	4096	2048	1024	512	256	128	64
KNN (5)	82.75%	86.82%	89.18%	89.13%	84.99%	75.26%	50.36%
KNN (10)	81.00%	85.53%	88.25%	88.47%	85.05%	74.85%	53.33%
KNN (15)	79.43%	84.39%	87.44%	87.44%	84.53%	75.96%	54.54%
FFNet (20)	94.6%	85.96%	75.78%	68.75%	57.37%	48.16%	38.96%
FFNet (30)	95.77%	89.59%	76.65%	69.97%	59.38%	47.92%	38.04%
FFNet (40)	96.71%	85.05%	80.22%	70.71%	61.25%	51.69%	42.19%
Fisher	81.05%	77.69%	68.69%	56.78%	49.52%	41.56%	38.54%

testing accuracy of different classifiers with different sizes of windowed segment (x axis). Table 2 summarized the average accuracy of all classes from each method.

From these results, a feed forward networks with 20, 30 and 40 hidden neuron performed on a large size of windowed segment gives better recognition accuracy than using a small windowed segment. By using Feed forward Neural network with 40 Hidden neural performed with a window of 4096 received the highest performance when compared with the other up to 96.71%. In the case of K-Nearest neighbor the best performance of windowed segment are between 1024 and 512. For every classifier, The use of small windows that are received the very low performance.

4 Conclusion

In this paper, we propose an algorithm for singer identification without using separate method music in background in polyphonic music based on spectrogram and classification technique. This approach is simpler than the existing methods. The results show all classifiers can identify a singger. In particular, Feed-Forward neural network with 40 hidden neural performed with a window of 4096 give the best results and K-Nearest neighbor the best performance of windowed segment are between 1024 and 512. A spectrogram created by using large windowed segment gives better recognition rate than a spectrogram created by using small windowed segment.

References

1. Whitman, B., Flake, G., Lawrence, S.: Artist detection in music with Minnow match. In: Proceedings of the 2001 IEEE Workshop on Neural Networks for Signal Processing, Falmouth, MA, pp. 559–568 (2001)
2. Berenzweig, A., Ellis, D., Lawrence, S.: Using voice segments to improve artist classification of music. In: AES 22nd International Conference, Espoo, Finland (2002)
3. Makeyev, O., Sazonov, E., Schuckers, S., Melanson, E., Neuman, M.: Limited receptive area neural classifier for recognition of swallowing sounds using short-time Fourier transform. In: Proc. International Joint Conference on Neural Networks IJCNN 2007, Orlando, USA, August 12-17, pp. 1417.1–1417.6 (2007)
4. Lin, C.-C., Chen, S.-H., Truong, T.-K., Chang, Y.: Audio Classification and Categorization Based on Wavelets and Support Vector Machine. IEEE Transactions on Speech and Audio Processing 13(5), 644–651 (2005)
5. Esmaili, S., Krishnan, S., Raahemifar, K.: Content Based Audio Classification and Retrieval Using Joint Time-Frequency Analysis. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2004), vol. 5, pp. 665–668 (May 2004)
6. Wang, J.-C., Lee, H.-P., Wang, J.-F., Lin, C.-B.: Robust environmental sound recognition for home automation. IEEE Transactions on Automation Science and Engineering 5(1), 25–31 (2008)

7. Yoshii, K., Goto, M., Okuno, H.G.: Rum Sound Recognition for Polyphonic Audio Signals by Adaptation and Matching of Spectrogram Templates With Harmonic Structure Suppression. IEEE Transactions on Audio, Speech, and Language Processing 15(1), 333–345 (2007)
8. Toyoda, Y., Huang, J., Ding, S., Liu, Y.: Environmental sound recognition by the instantaneous spectrum combined with the time pattern of power. In: Proceedings of the 2nd IASTED International Conference on Neural Networks and Computational Intelligence, pp. 169–172 (2004)
9. Makeyev, O., Sazonov, E., Schuckers, S., Melanson, E., Neuman, M.: Limited receptive area neural classifier for recognition of swallowing sounds using short-time Fourier transform. In: International Joint Conference on Neural Networks, IJCNN 2007, Orlando, USA, August 12-17, pp. 1417.1–1417.6 (2007a)
10. Makeyev, O., Sazonov, E., Schuckers, S., Lopez-Meyer, P., Melanson, E., Neuman, M.: Limited receptive area neural classifier for recognition of swallowing sounds using continuous wavelet transform. In: 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS 2007, pp. 3128–3131 (August 2007)
11. Ajmera, J., McCowan, I., Bourlard, H.: Speech/music segmentation using entropy and dynamism features in a HMM classification framework. Speech Communication 40(3), 351–363 (2003)
12. Rajapakse, M., Wyse, L.: Generic audio classification using a hybrid model based on GMMS and HMMS. Proceedings of the IEEE, 1550–1555 (2005)
13. Toyoda, Y., Huang, J., Ding, S., Liu, Y.: Environmental Sound Recognition by Multilayered Neural Networks. In: CIT 2004, pp. 123–127 (2004)
14. Georgoulas, G., Georgopoulos, V.C., Stylios, C.D.: Speech Sound Classification and Detection of Articulation Disorders with Support Vector Machines and Wavelets. In: 28th IEEE EMBS Annual International Conference, New York City, New York, USA, August 30-September 3 (2006)
15. Lin, C.-C., Chen, S.-H., Truong, T.-K., Chang, Y.: Audio classification and categorization based on wavelets and support vector machine. IEEE Transactions on Speech and Audio Processing 13(5), 644–651 (2005)
16. Kim, Y.E., Whitman, B.: Singer identification in popular music recordings using voice coding features. In: Proceedings of the 3rd International Conference on Music Information Retrieval, Paris, France (2002)
17. Fujihara, H., Goto, M., Kitahara, T., Okuno, H.G.: A Modeling of Singing Voice Robust to Accompaniment Sounds and Its Application to Singer Identification and Vocal-Timbre-Similarity-Based Music Information Retrieval. IEEE Transactions on Audio, Speech, and Language Processing 18(3), 638–648 (2010)
18. Demuth, H., Beale, M.: Neural Network Toolbox for Use with Matlab: User's Guide (version 4), p. 2000. The MathWorks Inc.
19. Peerapol, K.: Singing Voice Recognition based on Matching of Spectrogram Pattern. In: Proceedings of International Joint Conference on Neural Networks, Atlanta, Georgia, USA, June 14-19, pp. 978–981 (2009)

Robust Multi-stream Keyword and Non-linguistic Vocalization Detection for Computationally Intelligent Virtual Agents

Martin Wöllmer[1], Erik Marchi[2], Stefano Squartini[2], and Björn Schuller[1]

[1] Institute for Human-Machine Communication,
Technische Universität München, 80333 München, Germany
[2] 3MediaLabs - A3LAB, DIBET - Dipartimento di Ingegneria Biomedica,
Elettronica e Telecomunicazioni, Università Politecnica delle Marche,
60131 Ancona, Italy
{woellmer,schuller}@tum.de, s.squartini@univpm.it

Abstract. Systems for keyword and non-linguistic vocalization detection in conversational agent applications need to be robust with respect to background noise and different speaking styles. Focussing on the Sensitive Artificial Listener (SAL) scenario which involves spontaneous, emotionally colored speech, this paper proposes a multi-stream model that applies the principle of Long Short-Term Memory to generate context-sensitive phoneme predictions which can be used for keyword detection. Further, we investigate the incorporation of noisy training material in order to create noise robust acoustic models. We show that both strategies can improve recognition performance when evaluated on spontaneous human-machine conversations as contained in the SEMAINE database.

Keywords: Conversational agents, keyword spotting, multi-condition training, long short-term memory.

1 Introduction

Systems for advanced Human-Machine Interaction which offer natural and intuitive input and output modalities require robust and efficient machine learning techniques in order to enable spontaneous conversations with a human user. Since speech is the most natural human-to-human communication channel, the advancement of speech technology is an essential precondition for improving Human-Machine Interaction. Conversational agents which shall recognize, interpret, and react to human speech rely on speech processing technologies that can cope with various challenging conditions, such as background noise, disfluencies, and emotional coloring of speech. Reliably extracting meaningful keywords tends to be the most important functionality of speech processing modules providing linguistic information to the dialogue management [1].

As conversational agents are often used in noisy conditions, automatic speech recognition (ASR) and keyword spotting systems have to be based on features and models that lead to an acceptable recognition performance even if the speech

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 496–505, 2011.

signal is superposed by background noise. Thus, most systems apply speech feature normalization or enhancement techniques such as cepstral mean normalization, histogram equalization, or Switching Linear Dynamic Models [2]. A simple and efficient method to improve the noise robustness of the speech recognition back-end is to use matched or multi-condition training strategies [3] by incorporating noisy training material which reflects the noise conditions expected while running the system.

Another approach to enhance recognition performance in challenging conditions is to apply neural networks for generating state posteriors or phoneme predictions which are then decoded by a Hidden Markov Model (HMM). These so-called Tandem or hybrid systems are a popular alternative to the conventional HMM technique since they efficiently combine the advantages of both, neural networks and HMMs [4]. However, conventional Multilayer Perceptrons (MLP) or recurrent neural networks (RNN) as they are used in today's Tandem ASR systems have some inherent drawbacks such as the *vanishing gradient problem* [5] which limits the amount of contextual information that can be modeled by an RNN. Yet, due to co-articulation effects in human speech, context modeling is essential for accurate phoneme prediction. As an alternative to learning a fixed amount of context by processing a predefined number of consecutive feature frames via MLPs, the usage of Long Short-Term Memory (LSTM) networks [6] has recently been proposed for keyword spotting [7] and continuous ASR systems [8]. LSTM networks are able to model a self-learned amount of context information which leads to higher phoneme recognition accuracies when compared to standard RNNs [8].

In this contribution we investigate both, multi-condition training strategies for enhanced keyword spotting performance in noisy conditions, and the effect of incorporating LSTM phoneme prediction in a multi-stream ASR framework. Both techniques are evaluated with respect to their suitability for conversational agents. Thereby we focus on the *Sensitive Artificial Listener* (SAL) scenario which aims at maintaining a natural conversation with different virtual characters [9].

Section 2 describes the four virtual SAL characters that allow for emotional human-machine conversations via the SEMAINE system[1]. For our keyword spotting experiments we use spontaneous speech as contained in the SEMAINE database which is introduced in Section 3 and provides training material for the SEMAINE system. The multi-stream LSTM-HMM technique used for enhanced keyword and non-linguistic vocalization detection within the SEMAINE system is outlined in Section 4. Finally, Section 5 contains the results of our multi-condition training and multi-stream decoding experiments.

2 Sensitive Artificial Listeners

In contrast to most task-oriented dialogue systems, the *Sensitive Artificial Listeners* representing the SEMAINE system [9] focus on aspects of communication that are emotion-related and non-verbal. The system is designed for a one-to-one dialogue situation in which one user is conversing with one of four available

[1] http://semaine-project.eu/

virtual agent characters. Besides speech, the (multimodal) interaction involves head movements and facial expressions. The SAL characters have to recognize a limited set of emotionally relevant keywords, non-linguistic vocalizations such as *laughing* or *sighing*, and the prosody with which the words are spoken. Based on the interpreted input from audio and video, the system has to show appropriate listener behavior, e. g., multimodal *backchannels*, decide when to *take the turn*, and select a suitable phrase in order to maintain the conversation.

The four SAL characters roughly represent areas in the *arousal-valence* space: 'Spike' is angry (high arousal, low valence), 'Poppy' is happy (high arousal, high valence), 'Obadiah' is sad (low arousal, low valence), and 'Prudence' is matter-of-fact (moderate arousal, moderate valence). During the conversations, the virtual characters aim to induce an emotional state in the user that corresponds to *their* typical emotional state.

3 The SEMAINE Database

The SEMAINE database was recorded in order to provide training material for the speech and vision-based input components of the SEMAINE system. For this purpose, the functionality of the virtual agent system was imitated by a human operator using a Wizard-of-Oz scenario. Thus, users were encouraged to show emotions while naturally speaking about arbitrary topics.

The transcribed part of the database consists of 19 recordings with different English speaking users and has a total length of 6.2 h. Models used for the experiments in Section 5 are trained on recordings 1 to 10 (speech material from both, user and operator) and tested on recordings 11 to 19 (only speech from the user). The vocabulary size of the SEMAINE corpus is 3.4 k.

In addition to the SEMAINE database, two other spontaneous speech corpora were used for acoustic and language model training: the SAL corpus and the COSINE corpus. The SAL database was recorded under similar conditions as the SEMAINE corpus, which makes it well-suited for our application scenario. It has already been used in a large number of studies on emotional speech (for more details on the SAL database, see [10], for example). The COSINE corpus [11] contains multi-party conversations recorded in real world environments and is partly overlaid with indoor and outdoor noise sources. It consists of ten transcribed sessions with 11.4 h of speech from 37 different speakers and has a vocabulary size of 4.8 k.

4 Multi-stream LSTM-HMM

This section briefly outlines the multi-stream LSTM-HMM ASR system we use for enhanced keyword detection in emotionally colored speech (see Section 5.2). The main idea of this technique is to enable improved recognition accuracies by incorporating context-sensitive phoneme predictions generated by a Long Short-Term Memory network into the speech decoding process.

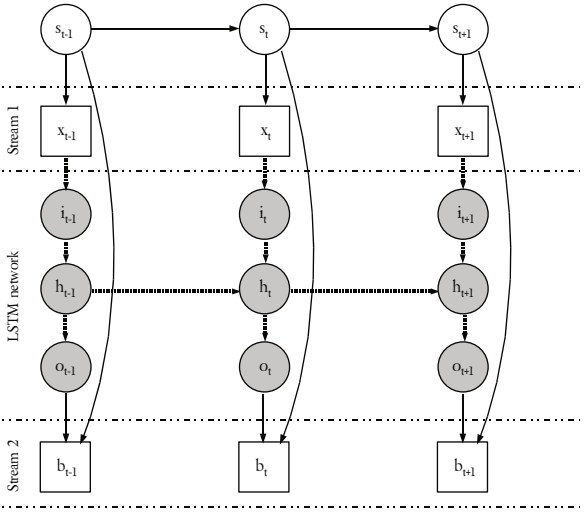


Fig. 1. Architecture of the multi-stream LSTM-HMM decoder: s_t: HMM state, x_t: acoustic feature vector, b_t: LSTM phoneme prediction feature, i_t, o_t, h_t: input, output, and hidden nodes of the LSTM network

LSTM networks [6] were introduced after the analysis of the error flow in conventional recurrent neural nets revealed that long range context is inaccessible to standard RNNs, since the backpropagated error either blows up or decays over time (vanishing gradient problem [5]). The LSTM principle is able to overcome the vanishing gradient problem and allows the network to learn the optimal amount of contextual information relevant for the classification task.

An LSTM layer is composed of recurrently connected memory blocks, each of which contains one or more memory cells, along with three multiplicative 'gate' units: the input, output, and forget gates. The gates perform functions analogous to read, write, and reset operations. More specifically, the cell input is multiplied by the activation of the input gate, the cell output by that of the output gate, and the previous cell values by the forget gate. The overall effect is to allow the network to store and retrieve information over long periods of time.

The structure of our multi-stream decoder can be seen in Figure 1: s_t and x_t represent the HMM state and the acoustic (MFCC) feature vector, respectively, while b_t corresponds to the discrete phoneme prediction of the LSTM network (shaded nodes). Squares denote observed nodes and white circles represent hidden nodes. In every time frame t the HMM uses two independent observations: the MFCC features x_t and the LSTM phoneme prediction feature b_t. The vector x_t also serves as input for the LSTM, whereas the size of the LSTM input layer i_t corresponds to the dimensionality of the acoustic feature vector. The vector o_t contains one probability score for each of the P different phonemes at each time step. b_t is the index of the most likely phoneme:

$$b_t = \max_{o_t}(o_{t,1}, ..., o_{t,j}, ..., o_{t,P}) \tag{1}$$

In every time step the LSTM generates a phoneme prediction according to Equation 1 and the HMM models $x_{1:T}$ and $b_{1:T}$ as two independent data streams. With $y_t = [x_t; b_t]$ being the joint feature vector consisting of continuous MFCC and discrete LSTM observations and the variable a denoting the stream weight of the first stream (i. e., the MFCC stream), the multi-stream HMM emission probability while being in a certain state s_t can be written as

$$p(y_t|s_t) = \left[\sum_{m=1}^{M} c_{s_t m}\mathcal{N}(x_t; \mu_{s_t m}, \Sigma_{s_t m})\right]^a \times p(b_t|s_t)^{2-a}. \tag{2}$$

Thus, the continuous MFCC observations are modeled via a mixture of M Gaussians per state while the LSTM prediction is modeled using a discrete probability distribution $p(b_t|s_t)$. The index m denotes the mixture component, $c_{s_t m}$ is the weight of the m'th Gaussian associated with state s_t, and $\mathcal{N}(\cdot; \mu, \Sigma)$ represents a multivariate Gaussian distribution with mean vector μ and covariance matrix Σ. The distribution $p(b_t|s_t)$ is trained to model typical phoneme confusions that occur in the LSTM network. In our experiments, we restrict ourselves to the 15 most likely phoneme confusions per state and use a floor value of 0.01 for the remaining confusion likelihoods.

The applied real-time LSTM phoneme predictor is publicly available as part of our on-line speech feature extraction engine openSMILE [12].

5 Experiments and Results

In the following we will show the effects of using multi-condition training for a keyword detector based on a conventional single-stream continuous ASR system (Section 5.1), and the performance gain that can be obtained when applying the multi-stream LSTM-HMM principle (Section 5.2).

5.1 Multi-condition Training

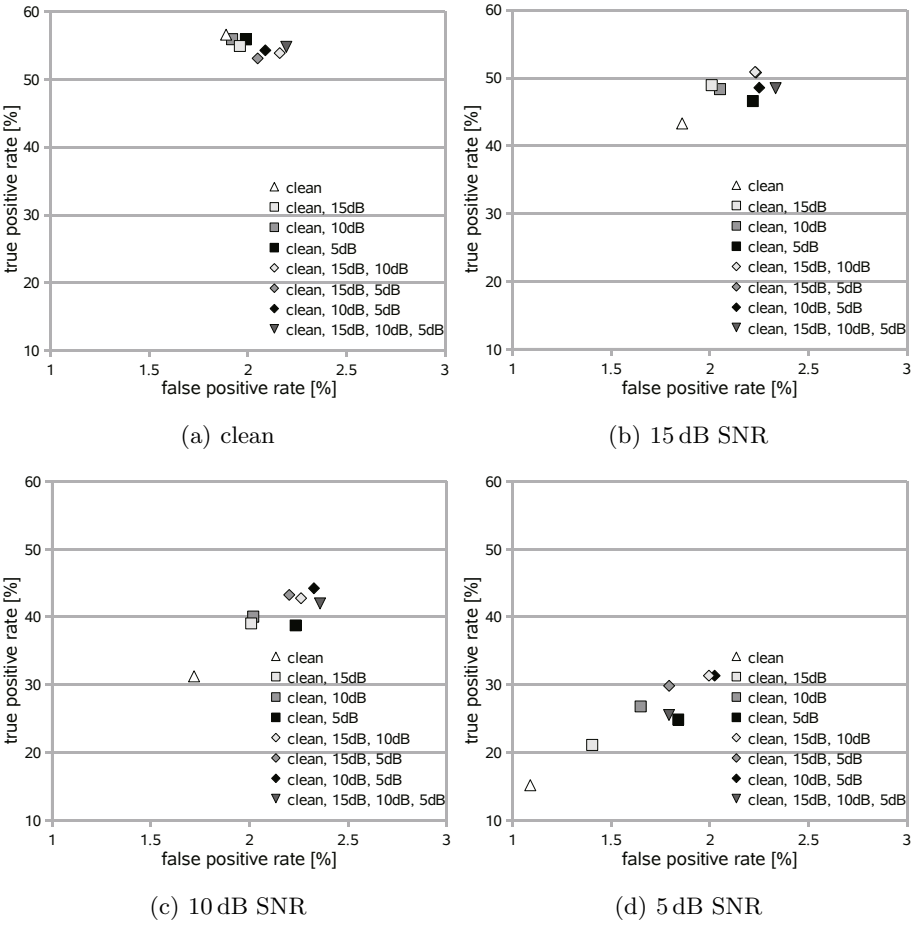
To improve keyword detection accuracy in noisy conditions, we investigated true positive and false positive rates when including noisy speech material in the training process. For all experiments, a part of the training material consisted of unprocessed versions of the SEMAINE database (recordings 1 to 10), the SAL corpus, and the COSINE database. This speech material will be referred to as *clean* in the ongoing (even though the COSINE corpus was partly recorded under noisy conditions). In addition to the 'clean' models, we evaluated different extensions of the training material by adding distorted versions of the SEMAINE and the SAL corpus. For this purpose, we superposed the clean speech with additive noise at different SNR levels: 15 dB, 10 dB, and 5 dB. We considered both, white Gaussian noise and babble noise from the NOISEX database. For evaluation, we used clean and distorted versions of the SEMAINE database

(recordings 11 to 19). Since conversational agents such as the SEMAINE system are often used while other people talk in the background, the babble noise evaluation scenario is most relevant for our application. We considered a set of 173 keywords and three different non-linguistic vocalizations (*breathing*, *laughing*, and *sighing*). The training/test distribution for *breathing*, *laughing*, and *sighing* was 124/54, 268/227, and 45/8, respectively. Keyword detection was based on simply searching for the respective words in the most likely ASR hypothesis. The applied trigram language model was trained on the SEMAINE corpus (recordings 1 to 10), the SAL database, and the COSINE database (total vocabulary size 6.1 k). Via openSMILE [12], 13 cepstral mean normalized MFCC features along with first and second order temporal derivatives were extracted from the speech signals every 10 ms. All cross-word triphone HMMs consisted of 3 emitting states with 16 Gaussian mixtures per state. For non-linguistic vocalizations, we trained HMMs consisting of 9 states.

Figures 2(a) to 2(d) show the Receiver Operating Characteristic (ROC) operating points for clean test material as well as for speech superposed with babble noise at 15 dB, 10 dB, and 5 dB SNR, respectively, when using different acoustic models. As can be seen in Figure 2(a), models exclusively trained on clean speech lead to the best performance for clean test data. We obtain a true positive rate of 56.58 % at a false positive rate of 1.89 % which is in the range of typical recognition rates for highly disfluent, spontaneous, and emotionally colored speech [7]. Including noisy training material slightly increases the false positive rate to up to 2.20 % at a small decrease of true positive rates. Yet, when evaluating the models on speech superposed by babble noise, multi-condition training significantly increases the true positive rates. A good compromise between high true positive rates and low false positive rates in noisy conditions can be obtained by applying the acoustic models denoted as 'clean, 15 dB, 10 dB' in Figures 2(a) to 2(d), i. e., models trained on the clean versions of the SEMAINE, SAL, and COSINE corpus, on the SEMAINE and SAL database superposed by babble noise at 15 dB SNR, and on the 10 dB versions of the SEMAINE and SAL database. For test data superposed by babble noise, this training set combination leads to the highest average true positive rate (41.66 %, see Table 1) at a tolerable average false positive rate. A similar result can be observed for the evaluation on test data corrupted by white noise (see Table 2). Models that are partly trained on speech superposed by white noise enable higher true positive rates in noisy conditions than 'clean' models. As for the babble noise scenario, a combination of clean, 15 dB SNR, and 10 dB SNR training data results in the best true positive/false positive compromise.

5.2 Multi-stream Decoding

To improve keyword detection in clean conditions, we implemented and evaluated the multi-stream LSTM-HMM decoder introduced in Section 4. Since the LSTM network was trained on framewise phoneme targets, we used an HMM system to obtain phoneme borders via forced alignment. The multi-stream system was trained on the clean versions of the SEMAINE, SAL, and COSINE databases



(a) clean (b) 15 dB SNR

(c) 10 dB SNR (d) 5 dB SNR

Fig. 2. ROC operating points obtained for different acoustic models when tested on clean speech and speech superposed by babble noise at 15, 10, and 5 dB SNR; acoustic models were trained on unprocessed versions of the SEMAINE, SAL, and COSINE corpus ('clean') and on noisy versions of the SEMAINE and SAL corpus using different SNR level combinations (babble noise)

and applied an LSTM network with a hidden layer consisting of 128 memory blocks. Each memory block contained one memory cell.

For LSTM network training we used a learning rate of 10^{-5} and a momentum of 0.9. Prior to training, all weights were randomly initialized in the range from -0.1 to 0.1. Input and output gates used tanh activation functions, while the forget gates had logistic activation functions. We trained the networks on the standard (CMU) set of 41 different English phonemes, including targets for *silence*, *breathing*, *laughing*, and *sighing*. The stream weight variable a was set to one.

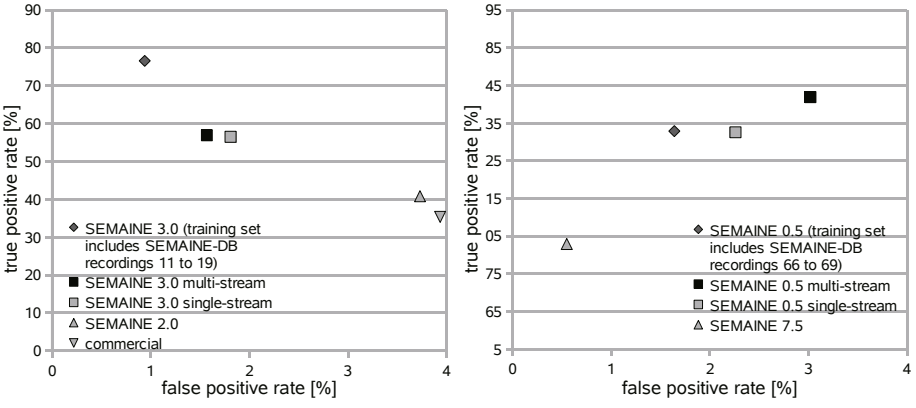
Table 1. Babble noise: Average true positive rates (tpr) and false positive rates (fpr) obtained with acoustic models trained on clean data and speech superposed by babble noise at different SNR conditions; clean and noisy test condition

training data	**test condition**			
SNR used for superposition	**babble noise**		**clean**	
with babble noise	**tpr [%]**	**fpr [%]**	**tpr [%]**	**fpr [%]**
clean	29.89	1.56	56.58	1.89
clean, 15 dB	36.37	1.81	54.91	1.96
clean, 10 dB	38.40	1.91	55.92	1.92
clean, 5 dB	36.73	2.10	55.90	1.99
clean, 15 dB, 10 dB	**41.66**	2.16	53.87	2.16
clean, 15 dB, 5 dB	41.29	2.08	53.08	2.05
clean, 10 dB, 5 dB	41.38	2.20	54.28	2.09
clean, 15 dB, 10 dB, 5 dB	38.67	2.16	54.79	2.20

Table 2. White noise: Average true positive rates (tpr) and false positive rates (fpr) obtained with acoustic models trained on clean data and speech superposed by white noise at different SNR conditions; clean and noisy test condition

training data	**test condition**			
SNR used for superposition	**white noise**		**clean**	
with white noise	**tpr [%]**	**fpr [%]**	**tpr [%]**	**fpr [%]**
clean	19.81	1.26	56.58	1.89
clean, 15 dB	39.40	2.28	57.31	2.06
clean, 10 dB	39.33	2.44	56.50	2.03
clean, 5 dB	23.65	1.20	56.02	2.01
clean, 15 dB, 10 dB	**42.47**	2.54	54.55	2.21
clean, 15 dB, 5 dB	42.11	2.57	54.28	2.16
clean, 10 dB, 5 dB	41.27	2.60	54.09	2.04
clean, 15 dB, 10 dB, 5 dB	**42.48**	2.69	50.42	2.27

The ROC operating points representing the keyword detection performance of the standard HMM (SEMAINE 3.0 single-stream) and the LSTM-HMM (SEMAINE 3.0 multi-stream) can be seen in Figure 3(a). All systems were evaluated on recordings 11 to 19 from the SEMAINE database. At a slight increase of the true positive rate, the incorporation of LSTM phoneme predictions can significantly reduce the false positive rate from 1.89 % to 1.57 %. For comparison, we also included the results for a preliminary version of the SEMAINE keyword detector (referred to as the SEMAINE 2.0 system [9]) which does not apply an in-domain language model and thus cannot compete with the current version (SEMAINE 3.0). Figure 3(a) also shows the performance obtained with a commercial recognizer as used in [13]. The comparably low performance of the commercial system indicates that using acoustic models tailored for the recognition of emotionally colored speech is essential for virtual agent applications such as the SEMAINE system. Since the final SEMAINE 3.0 keyword detector is trained on the *whole* SEMAINE database (including recordings 11 to 19),



(a) keyword detection (b) non-linguistic vocalization detection

Fig. 3. ROC operating points obtained for different variants of the SEMAINE keyword and non-linguistic vocalization detector

Figure 3(a) also shows the ROC performance obtained with models trained on all SEMAINE data. Note, however, that this configuration does not allow for a realistic performance assessment since training and test sets are not disjoint in this case. The reliability of non-linguistic vocalization detection (i. e., recognizing the events *breathing*, *laughing*, and *sighing*) can be seen in Figure 3(b). Again the multi-stream approach leads to a higher true positive rate, however, – in contrast to the keyword detection experiment – at the expense of a higher false positive rate.

6 Conclusion

This paper investigated how a keyword detector incorporated in a conversational agent system can be improved via multi-stream LSTM-HMM decoding and multi-condition training. We proposed a multi-stream system that models context-sensitive phoneme predictions generated by a Long Short-Term Memory network. In conformance with our previous observations concerning LSTM-based keyword spotting [7], we found that the LSTM principle is well-suited for robust phoneme prediction in challenging ASR scenarios. Performance gains in noisy conditions could be obtained applying multi-condition training. Since virtual agents are often used while people talk in the background, we mainly considered test conditions during which the speech signal is superposed by babble noise. Incorporating training material that is overlaid by background voices at different SNR conditions could enhance the noise robustness of keyword detection.

To further improve multi-stream LSTM-HMM keyword detection for conversational agents, future experiments should evaluate alternative network topologies such as *bottleneck* LSTM architectures as well as bidirectional context modeling for refinement of sentence hypotheses at the end of an utterance.

Acknowledgments. The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 211486 (SEMAINE).

References

1. McTear, M.F.: Spoken dialogue technology: enabling the conversational user interface. ACM Computing Surverys 34(1), 90–169 (2002)
2. Droppo, J., Acero, A.: Environmental robustness. In: Handbook of Speech Processing, pp. 658–659. Springer, Heidelberg (2007)
3. Schuller, B., Wöllmer, M., Moosmayr, T., Rigoll, G.: Recognition of noisy speech: A comparative survey of robust model architecture and feature enhancement. Journal on Audio, Speech, and Music Processing (2009), ID 942617
4. Zhu, Q., Chen, B., Morgan, N., Stolcke, A.: Tandem connectionist feature extraction for conversational speech recognition. In: Bengio, S., Bourlard, H. (eds.) MLMI 2004. LNCS, vol. 3361, pp. 223–231. Springer, Heidelberg (2005)
5. Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J.: Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In: Kremer, S.C., Kolen, J.F. (eds.) A Field Guide to Dynamical Recurrent Neural Networks, pp. 1–15. IEEE Press, Los Alamitos (2001)
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation 9(8), 1735–1780 (1997)
7. Wöllmer, M., Eyben, F., Graves, A., Schuller, B., Rigoll, G.: Bidirectional LSTM networks for context-sensitive keyword detection in a cognitive virtual agent framework. Cognitive Computation 2(3), 180–190 (2010)
8. Wöllmer, M., Eyben, F., Schuller, B., Rigoll, G.: Recognition of spontaneous conversational speech using long short-term memory phoneme predictions. In: Proc. of Interspeech, Makuhari, Japan, pp. 1946–1949 (2010)
9. Schröder, M., Cowie, R., Heylen, D., Pantic, M., Pelachaud, C., Schuller, B.: Towards responsive sensitive artificial listeners. In: Proc. of 4th Intern. Workshop on Human-Computer Conversation, Bellagio, Italy, pp. 1–6 (2008)
10. Wöllmer, M., Schuller, B., Eyben, F., Rigoll, G.: Combining long short-term memory and dynamic bayesian networks for incremental emotion-sensitive artificial listening. IEEE Journal of Selected Topics in Signal Processing 4(5), 867–881 (2010)
11. Stupakov, A., Hanusa, E., Bilmes, J., Fox, D.: COSINE - a corpus of multi-party conversational speech in noisy environments. In: Proc. of ICASSP, Taipei, Taiwan (2009)
12. Eyben, F., Wöllmer, M., Schuller, B.: openSMILE - the Munich versatile and fast open-source audio feature extractor. In: Proc. of ACM Multimedia, Firenze, Italy, pp. 1459–1462 (2010)
13. Principi, E., Cifani, S., Rocchi, C., Squartini, S., Piazza, F.: Keyword spotting based system for conversation fostering in tabletop scenarios: preliminary evaluation. In: Proc. of HSI, Catania, Italy, pp. 216–219 (2009)

On Control of Hopf Bifurcation in a Class of TCP/AQM Networks

Jianzhi Cao and Haijun Jiang

College of Mathematics and System Sciences, Xinjiang University, Urumqi, 830046, China
jianghai@xju.edu.cn,cjz2004987@163.com

Abstract. In this paper, we consider Hopf bifurcation control for an Internet congestion model, namely fluid flow model of TCP/AQM networks with a single link. It has been shown that the system without control can undergo Hopf bifurcation when the communication delay R passes through a critical value. To control the bifurcation, a washout filter based control model is proposed to delay the onset of undesirable Hopf bifurcation. Numerical simulation results confirm that the controller is efficient in controlling Hopf bifurcation.

Keywords: Bifurcation control, Time delay, Washout filter, Networks.

1 Introduction

In recent years, controlling and anti-controlling bifurcation and chaos have attracted many researchers from various disciplines. Bifurcation control refers to the task of designing a controller to suppress or reduce some existing bifurcation dynamics of a given nonlinear system, thereby achieving some desirable dynamical behaviors. As early as in the 60th of last century, Andronov *et al*. have been studied bifurcation control, in the sequence, Abed *et al*. have developed the related theory. Since then, the research in this fields was rapidly activated. For example, in [5], Bleich and Socolar used time-delayed feedback to obtain stable periodic orbits in a chaotic system. In [9], Yu and Chen developed a nonlinear feedback controller with polynomial functions to control Hopf bifurcation in the Lorenz and Rossler systems. However, it has been noted that the application of washout filter approach in controlling is not so popular. The use of washout filters ensures that low frequency orbits of the system are retained in the closed loop system, with only the transient dynamics and higher frequency orbits modified (see [7, 11, 16]).

Internet congestion, appearing when the required resources overrun the capacity of the Internet communication, is a serious problem in practical applications. Over the past years, many Internet congestion control mechanisms are developed to ensure the reliable and efficient exchange of information across the Internet. Transmission Control Protocol (TCP) and Active Queue Management (AQM) are among the core of those congestion control mechanisms.

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 506–514, 2011.

In [2, 14], a dynamic model of TCP/AQM networks was developed, and described by the following system:

$$\begin{cases} \dot{W}(t) = \dfrac{1}{R} - \dfrac{W(t)W(t)}{2R} Kq(t-R), \\ \dot{q}(t) \ = N\dfrac{W(t)}{R} - C, \end{cases} \tag{1}$$

where $W(t)$ is the average of TCP windows size (packet), $q(t)$ denotes the average queue length (packet), See [2] for detailed explanation for the model (1) and the parameters. In [2, 14], the authors studied the local stability, Hopf bifurcation and direction of the bifurcation periodic solutions of system (1). In reality, the occurrence of a Hopf bifurcation is some times harmful to the system. Therefore, in this paper, by applying the washout filter approach, we are interesting in designing a controller to delay the onset of Hopf bifurcation. We will show, with a Hopf bifurcation controller, that one can increase the critical value of the delay R, which benefits congestion controls.

The rest of this paper is organized as follows. In section 2, we recall the main results for the Hopf bifurcation of the TCP/AQM networks obtained in [2], then, based on washout filter, we study the stability and Hopf bifurcation of the control model of (1). To verify the theoretical analysis, numerical simulations are carried out for an example in section 3.

2 Bifurcation Control Based on Washout filter

In this section, the results of Hopf bifurcation for the model (1), obtained in [2], are summarized here for completeness and convenience.

Theorem 1. *(Theorem 1, [2]) For system (1), the following results hold:*

(1) When $R < R_0$, the equilibrium point E is locally asymptotically stable.
(2) When $R > R_0$, the equilibrium point E is unstable.
(3) When $R = R_0$, system (1) exhibits a Hopf bifurcation.

$$\Big(\text{ where } R_0 = \frac{1}{\omega_0}\arctan(\frac{a}{\omega_0}), \quad E = (\frac{RC}{N}, \frac{2N^2}{R^2C^2K}) \Big).$$

Theorem 2. *(Theorem 2, [2]) For system (1), when $R = R_0$, the direction and stability of periodic solutions of the Hopf bifurcation is determined by the formulas (2) and the following results hold:*

(i) μ_2 determines the direction of the Hopf bifurcation. If $\mu_2 > 0 (< 0)$, the Hopf bifurcation is supercritical (subcritical) and the bifurcating periodic solutions exist for $R > R_0 (R < R_0)$.
(ii) β_2 determines the stability of the bifurcating periodic solution. If $\beta_2 < 0 (> 0)$, then the bifurcating periodic solutions are stable (unstable).

(iii) T_2 determines the period of the bifurcating periodic solution. If $T_2 > 0 (< 0)$, the period increases (decreases) The parameters μ_2, β_2, T_2 are given by

$$\begin{aligned} \mu_2 &= -\frac{Re\{C_1(0)\}}{Re\lambda'(0)}, \\ T_2 &= -\frac{Im\{C_1(0)\} + \mu_2 Im\lambda'(0)}{\omega_0}, \\ \beta_2 &= -Re\{C_1(0)\}. \end{aligned} \tag{2}$$

The detailed derivation of the above formulas can be found in [2].

We consider a general form of dynamical system with parameter μ:

$$\dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{y}; \mu) \quad (\mathbf{f}(t, \mathbf{0}; \mu) = \mathbf{0}). \tag{3}$$

Adding a control action u to the equation on one component y_i, and taking u as the following form:

$$\begin{cases} u = g(x, k), \\ \dot{\omega} = y_i - d\omega \triangleq x, \end{cases} \tag{4}$$

where g is a nonlinear function and k is the control parameter.

In order to keep the structure of all equilibrium point of Eq.(4), the following constrains should be fulfilled:

$$\begin{cases} d \quad\quad > 0, \\ g(0, k) = 0. \end{cases} \tag{5}$$

Furthermore, the control system can be designed as follows:

$$\begin{cases} \dot{y} = f(t, y; \mu) + u, \\ u = g(x, k), \\ \dot{\omega} = y - d\omega \triangleq x. \end{cases} \tag{6}$$

By employing the above approach, we can obtain the following control system for (1):

$$\begin{cases} \dot{W}(t) = \dfrac{1}{R} - \dfrac{W(t)W(t)}{2R} Kq(t-R) + g(\xi), \\ \dot{q}(t) \;\; = \dfrac{N}{R} W(t) - C, \\ \dot{u} \quad\; = W(t) - \alpha u \triangleq \xi, \\ g(\xi) \;\; = -K_1\xi - K_2\xi^2, \end{cases} \tag{7}$$

where $\alpha > 0$ is a constant scaling parameter and K_1, K_2 are parameters, which can be used to control the Hopf bifurcation.

The linearized equation of system (7) is the following form:

$$\begin{cases} \dot{W}(t) = -\dfrac{2N}{R^2C} W(t) - \dfrac{KRC^2}{2N^2} q(t-R) - K_1(W(t) - \alpha u), \\ \dot{q}(t) \;\; = \dfrac{N}{R} W(t), \\ \dot{u} \quad\; = W(t) - \alpha u. \end{cases} \tag{8}$$

The associated characteristic equation of system (8) is the following third degree exponential polynomial equation:

$$\lambda^3 + (a + K_1 + \alpha)\lambda^2 + a\alpha\lambda + b(\lambda + \alpha)e^{-\lambda R} = 0, \tag{9}$$

where $a = \dfrac{2N}{R^2C} > 0, \quad b = \dfrac{KC^2}{2N} > 0.$

Obviously, $\pm i\omega(\omega > 0)$ is a root of Eq.(9) if and only if ω satisfies

$$-i\omega^3 - (a + K_1 + \alpha)\omega^2 + a\alpha i\omega + b(i\omega + \alpha)\Big(\cos(\omega R) - i\sin(\omega R)\Big) = 0.$$

Separating the real and imaginary parts, we have

$$\begin{cases} -(a + K_1 + \alpha)\omega^2 + b\omega\sin(\omega R) + b\alpha\cos(\omega R) = 0, \\ \quad -\omega^3 + a\alpha\omega + b\omega\cos(\omega R) - b\alpha\sin(\omega R) \quad = 0, \end{cases} \tag{10}$$

which implies

$$z^3 + a_1 z^2 + a_2 z + a_3 = 0, \tag{11}$$

where

$$\begin{aligned} z &= \omega^2, \\ a_1 &= (a + K_1 + \alpha)^2 - 2a\alpha > 0, \\ a_2 &= a^2\alpha^2 - b^2, \\ a_3 &= -b^2\alpha^2 < 0. \end{aligned}$$

Denote

$$h(z) = z^3 + a_1 z^2 + a_2 z + a_3, \tag{12}$$

since $a_3 < 0$ and $h(+\infty) = +\infty$, then Eq.(12) has at least one positive root. Without loss of generality, we assume that it has three positive roots defined by z_1, z_2 and z_3, respectively. Furthermore, we have $\omega_1 = \sqrt{z_1}$, $\omega_2 = \sqrt{z_2}$ and $\omega_3 = \sqrt{z_3}$.

By (10), we have

$$R_k^{(j)} = \frac{1}{\omega_k}\Big(\arccos\Big\{\frac{\omega_k^4 - a\alpha\omega_k^2 + \alpha(a + K_1 + \alpha)\omega_k^2}{b(\omega_k^2 + \alpha^2)}\Big\} + 2j\pi\Big),$$

where $k = 1, 2, 3, \quad j = 0, 1, 2, \cdots$. Then $\pm i\omega_k$ is a pair of purely imaginary roots of Eq.(9) with $R_k^{(j)}$.

Note that when $R = 0$, Eq.(9) becomes:

$$\lambda^3 + (a + K_1 + \alpha)\lambda^2 + (a\alpha + b)\lambda + b\alpha = 0. \tag{13}$$

By Routh-Huruitz criterion, we know that all the roots of (13) have negative parts, i.e. the positive equilibrium point E is locally asymptotically stable for $R = 0$.

Let $\lambda(R)=\alpha(R)+i\omega(R)$ be the root of Eq.(9) satisfying

$$\alpha(R_k^{(j)})=0, \quad \omega(R_k^{(j)})=\omega_k.$$

Substituting $\lambda(R)$ into Eq.(9) and differentiating both sides with respect to R, we have

$$\Big(3\lambda^2+2(a+K_1+\alpha)\lambda+a\alpha+b(1-R(\lambda+\alpha))e^{-\lambda R}\Big)\frac{d\lambda}{dR}=-a'\lambda^2-a'\alpha\lambda+b\lambda(\lambda+\alpha)e^{-\lambda R},$$

where $a'=-\dfrac{4N}{CR^3}$.
Then, we can easily obtain

$$\begin{aligned} Re\Big(\frac{d\lambda}{dR}\Big)^{-1}\Big|_{R=R_k^{(j)}} &= Re\Big\{-\frac{3\lambda+2(a+K_1+\alpha)}{-a'\lambda-a'\alpha+b(\lambda+\alpha)e^{-\lambda R}}\Big\}_{R=R_k^{(j)}} \\ &\quad +Re\Big\{-\frac{a\alpha+b(1-R(\lambda+\alpha))e^{-\lambda R}}{-a'\lambda^2-a'\alpha\lambda+b\lambda(\lambda+\alpha)e^{-\lambda R}}\Big\}_{R=R_k^{(j)}}. \end{aligned}$$

After some calculations, we have

$$\mathrm{Re}\Big(\frac{d\lambda}{dR}\Big)\Big|_{R=R_k^{(j)}}=\mathrm{Re}\Big(\frac{d\lambda}{dR}\Big)^{-1}\Big|_{R=R_k^{(j)}}\neq 0,$$

then, the transversality condition is hold.

Remark 1. *We should point out that Lemma 5 in [2] is incorrect, it is obviously that the differentiating both sides with respect to R is wrong, this leads to the mistake of all the proof and the right result for the transversality condition see [14] .*

Till now, we can employ a result from Ruan and Wei [10] to analyze Eq.(9), for the convenience of the reader, stated as follows.

Lemma 1. *Consider the exponential polynomial*

$$\begin{aligned} P(\lambda,e^{-\lambda\tau_1},\ldots,e^{-\lambda\tau_m}) &= \lambda^n+p_1^{(0)}\lambda^{n-1}+\cdots+p_{n-1}^{(0)}\lambda \\ &\quad +p_n^{(0)}+[p_1^{(1)}\lambda^{n-1}+\cdots+p_{n-1}^{(1)}\lambda+p_n^{(1)}]e^{-\lambda\tau_1} \\ &\quad +\cdots+[p_1^{(m)}\lambda^{n-1}+\cdots+p_{n-1}^{(m)}\lambda+p_n^{(m)}]e^{-\lambda\tau_m}. \end{aligned}$$

where $\tau_i\geq 0(i=1,2,\ldots,m)$, $p_j^{(i)}(i=1,2,\ldots,m;j=1,2,\ldots n)$ are constants. As $(\tau_1,\tau_2,\cdots,\tau_m)$ vary, the sum of the order of the zeros of $P(\lambda,e^{-\lambda\tau_1},\ldots,e^{-\lambda\tau_m})$ on the open right half plane can change only if a zero appears on or crosses the imaginary axis.

From the above discussion and the Hopf bifurcation theorem (see [4]), we obtain the following result.

Theorem 3. *For system (7), the following statements are true:*
The equilibrium point E is asymptotically stable for the delay $R \in [0,\ R_0)$ and unstable for $R > R_0$. System (7) with $R = R_k^{(j)}$ $(k = 1, 2, 3,\ j = 0, 1, 2, \cdots)$ undergoes a Hopf bifurcation $(R_0 = \min_{k,j}\{R_k^{(j)}\})$.

3 Numerical Examples

In this section, some numerical examples are given to verify the results obtained in the previous section. For comparison, we choose the same parameter as [2]

$$N = 50, \quad K = 0.001, \quad C = 1000.$$

It follows from system (1) that

$$W^* = 4,\ q^* = 125,\ a = 2.5,\ b = 10,\ R = 0.2,\ R_0 = 0.27471.$$

From Theorem 1, it is shown that when $R < R_0$, trajectories converge to the equilibrium point, while as R is increased to pass R_0, E loses its stability and a Hopf bifurcation occurs. The dynamical behavior of this uncontrolled model (1) is illustrated in Figs. 1-2.

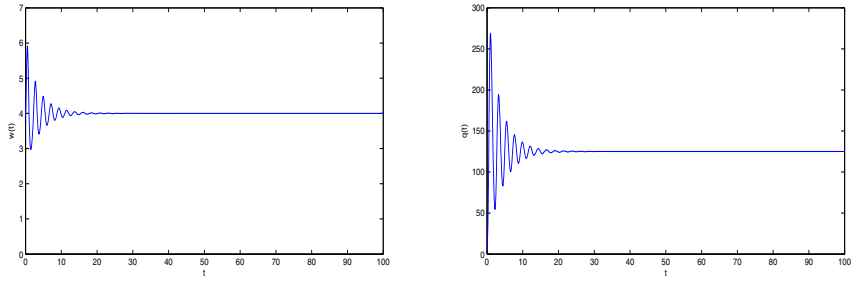


Fig. 1. Waveform of without control system (1) $R = 0.2,\ R_0 = 0.27471$

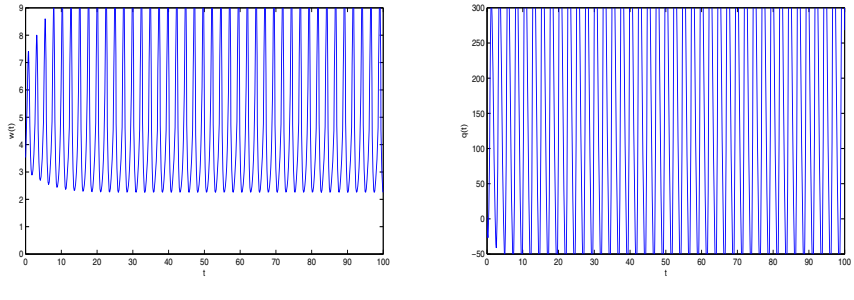


Fig. 2. Waveform of without control system (1) $R = 0.23,\ R_0 = 0.20002$

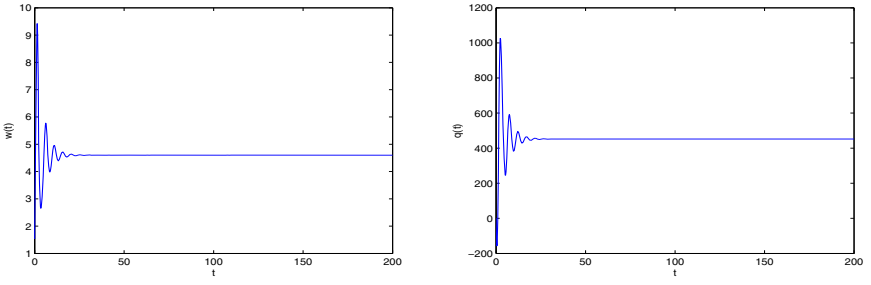


Fig. 3. Waveform of the control system (7) with $R = 0.23,\ K_1 = 0.25,\ \ K_2 = 0,\ \alpha = 2.3$

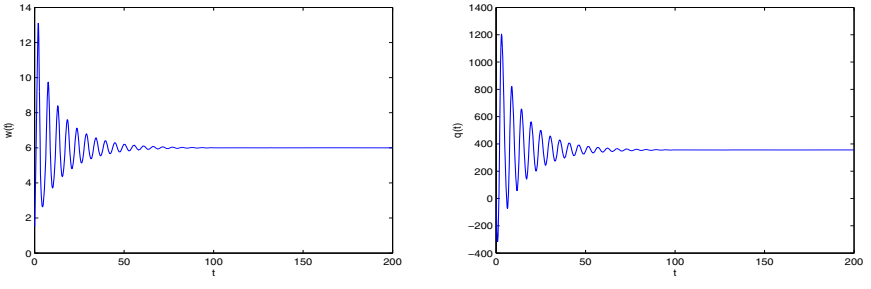


Fig. 4. Waveform of the control system (7) with $R = 0.3,\ K_1 = 0.25,\ \ K_2 = 0,\ \alpha = 2.3$

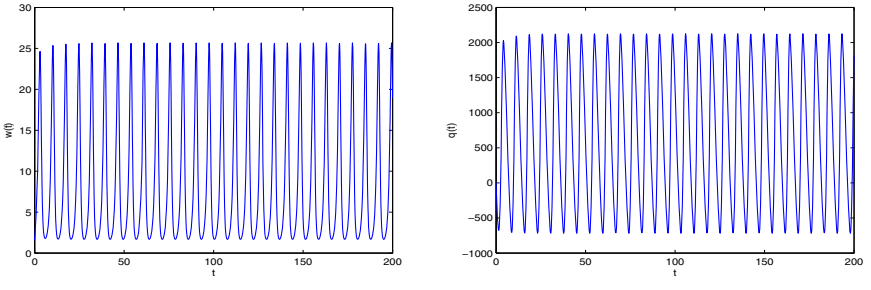


Fig. 5. Waveform of the control system (7) with $R = 0.43,\ K_1 = 0.25,\ \ K_2 = 0,\ \alpha = 2.3$

Now we choose appropriate value of K_1 and K_2 to control the Hopf bifurcation. It is easy to see from Theorem 3 that an appropriate value of K_1 and K_2, we can delay the onset of the Hopf bifurcation(see Figs. 3-6).

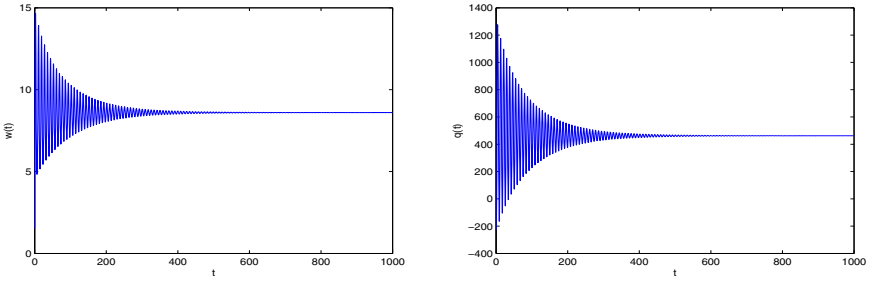


Fig. 6. Waveform of the control system (7) with $R = 0.43$, $K_1 = 1.78$, $K_2 = 0.24$, $\alpha = 2.3$

Acknowledgments

This work was supported by The National Natural Science Foundation of P.R. China (60764003), The Natural Science Foundation of Xinjiang (2010211A07) and The Scientific Research Programmes of Colleges in Xinjiang (XJEDU2007G01).

References

1. Chen, D., Wang, H.O., Chen, G.: Anti-control of Hopf bifurcations through washout filters. In: Proceedings of the 37th IEEE Conference on Decision and Control, vol. 3, pp. 3040–3045 (1998)
2. Ding, D., Zhu, J., Luo, X.: Hopf bifurcation analysis in a fluid flow model of Internet congestion control algorithm. Nonlinear Anal: RWA 10, 824–839 (2009)
3. Hu, H., Huang, L.: Linear stability and Hopf bifurcation in an exponential RED algorithm model. Nonlinear Dyn. 59, 463–475 (2010)
4. Hale, J.K.: Theory of Functional Differential Equations. Springer, New York (1977)
5. Bleich, M., Socolar, J.: Stability of periodic orbits controlled by time-delay feedback. Phys. Lett. A 210, 87–94 (1996)
6. Xiao, M., Cao, J.: Delayed feedback-based bifurcation control in a Internet congestion model. J. Math. Anal. Appl. 332, 1010–1027 (2007)
7. Xiao, M., Cao, J.: On control of hopf bifurcation in BAM neural network with delayed self-feedback. In: Wang, J., Yi, Z., Żurada, J.M., Lu, B.-L., Yin, H. (eds.) ISNN 2006. LNCS, vol. 3971, pp. 285–290. Springer, Heidelberg (2006)
8. Vasegh, N., Sedigh, A.K.: Delayed feedback control of time-delayed chaotic systems: A nalytical approach at Hopf bifurcation. Phys. Lett. A 372, 5110–5114 (2008)
9. Yu, P., Chen, G.: Hopf bifurcation control using nonlinear feedback with polynomial functions. Int. J. Bifurcat Chaos 14, 1683–1704 (2004)
10. Ruan, S., Wei, J.: On the zeros of transcendental functions with applications to stability of delay differential equations with two delays. Dyn. Contin. Discrete Impuls. Syst. Ser. A Math. Anal. 10, 863–874 (2003)
11. Zhou, S., Liao, X., Wu, Z., Wong, K.: Hopf bifurcation in a control system for the Washout filter-based delayed neural equation. Chaos Solitons Fractals 23, 101–115 (2005)

12. Zhou, S., Liao, X., Yu, J., Wong, K.: On control of Hopf bifurcation in time-delayed neural network system. Phys. Lett. A 338, 261–271 (2005)
13. Lan, Y., Li, Q.: Control of Hopf bifurcation in a simple plankton population model with a non-integer exponent of closure. Appl. Math. Comut. 200, 220–230 (2008)
14. Zheng, Y., Wang, Z.: Stability and Hopf bifurcation of a class of TCP/AQM networks. Nonlinear Anal: RWA 11, 1552–1559 (2010)
15. Song, Y., Han, M., Peng, Y.: Stability and Hopf bifurcations in a competitive Lotka-Volterra system with two delays. Chaos Solitons Fractals 22, 1139–1148 (2004)
16. Chen, Z., Yu, P.: Hopf bifurcation control for an Internet congestion model. Int. J. Bifurcat. Chaos 8, 2643–2651 (2005)

State Feedback Control Based on Twin Support Vector Regression Compensating for a Class of Nonlinear Systems

Chaoxu Mu, Changyin Sun, and Xinghuo Yu

School of Automation, Southeast University, Nanjing, Jiangsu, China
cysun@seu.edu.cn

Abstract. In this paper, we introduce a new twin support vector regression (TSVR) algorithm, which estimates an unknown function by approaching its up and lower boundary, the ending function is obtained by the mean of the two function. For the class of nonlinear systems composed by linear and nonlinear parts, we use TSVR with a wavelet kernel to estimate the unknown nonlinear part in the original system and to counteract it, and then a state feedback control is carried out to realize a close loop control for the compensated system. Simulation results show that the TSVR with the wavelet kernel has good approaching ability and generalization. The whole close loop system with a state feedback control is stable when the compensating errors satisfy certain conditions.

1 Introduction

Support Vector Machine (SVM) as a powerful learning algorithm has made great progress in pattern classification and regression for last decade. SVM based on the statistical learning theory balance the optimization between structural complexity and empirical risk. The solving of optimization problem involves the minimization of a convex quadratic function subject to linear inequality constraints. Advantages of SVM are that no number of hidden units has to be determined and the curse of dimensionality can also be avoided in comparison with neural networks [1-4].

Recently, Mangasarian and Wild proposed the generalized eigenvalue proximal support vector machine (GEPSVM), which is a nonparallel plane classifier for binary data classification [5]. In this approach, data points of each class are proximal to one of two nonparallel planes. The nonparallel planes are eigenvector to the smallest eigenvalue of two related generalized eigenvalue problems. Jayadeva and Khemchandani proposed a new nonparallel plane classifier, termed as the twin support vector machine (TSVM) for binary data classification [6]. TSVM also generate two nonparallel planes such that each plane is close to one of the two classes and is as far as possible from the other. However, the formulation of TSVM is much different from that of GEPSVM and is similar with standard SVM. The solving of TSVM is a pair of quadratic programming problems (QPPs), whereas, it solves a single QPP in SVM. This strategy of

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 515–524, 2011.

solving two smaller sized QPP, rather than one large QPP, makes TSVM less complexity than SVM. TSVM has become one of the popular methods in machine learning because of its low computational complexity. For example, Ghorai and Mukherjee et al formulated a simpler nonparallel plane proximal classifier to speed up the training according to TSVM [7], Kumar and Gopal firstly enhanced the TSVM using smoothing techniques and then proposed a least square version of TSVM for binary classification [8-9]. As for support vector regression (SVR), there exist some corresponding approximation algorithms as classification. In this paper, we introduce a nonparallel plane approximation in the spirit of TSVM, termed as the twin support vector regression (TSVR).

TSVR also aims at generating two nonparallel functions such that each function determines the ε-insensitive lower or up boundary of the unknown approximation. Similar to TSVM, TSVR also solve a pair of QPPs instead of a single QPP in SVR and one group of constraints for all data points are used in each QPP in TSVR. TSVR is less computational complexity than that of SVR , but also shows good generalization [10].

Some nonlinear systems can be expressed by the sum of linear and nonlinear parts, such as chaotic system [11]. Many scholars hope to realize that the control to the kind of systems by intelligent control. Neural networks have been applied to identification and control to this kind of systems. In this paper, we use TSVR with a wavelet kernel to estimate the nonlinear part, and then the nonlinear TSVR model is as a compensator to counteract the nonlinear part in the original system, and finally a state feedback control is carried out in a close loop system to realize effective control.

The paper is organized as follows. Section 2 firstly introduces linear and nonlinear twin support vector regression. In Section 3, we illustrate the state feedback control based on TSVR nonlinear compensating. Section 4 deals with experimental results to validate the effectiveness of the state feedback control for the compensated system, and Section 5 gives some concluding remarks.

2 Twin Support Vector Regression with Wavelet Kernel

2.1 Twin Support Vector Regression

TSVR is similar to TSVM that derives a pair of nonparallel planes around data points. However, there are still some differences in essence. TSVR aims to find a suitable function by approaching the up and lower boundary, while TSVM is to construct the classifier by two hyperplanes. Two quadratic programming problems in TSVM have the typical SVM formulation except that not all points appear in one constraint condition in TSVM, while all points are presented in the constraint condition for each quadratic programming problem in the TSVR. Denote a data set as $\{x_i, y_i\}_{i=1}^{N}$ as $\{X, Y\}$, where $x_i \in R^n$ and $y_i \in R$. The TSVR optimal problem is described by the following formula,

$$\min \frac{1}{2} \|Y - e\varepsilon_1 - (Xw_1 + eb_1)\|^2 + c_1 e^{\mathrm{T}} \delta$$
$$\text{s.t.} \quad Y - (Xw_1 + eb_1) \geq e\varepsilon_1 - \delta, \delta \geq 0, \tag{1}$$
$$\min \frac{1}{2} \|Y + e\varepsilon_2 - (Xw_2 + eb_2)\|^2 + c_2 e^{\mathrm{T}} \eta$$
$$\text{s.t.} \quad Xw_2 + eb_2 - Y \geq e\varepsilon_2 - \eta, \eta \geq 0. \tag{2}$$

c_1, c_2, ε_1 and ε_2 are positive parameters, e is an unit column vector with suitable dimensions, and δ and η are slack vectors. We obtain $f_1(x) = w_1^{\mathrm{T}} x + eb_1$ and $f_2(x) = w_2^{\mathrm{T}} x + eb_2$ by solving the TSVR problem, which respectively determine the lower and the up boundary of the approached function. The ending function is decided by the mean of the two functions. The first item in (1) is the sum of squared distances from $y_1(x) = w_1^{\mathrm{T}} x + eb_1 + \varepsilon_1$ to the training points and the first item in (2) is also the sum of squared distances from $y_2(x) = w_2^{\mathrm{T}} x + eb_2 - \varepsilon_2$ to the training points. Therefore, minimizing these items make $f_1(x)$ close to ε_1-insensitive lower boundary and $f_2(x)$ close to ε_2 insensitive up boundary as much as possible. The constraint conditions arouse that $f_1(x)$ and $f_2(x)$ are at a distance of at least ε_1 or ε_2 from the training points. In another words, all training points should be larger than $f_1(x)$ at least ε_1, while they should be smaller $f_2(x)$ at least ε_2. The slack vectors δ and η measure separately the errors by the distances from data points to $y_i(x)$. δ measures the errors from training points to $f_1(x)$ are closer than ε_1 and η measures the errors from training points to $f_2(x)$ are closer than ε_2. The second term of the objective function is the sum of all error variables, and its minimization aims accurate approaching.

To solve the problem of TSVR in a dual space, α, β, μ, ν are Lagrange multiplier vectors and the Lagrangians are given by the following formula,

$$L_1(w_1, b_1, \delta, \alpha, \beta) = \frac{1}{2} \|Y - e\varepsilon_1 - (Xw_1 + eb_1)\|^2$$
$$+ c_1 e^{\mathrm{T}} \delta - \alpha^{\mathrm{T}} (Y - e\varepsilon_1 - (Xw_1 + eb_1) + \delta) - \beta^{\mathrm{T}} \delta. \tag{3}$$
$$L_2(w_2, b_2, \eta, \mu, \nu) = \frac{1}{2} \|Y + e\varepsilon_2 - (Xw_2 + eb_2)\|^2$$
$$+ c_2 e^{\mathrm{T}} \eta - \mu^{\mathrm{T}} (Xw_2 + eb_2 - Y - e\varepsilon_2 + \eta) - \nu^{\mathrm{T}} \eta. \tag{4}$$

We have the following equations for the above original problem (1) and (2) according to Karush-Kuhn-Tucker (KKT) optimal conditions,

$$\frac{\partial L_1}{\partial w_1} = 0, \frac{\partial L_1}{\partial b_1} = 0, \frac{\partial L_1}{\partial \delta} = 0, \frac{\partial L_2}{\partial w_2} = 0, \frac{\partial L_2}{\partial b_2} = 0, \frac{\partial L_2}{\partial \eta} = 0.$$

Since $\alpha, \beta, \mu, \nu \geq 0$, we have $0 \leq \alpha \leq c_1 e$ and $0 \leq \mu \leq c_2 e$. The following equation are obtained,

$$- \begin{bmatrix} X^{\mathrm{T}} \\ e^{\mathrm{T}} \end{bmatrix} ((Y - e\varepsilon_1) - \begin{bmatrix} X & e \end{bmatrix} \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} - \alpha) = 0, \tag{5}$$
$$- \begin{bmatrix} X^{\mathrm{T}} \\ e^{\mathrm{T}} \end{bmatrix} ((Y + e\varepsilon_2) - \begin{bmatrix} X & e \end{bmatrix} \begin{bmatrix} w_2 \\ b_2 \end{bmatrix} + \mu) = 0. \tag{6}$$

We can obtain w_1, b_1, w_2 and b_2 from the equation (5) and (6).

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = (\begin{bmatrix} X^{\mathrm{T}} \\ e^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} X\ e \end{bmatrix})^{-1} \begin{bmatrix} X^{\mathrm{T}} \\ e^{\mathrm{T}} \end{bmatrix} ((Y - e\varepsilon_1) - \alpha), \tag{7}$$

$$\begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = (\begin{bmatrix} X^{\mathrm{T}} \\ e^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} X\ e \end{bmatrix})^{-1} \begin{bmatrix} X^{\mathrm{T}} \\ e^{\mathrm{T}} \end{bmatrix} ((Y + e\varepsilon_2) + \mu). \tag{8}$$

As $\begin{bmatrix} X^{\mathrm{T}} \\ e^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} X\ e \end{bmatrix}$ is always positive semidefinite, it is possible that it may not be well conditioned in some situations. In order to overcome the defect, we introduce a regularization term to take care of the possible ill condition [12]. I is an identity matrix of appropriate dimensions. Therefore, (7) and (8) can be written as

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = (\begin{bmatrix} X^{\mathrm{T}} \\ e^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} X\ e \end{bmatrix} + \lambda I)^{-1} \begin{bmatrix} X^{\mathrm{T}} \\ e^{\mathrm{T}} \end{bmatrix} ((Y - e\varepsilon_1) - \alpha), \tag{9}$$

$$\begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = (\begin{bmatrix} X^{\mathrm{T}} \\ e^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} X\ e \end{bmatrix} + \lambda I)^{-1} \begin{bmatrix} X^{\mathrm{T}} \\ e^{\mathrm{T}} \end{bmatrix} ((Y + e\varepsilon_2) + \mu). \tag{10}$$

2.2 Twin Support Vector Regression with Wavelet Kernel

In order to extend the above linear regression to nonlinear regression, the kernel-generated space is considered. The lower and up boundary of estimated function are $f_1(x) = w_1^{\mathrm{T}} k(x, X) + b_1$ and $f_2(x) = w_2^{\mathrm{T}} k(x, X) + b_2$ separately. The kernel TSVR problem can be considered as follows,

$$\begin{aligned} &\min \frac{1}{2} \left\| Y - e\varepsilon_1 - (k(X, X^{\mathrm{T}})w_1 + eb_1) \right\|^2 + c_1 e^{\mathrm{T}} \delta \\ &\text{s.t.} \quad Y - (k(X, X^{\mathrm{T}})w_1 + eb_1) \geq e\varepsilon_1 - \delta, \delta \geq 0, \end{aligned} \tag{11}$$

$$\begin{aligned} &\min \frac{1}{2} \left\| Y + e\varepsilon_2 - (k(X, X^{\mathrm{T}})w_2 + eb_2) \right\|^2 + c_2 e^{\mathrm{T}} \eta \\ &\text{s.t.} \quad k(X, X^{\mathrm{T}})w_2 + eb_2 - Y \geq e\varepsilon_2 - \eta, \eta \geq 0. \end{aligned} \tag{12}$$

Similarly, the Lagrangians for the formula (11) and (12) are given by the following formula,

$$\begin{aligned} L_1'(w_1, b_1, \delta, \alpha, \beta) &= \frac{1}{2} \left\| Y - e\varepsilon_1 - (k(X, X^{\mathrm{T}})w_1 + eb_1) \right\|^2 \\ &+ c_1 e^{\mathrm{T}} \delta - \alpha^{\mathrm{T}} (Y - e\varepsilon_1 - (k(X, X^{\mathrm{T}})w_1 + eb_1) + \delta) - \beta^{\mathrm{T}} \delta, \end{aligned} \tag{13}$$

$$\begin{aligned} L_2'(w_2, b_2, \eta, \mu, \nu) &= \frac{1}{2} \left\| Y + e\varepsilon_2 - (k(X, X^{\mathrm{T}})w_2 + eb_2) \right\|^2 \\ &+ c_2 e^{\mathrm{T}} \eta - \mu^{\mathrm{T}} (k(X, X^{\mathrm{T}})w_2 + eb_2 - Y - e\varepsilon_2 + \eta) - \nu^{\mathrm{T}} \eta. \end{aligned} \tag{14}$$

According to KKT conditions the following equations are obtained,

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = G^{-1} \begin{bmatrix} k(X^{\mathrm{T}}, X) \\ e^{\mathrm{T}} \end{bmatrix} (Y - e\varepsilon_1 - \alpha), \tag{15}$$

$$\begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = G^{-1} \begin{bmatrix} k(X^{\mathrm{T}}, X) \\ e^{\mathrm{T}} \end{bmatrix} (Y + e\varepsilon_2 + \mu), \tag{16}$$

where $G = \left(\begin{bmatrix} k(X^{\mathrm{T}}, X) \\ e^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} k(X, X^{\mathrm{T}}) \; e \end{bmatrix} \right)$. Still there are possible to have ill-conditioning case in the formula (15) and (16) and we may use the mentioned above skill when an ill-conditioning case happens.

The support vector kernel function $k(\cdot, \cdot)$ must satisfy the Mercer condition. In other words, if a function can satisfy the mercer condition, it is an allowed kernel function. The following theorem explains the method to judge and construct a kernel function.

Theorem 1. *For $x \in R^n$ and $y \in R^n$, all functions $g(x) \neq 0$ which satisfy the condition of $\int_{R^n} g^2(x)dx < \propto$, if and only if the condition of $\iint_{R^n \otimes R^n} k(x,y)g(x)g(y)dxdy \geq 0$ is satisfied, the symmetry function $k(x, y)$ is an admissible support vector kernel [13].*

Theorem 2. *Theorem 2. A translation invariant kernel $k(x, y) = k(x - y)$ is allowed for a support vector kernel if and only if the Fourier transform $F[k(w)] = (2\pi)^{\frac{n}{2}} \int_{R_n} e(-jwx)k(x)dx \geq 0$ holds [14].*

When one wants to use a wavelet function as a support vector kernel, the Mercer condition and the properties of the wavelet function are both considered. Assuming that $\varphi(x)$ is a mother wavelet, then we can get the dot product wavelet kernel with the form of $k(x, y) = \Pi_{i=1}^{n} \varphi(\frac{x_i - b_i}{a_i}) \varphi(\frac{y_i - b_i}{a_i})$ and the translation invariant wavelet kernel that satisfies Theorem 2 with the form of $k(x, y) = \Pi_{i=1}^{n} \varphi(\frac{x_i - y_i}{a_i})$. a_i and b_i are dilation and translation factor, respectively. Here we choose a translation invariant wavelet kernel which is constructed by a wavelet function under the conditions of wavelet frames in this paper. The wavelet function is chosen as follows $\varphi(x) = cos(\theta x)e^{-x^2}$.

Theorem 3. *Theorem 3. A wavelet kernel under the wavelet frame is constructed as an admissible support vector kernel, which has the expression $k(x, y) = \Pi_{i=1}^{n} \varphi\left(\frac{x_i - y_i}{a_i}\right) = \Pi_{i=1}^{n} \left(cos(\theta \frac{x_i - y_i}{a_i}) \right) e^{-\frac{\|x_i - y_i\|^2}{a_i^2}}$ [15].*

Using the above wavelet kernel in the lower and up bound function $f_1(x)$ and $f_2(x)$ of TSVR, we can obtain the ending expression as $f(x) = \frac{1}{2}(w_1^{\mathrm{T}} + w_2^{\mathrm{T}})k(x, X) + \frac{1}{2}(b_1 + b_2)$.

3 State Feedback Control Based on TSVR Nonlinear Compensating

We study the class of nonlinear systems which can be written as the sum of linear and nonlinear functions. Its state equations are given in the following formula,

$$\dot{x} = f_l(x) + f_n(x) = Ax + Bu + f_n(x), \tag{17}$$

where $x \in R^n$ and $u \in R^p$ are the state and the input of system respectively. $f_l(\cdot)$ expresses a linear relationship and $f_n(\cdot)$ expresses a nonlinear relationship. We describe $f_l(x) = Ax + Bu$ with a systematic matrix $A_{n\times n}$ and a input matrix $B_{n\times p}$.

We use TSVR with the wavelet kernel to obtain an estimated function $\hat{f}_n(x)$, which is required to approach $f_n(x)$ as much as possible. Define $\varsigma(x)$ to evaluate the approaching error between $f_n(x)$ and $\hat{f}_n(x)$, so that we have the equation $\varsigma(x) = f_n(x) - \hat{f}_n(x)$. And in a perfect condition, $\varsigma(x)$ should be equal to zero. We adopt nonlinear compensating for the class of nonlinear systems, using $\hat{f}_n(x)$ to counteract $f_n(x)$. After $\hat{f}_n(x)$ is obtained by the TSVR method, the nonlinear state equation can be rewritten as follows,

$$\dot{x} = Ax + Bu + f_n(x) - \hat{f}_n(x) = Ax + Bu + \varsigma(x). \tag{18}$$

When the condition $\varsigma(x) \approx 0$ holds, the compensated nonlinear system has a dominate linear character by omitting the small nonlinearity. In order to control this kind of systems, we introduce two assumed conditions [16]. One is the linear part of original system is controllable and the other is the nonlinear part of original system is input and output measurable. On these conditions, we can design a linear state feedback controller to realize the system stable. As the nonlinear part is input and output measurable, we use TSVR with wavelet kernel to estimate.

Considering that $f_n(x)$ and $\hat{f}_n(x)$ are close enough, so we design a linear state feedback controller under the ideal condition as the expression $u = -Kx$, where K denotes a state feedback matrix which can be obtained by pole allocation. So the state equation of the close loop system can be written as $\dot{x} = Ax - BK^{\mathrm{T}}x + \varsigma(x)$. When the compensating error $\varsigma(x)$ is approximately equal to zero, the system is controllable and we can select an adaptive matrix to make the controlled system to have good performance. When the compensating error can not be omitted, the following theorem gives the asymptotic stable condition.

Theorem 4. Assuming the balance point of the controlled system is the original point, the system is asymptotically stable in the balance point, if and only if the condition $\frac{\|\varsigma(x)\|_2}{\|x\|_2} \to 0$ holds when $\|x\|^2 \to 0$ [17].

We use TSVR to obtain the approaching function, so that it counteracts and then a feedback controller is designed to make the close loop system stable at the expected pole location. The whole frame of state feedback control with TSVR nonlinear compensating is showed on the Fig. #1.

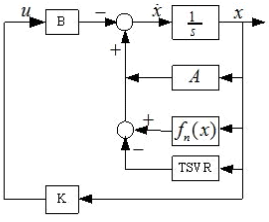


Fig. 1. State feedback control based on TSVR nonlinear compensating

4 Simulation

In this section, we validate the state feedback control based on TSVR nonlinear compensating on a nonlinear system example. The studied plant has the following state equation,

$$\dot{x} = \begin{bmatrix} -1 & 0 & 1 \\ 1 & 0 & 0.1 \\ 0 & 0 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} u + \begin{bmatrix} sin(x_1) + cos(x_2) + cos(x_3) \\ sin(x_1)sin(x_2)cos(x_3) \\ x_1 x_2 x_3 \end{bmatrix}. \tag{19}$$

The nonlinear state equation can be considered that three nonlinear thoroughfares add to the linear term. Based on the above assumption, we can obtain input and output data from the nonlinear terms. Then we estimate every nonlinear item by a TSVR model. We give the nonlinear thoroughfare white noises as the input and the nonlinear output also can be obtained. For the studied system, we estimate nonlinear functions of all three thoroughfares. For the first thoroughfare, we obtain a set with 400 data points, 200 data points for training and other 200 data points for testing the generalization of the TSVR model. The wavelet kernel is the following form, $k(x,y) = \Pi_{i=1}^{n} \left(cos(\theta \frac{x_i - y_i}{a_i}) \right) e^{-\frac{\|x_i - y_i\|^2}{a_i^2}}$, where parameters $\theta = 1.5$ and $\alpha_i = 3$. We select $c_1 = c_2 = 1.5$ and $\varepsilon_1 = \varepsilon_2 = 0.1$ as parameter values in TSVR. The training curves, including the lower boundary curve in black dashed line, the up boundary curve in red dashed line and the ending training curve in blue real line, are shown in the Fig. #2. Then we use other 200 data points to test the nonlinear model approached by TSVR as shown in the second figure of Fig. #2, where estimated values of testing points in blue color and expected values in red color. We can observe that the estimated values are close to the expected values with very small errors which can be considered as zeros.

For the second and the third thoroughfare, we similarly obtain 400 data points and 200 for training and the other 200 for testing. All parameters in TSVR are the same as the frontal declaration. The training and testing of the second and the third thoroughfare are executed like the first one.

In order to show the approaching ability of TSVR with wavelet kernel, we give the training and the testing results. We use the index of root square mean

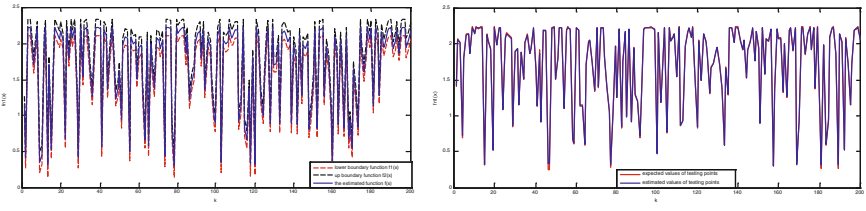


Fig. 2. The training and testing curves of the first thoroughfare by TSVR

error (RMSE) and mean absolute error (MAE). $RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - f_i)^2}{n}}$ and $MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - f_i|$. Table 1 gives the experimental compare between TSVR and SVR with wavelet kernels. From that, we can observe that TSVR method for modeling usually has better accuracy and less computational cost than SVR method.

Table 1. Modeling results by SVR and TSVR with wavelet kernels

		Channel 1		Channel 2		Channel 3	
		Training	Testing	Training	Testing	Training	Testing
SVR	RSME	0.0380	0.0354	0.0066	0.0072	0.0447	0.0481
	MAE	0.0284	0.0262	0.0052	0.0058	0.0333	0.0345
	Time	1.1s		0.7s		0.5s	
TSVR	RSME	0.0153	0.0166	0.0074	0.0079	0.0127	0.0135
	MAE	0.0111	0.0112	0.0057	0.0058	0.0094	0.0092
	Time	0.532s		0.631s		0.723s	

Assuming that the initial point $x(0) = [1, -2, 3]$, we design a feedback controller $u = -Kx$ to make the system stable at the original point, where $K = [0.855, 3.521, 0.624]$ and the pole points of the system are -5, $-2 + j$ and $-2 - j$. The left one in the Fig. #3 gives the convergence curves from the initial point to the original point under the state feedback control based on TSVR nonlinear compensating. We compare the method performance with the performance of SVR method. And then we also show state curves by dot and dash lines for the system without any nonlinear compensating. Obviously, the compensated system can be controlled to the state original point by feedback, but the nonlinear system without compensating can't get to the original point under a feedback controller. And the performance with TSVR compensating is better than with SVR compensating. The ideal result for nonlinear compensating is to counteract the nonlinear term to become a linear system absolutely. In simulation, we cancel the nonlinear term to execute feedback control on the linear system directly. The right one in the Fig. #3 illustrates the control performance. Hereby, the state feedback control can't work well on the nonlinear system without compensating, and the TSVR compensating method can change the control of nonlinear systems into a simple control of approximate linear system. The method is good performance and very valuable in practice.

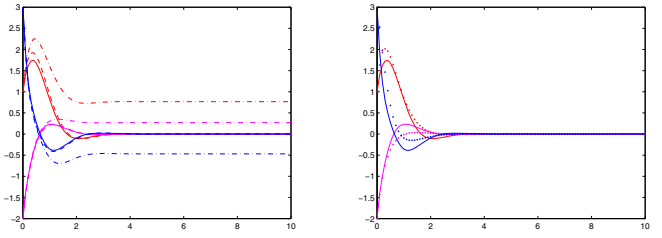


Fig. 3. The left figure is convergent curves with feedback control, TSVR compensating by with real lines, SVR compensating by dash lines and non-compensating by dot and dash lines. The right one is comparable curves between feedback control with TSVR compensating by real lines and direct feedback control by dot lines.

According to Theorem 4, when the states run to zeros, the term $\|\varsigma(x)\|^2$ also runs to zeros, the system is stable. We check the error curves of nonlinear compensating by the TSVR. From the Fig. #4, we can see the compensating error curves run gradually to zeros accompanying with the states to zeros, which is corresponding to Theorem 4. So we can judge that the close loop system of state feedback control with TSVR nonlinear compensating is stable.

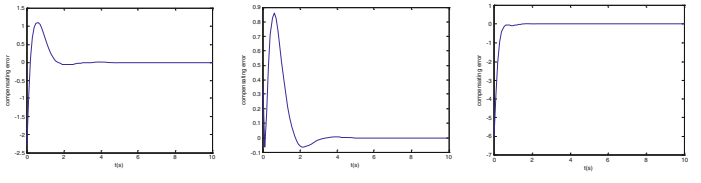


Fig. 4. The compensating error curves of three nonlinear thoroughfares from left to right, respectively

5 Conclusion

In this paper, we have introduced a new twin support vector regression algorithm with a wavelet kernel. A large quadratic programming problem in SVR is changed into two smaller quadratic programming problems in TSVR, which leads to still good approximate ability and generalization, but less computational complexity. Furthermore, for a class nonlinear system, we adopt TSVR with a wavelet kernel to approach the nonlinear part and the estimated model functionally counteracts the nonlinear part, which makes state feedback control available. In the simulation section, the TSVR method shows good approximation to the nonlinear part of system and the compensated system is stable, which validates the proposed method effective and good performance.

Acknowledgment

This work is partly supported by National Nature Science Foundation under Grant 61034002 and 60874013, and partly by the Scientific research Foundation of Graduate of Southeast University.

References

1. Vapnik, V.N.: The natural of statistical learning theory. Springer, New York (1995)
2. Burges, C.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2, 1–43 (1998)
3. Cortes, C., Vapnik, V.N.: Support vector networks. Machine Learning 20, 273–297 (1995)
4. Sun, C.Y., Mu, C.X., Li, X.M.: A weighted LS-SVM approach for the identification of a class of nonlinear inverse systems. Science in China Series F: Information Sciences 52(5), 770–779 (2009)
5. Mangasarian, O.L., Wild, E.W.: Multisurface proximal support vector classification via generalized eigenvalues. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(1), 69–74 (2006)
6. Jayadeva, R., Khemchandani, Chandra, S.: Twin support vector machines for pattern classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(5), 905–910 (2007)
7. Ghorai, S., Mukherjee, A., Dutta, P.K.: Nonparallel plane proximal classifier. Signal Processing 89, 510–522 (2009)
8. Kumar, M.A., Gopal, M.: Least squares twin support vector machines for pattern classification. Expert Systems with Applications 36, 7535–7543 (2009)
9. Kumar, M.A., Gopal, M.: Application of smoothing technique on twin support vector machines. Pattern Recognition Letter. 29, 1842–1848 (2008)
10. Peng, X.: TSVR: An efficient twin support vector machine for regression. Neural Networks 23(3), 365–372 (2010)
11. Kim, K.B., Park, J.B., Choi, Y.H., et al.: Control of chaotic dynamical systems using radial basis function network approximators. Information Sciences 130, 165–183 (2000)
12. Saunders, C., Gammerman, A., Vovk, V.: Ridge regression learning algorithm in dual variables. In: Proceeding of 15th International Conference on Machine Learning, pp. 515–521 (1998)
13. Smola, A., Scholkopf, B., Muller, K.R.: The connection between regularization operators and support vector kernels. Neural Networks 11, 637–649 (1998)
14. Zhang, L., Zhou, W.D., Jiao, L.C.: Wavelet support vector machine. IEEE Transaction on Systems, Man, and Cybernetics-Part B: Cybernetics 34, 1–6 (2003)
15. Yu, Z.H., Cai, Y.L.: Least squares wavelet support vector machines for nonlinear system identification. In: Wang, J., Liao, X.-F., Yi, Z. (eds.) ISNN 2005. LNCS, vol. 3497, pp. 436–441. Springer, Heidelberg (2005)
16. Liu, H., Liu, D., Ren, H.P.: Chaos control based on least square support vector machines. Acta Physica Sinica 54(9), 4019–4025 (2005)
17. Brauer, F., Nohel, J.A.: The qualitative theory of ordinary differential equations: an introduction. Dover Publications, Dover (1989)

Genetic Dynamic Fuzzy Neural Network (GDFNN) for Nonlinear System Identification

Mahardhika Pratama[1,*], Meng Joo Er[1], Xiang Li[2], Lin San[1], J.O. Richard [1], L.-Y. Zhai[1], Amin Torabi[1], and Imam Arifin[3]

[1] School of Electrical and Electronic Engineering, Nanyang Technological University, Nanyang Avenue, 639798, Singapore
MAHARDHI1.@e.ntu.edu.sg, EMJER@ntu.edu.sg
[2] Singapore Institute of Manufacturing Technology, 71 Nanyang Drive, 638075, Singapore
xli@SIMTech.a-star.edu.sg
[3] Electrical Engineering Department, Sepuluh Nopember Institute of Technology, Surabaya, Indonesia
arifin-i@its.ac.id

Abstract. This paper discusses an optimization of Dynamic Fuzzy Neural Network (DFNN) for nonlinear system identification. DFNN has 10 parameters which are proved sensitive to the performance of that algorithm. In case of not suitable parameters, the result gives undesirable of the DFNN. In the other hand, each of problems has different characteristics such that the different values of DFNN parameters are necessary. To solve that problem is not able to be approached with trial and error, or experiences of the experts. Therefore, more scientific solution has to be proposed thus DFNN is more user friendly, Genetic Algorithm overcomes that problems. Nonlinear system identification is a common testing of Fuzzy Neural Network to verify whether FNN might achieve the requirement or not. The Experiments show that Genetic Dynamic Fuzzy Neural Network Genetic (GDFNN) exhibits the best result which is compared with other methods.

Keywords: Dynamic Fuzzy Neural Network, Fuzzy Neural Network, Genetic Dynamic Fuzzy Neural Network, Genetic Algorithm.

1 Introduction

The basic idea to unite the NN and fuzzy logic controller is emerged by R.J.Jang to establish Adaptive Network Using Fuzzy Inference System (ANFIS). The structure of fuzzy logic with neural network architecture is combined, thus the difficulties in the obtaining the shape of membership function and suitable rule fuzzy logic controller are handled, because the principle of learning on the neural network is utilized. In the other hand, the problem to find structure of NN can be overcome due to IF-THEN rules of fuzzy logic.

The integrated neuro-fuzzy system combines advantages of both NN and FIS. Application of both technologies are categorized into following four cases [1] :

* Corresponding author.

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 525–534, 2011.

- A NN is used to automate the task of designing and fine tuning the membership functions of fuzzy systems.
- Both fuzzy inference and neural network learning capabilities acting separately.
- A NN is worked as correcting mechanisms for fuzzy systems.
- A NN is customized the standard system according to each users preferences and individual needs.

Applications of ANFIS controller have been done in the several purposes such as: they used ANFIS with PSO to control velocity control of DC motor [2], ANFIS was used as controller unmanned air vehicle [3], they developed ANFIS as stability controller of inverted pendulum [4], ANFIS was compared with radial basis function neuro fuzzy with hybrid genetic and pattern search algorithm [5].

The combination of neuro fuzzy which determines Gaussian membership function is called fuzzy neural network. The researches conducted in the area of fuzzy neural network are appeared with different objectives, such as: proposes an near optimal learning principle [6], extends the fuzzy neural network to be recurrent [7]. However, majority of the researches employ back propagation to be invoked in the learning phase. Back Propagation is definitely slow to find global optima [8]. Even, BP is often trapped in the local optima value. Some Research establishes hybrid learning or even learning using evolutionary computation. An underlying thing, Evolutionary Computation relies on random value, such that the learning time is also high to be employed in the learning phase, in addition if many values need to be obtained. However, to accomplish optimization problems, evolutionary computation remains a reasonable solution. Wu Shi Qian and Er Meng Joo deal with the principle of Dynamic Fuzzy Neural Network, which uses hierarchical learning approached instead of BP, is to be a function identifier. It also works to be noise cancellation [9].

Genetic algorithm with the genetic principle (mutation, crossover) to produce next generation is reasonable solution to solve optimization problem. Using objective function to be the representation of aim, and Applying genetic operation with certain probabilities, optimal value may be able to be acquired. Simplicity of the concept makes genetic algorithm to be widely implemented into the real world problems.

Parameters of Dynamic Fuzzy Neural Network (DFNN) are proved sensitive. Not suitable values may result bad performance of the DFNN, in the other hand different problems require different parameters. Expert experience and trial error could be deemed, nevertheless those are not really the solutions. A scientific approach using Genetic Algorithm (GA) is proposed so that it always guarantees that every implementation into the variety problems always acquires the optimal performance of DFNN.

This paper is organized as follows: Section 2 describes the literature review of dynamic fuzzy neural network including a learning principle, Genetic Algorithm principle. Section 3 explains the idea of the Genetic Dynamic Fuzzy Neural Network (GDFNN). Simulation and discussion are bravely exhausted in the Section 4. Several Conclusions are arranged in the rest of this paper.

2 Literature Review

This section is enhanced the materials of DFNN including structure of the DFNN, the criteria to generate neurons, learning principle, and pruning technology. Genetic algorithm application that is the concern of this paper also brightly discussed.

2.1 Structure of DFNN

DFNN is consisted of 5 layers which are input layer, Membership Function (MF), hidden layer, normalized layer, and output layer. Output layer gives Takagi Sugeno Kang (TSK) model which is utilized by ANFIS. MF's are determined as Gaussian function which is one of the Radial Basis Function (RBF) function. For Simplicity, This paper only considers multi inputs, and one output case, nevertheless DFNN might be extended to be multi inputs, and multi outputs concurrently as shown on Fig.

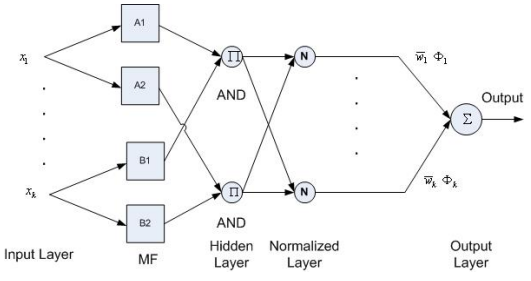


Fig. 1. The structure of DFNN

Layer 1. This layer doesn't perform any mathematical operation, it just pass inputs into the next layers (x_i , i=1, 2,….,k).

Layer 2. Inputs are processed in this layer which is mapped into the Gaussian functions, the numbers of MF are determined from the neurons generation criteria.

$$\mu_{ij} = -\frac{(x_i - c_{ij})^2}{\sigma_j^2}, i = 1,2,...,k \quad j = 1,2,....,u \tag{1}$$

Layer 3. This layer retrieves output of layer 2, and then those numbers are multiplied with output from other MF's respectively. This layer is commonly called rule layer. The outputs of this layer are normally called firing strength.

$$R_j = \exp(\frac{\sum_{i=1}^{k}(x_i - c_{ij})^2}{\sigma_j^2}) \tag{2}$$

Layer 4. This layer is called normalized layer which is processed the firing strengths into a range [0,1].

$$\Phi_j = \frac{R_j}{\sum_{x=1}^{u} R_x} \tag{3}$$

Layer 5. Outputs normalization are multiplied with weights vector which to retrieve one output signal from the output layer.

$$y = \sum_{j=1}^{u} w_j \Phi_j \tag{4}$$

w_j is able to be a constant or linear function as same as TSK model, for TSK model, it could be rewritten as stated on the Equation 5.

$$w_j = k_{j0} + k_{j1} x_1 + ... + K_{jk} x_k, \ j = 1,2,...., u \tag{5}$$

For constant case, weight can be considered as :

$$w_j = c_j \tag{6}$$

2.2 Learning Principle of DFNN

The allocation of the RBF unit is important to give a significant output. The idea of DFNN creates criteria of allocations of RBF units DFNN so that they can give to cover input space. The structure of DFNN is not able to be determined in a prior, therefore it is able to automatically generate the RBF units which can establish a structure of DFNN, thereby good coverage of RBF units can be achieved. There are two underlying concepts in the learning of DFNN, they are criteria of neuron generation, and Hierarchical learning.

Neuron Generation Criterion. This criterion describes when the neuron should be added or not in order to have feasible structure of DFNN. First factor should be considered in which the system error is greater that a pre-determined value thus the neuron should be adjusted. It can be formulated as follows:

$$e_i = \left\| t_i - y_i \right\| \tag{7}$$

t_i is a vector of target value, y_i is a vector of actual output value.

$$e_i > k_{de} \tag{8}$$

if Equation 8 is satisfied, then the neuron would better to be added which k_{de} should be determined in a priori. Second factor is able to be derived with the concept of how close the input space with the center of RBF function is. It is able to be modeled in the mathematical form as follows:

$$d_i(j) = \left\| X_i - C_j \right\|, \; j = 1,2,....,u \tag{9}$$

In which X_i, C_j are the vectors of input and center respectively.

If

$$\arg\min d_i(j) > k_d \tag{10}$$

Then neuron should be added. $\arg\min d_i(j)$ is called $d_{\min}$.

Hierarchical Learning. The fundamental idea of this concept is the accommodation boundary of each RBF unit is not fixed but changed dynamically based on the following manner: at the first time the parameters is set to be large to acquire rough but they are global values, after that they are decreasing monotonically. It is able to be implemented in these expressions:

$$k_e = \max[e_{\max} \times \beta^i, e_{\min}] \tag{11}$$

$$k_d = \max[d_{\max} \times \gamma^i, d_{\min}] \tag{12}$$

The delighted result are going to be retrieved if k_e, k_d are close to $e_{\min}, d_{\min}$ respectively.

After the neurons have been generated, the values of those need to be assigned. From the observations of Wu Shi Qian [8], the width plays important role. If the width is less than the distance between centers and inputs then the DFNN doesn't give a meaningful output, however if the width is too large then the firing strength will give value nearby 1, therefore the width and center are tuned as follows :

$$X_i = C_i \tag{12}$$

$$\sigma_i = k \times \sigma_0 \tag{13}$$

k is the overlap factor which is determined by overlap response of the RBF units. At the first observation width need to be set $\sigma_1 = \sigma_0$, σ_0 is predetermined value. those, which have been aforementioned, are the case when $\left\| e_i \right\| > k_e$, and $k_d < d_{\min}$. There are there other cases which are considered.

$\left\| e_i \right\| \le k_e$, $k_d \ge d_{\min}$, it implies good result, nothing is done. $\left\| e_i \right\| \le k_e$, $k_d < d_{\min}$, this condition pretend to only adjust weight. $\left\| e_i \right\| > k_e$, $k_d \ge d_{\min}$ the width of the nearest RBF nodes and all weights should be updated. The nearest $z-th$ RBF nodes are updated with following manner:

$$\sigma^i{}_z = k \times \sigma_z{}^{i-1} \tag{14}$$

k is predefined constant

For updating weights, it can be simply found with employing Φ^+ which is pseudoinverse of Φ.

$$W = T.\Phi^+ \tag{15}$$

$$\Phi^+ = (\Phi^T.\Phi)^{-1}\Phi^T \tag{16}$$

Comparing with back propagation algorithm, this method is much simple such that it is able to reduce the computational time therefore it is feasible to be applied in the real time applications.

Sometime the neurons gives good contributions to the output of the neurons, however sometime they don't contribute well, thus it leads to utilize pruning technology. The less contribution neurons are deleted.

$$\eta_i = \sqrt{\frac{\delta_i.\delta_i^T}{r+1}} \tag{17}$$

If

$$\eta_i < k_{err} \tag{18}$$

then the neurons are deleted.

To be more detail according to mathematical derivations, those are revealed in [8]. This method is called Error Reduction Ratio (ERR).

2.3 Genetic Algorithm

Genetic Algorithm is a powerful tool to accomplish optimization problems based on the principle of genetic operators. It could guarantee that good values are resulted. The principle of genetic algorithm is at the first time it generates numbers of random variables called chromosomes, each chromosomes are consisted of gen. Numbers of chromosome used during the process are specified beforehand. This paper describes chromosome is organized from variables that are desired to be optimized. Rely on the fixed probability, selection process is conducted, the manner of selections are actually miscellaneous, this paper is utilized Roulette Wheel principle which is probability of the selection depends on the number of fitness function. Chromosomes selected are going to be processed using genetic operator. Elitist concept is also considered which employs the fittest chromosome to be a parent. Genetic processes are crossover and mutation. Uniform crossover is used here. The process of uniform crossover is explained as follows:

Step 1: Put the two parents together
Step 2: Swap Gen on chromosome with fixed probability

The numbers of times uniform crossover (M) is calculated as follows:

$$M = R.N \tag{19}$$

R is actually recombination rate, and N is numbers of chromosome. Uniform Crossover don't use chromosome from the elitist. Second Operator is called mutation, Mutation operator replaces the chosen parameter from the random chromosomes into

the random value. This process is iteratively applied until the randomly generated numbers bigger than mutation rate. The evaluation function chosen exhibits on Equation 20.

$$Fitness(i) = \frac{k_0}{\sqrt{k_1.RMSE(i) + k_2.Num(i) + k_3.Time(i)}} \tag{20}$$

RMSE, num, time are root mean square error, number of used rules, and learning time in the i^{th} iteration. In this paper, k_0, k_1, k_2, k_3 are set to be 10^4,10^8,1,10 . That fitness function is intended to get the optimum parameters such a way those are able to lead DFNN to has good accuracy, efficient structure, and short learning time. GA is conducted in the iterative manner until the stopping criteria are fulfilled. First stopping criteria is the iteration is going to be stopped if the result is already converge, Second stopping criteria is while there is no changes on average fitness in 5 generations, the process will be ended.

3 GDFNN

As revealed above, DFNN has 10 parameters which have to be determined earlier, those are $e_{\max}$, $e_{\min}$, $d_{\max}$, $d_{\min}$, β, γ, σ_0, k , k_w, k_{err} . The searching process is repeated until stopping criteria are satisfied. During the process, assigned parameters value from GA are used as learning parameter of DFNN, thereby RMSE, number of rule, and learning time are included to be evaluation parameters. Parameters obtained by GA are as: $e_{\max} = 0.5297$, $e_{\min} = 0.4967$, $d_{\max} = 0.9270$, $d_{\min} = 0.8520$, $\beta, = 0.3463$ $\gamma = 0.5518$, $\sigma_0 = 0.53198$, $k = 0.9845$, $k_w = 0.7931$, $k_{err} = 0.9133$. For a fair comparison, the parameters of DFNN are obtained from its original paper [8].

The parameters of DFNN exhibits that with GA the optimal parameters are deviated from the original parameters, thus they are main reason that parameters of DFNN are sensitive and not able to be determined with trial error in order to get good generalization within training process. The optimization process's repeated until the stopping criteria are satisfied.

$d_{\max}$ = max of accommodation criterion
$d_{\min}$ = min of accommodation criterion
γ = decay constant
$e_{\max}$ = max of output error
$e_{\min}$ = min of output error
β = convergence constant
σ_0 = the width of the first rule
k = overlap factor of RBF units
k_w = width updating factor
k_{err} = significance of a rule

4 Simulation and Discussion

At this section, the proposed technique is applied in the non linear system identification which is a common evaluation in the testing of Fuzzy Neural Network (FNN). The GDFNN is compared with Genetic Dynamic Fuzzy Neural Network Back Propagation (GDFNNBP) which is actually DFNN with learning principle using back propagation, and DFNN without optimization phase. The aims are verified that genetic algorithm is able to improve the performance of DFNN, and Hierarchical learning, which is the main idea of DFNN, is still better than learning via back propagation.

4.1 Non Linear System Identification

The plant identified is second order highly nonlinear difference function which is defined in the Equation 21. The identification technique is utilized Seri-Parallel method that guarantees the stability of the estimated system. Sinusoidal input is applied to the system defined by Equation 23.

$$y(t+1) = \frac{y(t)\,y(t-1)[y(t)+2{,}5]}{1+y^2(t)+y^2(t-1)} + u(t) \tag{21}$$

$$\hat{y}(t+1) = f(y(t), y(t-1), u(t)) \tag{22}$$

$$u(t) = \sin(2\pi/25) \tag{23}$$

The learning result including Root Mean Square Error (RMSE), and neuron generation are shown on the Fig 2 and Fig 3.

From the Fig 2, that is clear that the proposed method shows a superior result, RMSE of GDFNN is smallest compared with GDFNNBP, and DFNN. GDFNN is fastest to reach the smallest error. The second best method's DFNN. The worst method's GDFNNBP. From Fig 3, GDFNN is also fastest to establish the feasible network structure compared with DFNN, and less rule than GDFNNBP. GDFNN

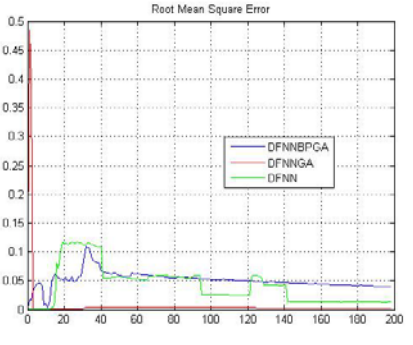


Fig. 2. Root Mean Square Error (RMSE)

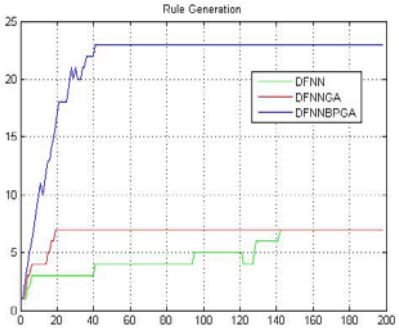


Fig. 3. Rule Generations

acquires the most accurate result, and less rules comparing with other methods. It also verified that all of the learning procedures are worked well, thereby accurate, and efficient network could be achieved. Even though, GDFNNBP acquires a shortest learning time, however the worst RMSE is got, it is reasonable because back propagation often retrieves just local minima values not a global optima. DFNN actually results enough to be employed in the case of Nonlinear System Identification, however, Using GA shows much improvements. For the testing, Mean Absolute Percentage Error (MAPE) is utilized to measure the prediction accuracy of the all methods. Table 2 exhibits the details of the learning and testing result. Fig 4 shows the prediction of GDFNN.

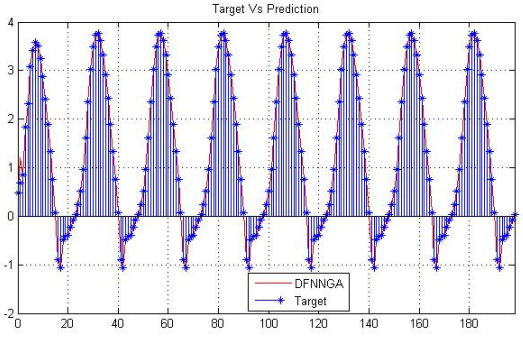


Fig. 4. Testing of GDFNN

Table 2. The Parameters of Evaluation

	TIME	RMSE	RULE	MAPE
DFNN	1.1431	0.0135	7	0.0793
GDFNN	1.3232	0.0025	7	0.0048
GDFNNBP	0.4854	0.041	23	0.086

From Table 1, GDFNN shows the longest training time, nevertheless there are much different in the RMSE, and MAPE. Between DFNN, and GDFNNBP acquire almost same performance, however DFNN need less rule than DFNNBP.

5 Conclusions

All of the methods used in this paper actually are feasible to be used to solve real problems, it can be seen from the MAPE that less than 0.1. However, to look for the best methods, thus GDFNN is the best method. The underlying thing that suggested to employ GA, GA is generated with the random numbers principle so that the searching process is somewhat long, nevertheless GA always guarantee the optimal numbers will be resulted as long as the determined of GA parameters are correct.

Acknowledgments. This project is fully supported with A-Star Science and Engineering Research Council (SERC) Singapore and Poland. The author personally thanks to Singapore Institute of Manufacturing Technology (SIMTech) to the opportunity to join within this project.

References

1. Bawane, N., Kothari, A.G., Kothari, D.P.: ANFIS Based on HVDC Control and Fault Identification of HVDC Converter. HAIT Journal of Science and Engineering B 2(5-6), 673–689 (2005)
2. Allaoula, B., Laoufi, A., Gasbaoui, B., Abderahmani, A.: Neuro Fuzzy DC Motor Speed Control Using PSO. Leonardo Electronic Journal of Practices and Technologies, 1–18 (2009) ISSN
3. Kurnaz, S., Cetin, O., Kaynak, O.: ANFIS Based on Autonomous Flight Control of UAV. Expert System with Applications (2010)
4. Saifizul, A.A., Zainon, Z., Abu Osman, N.A., Azlan, C.A., Ungku Ibrahim, U.F.S.: Intelligent Control for Self Erecting Pendulum via ANFIS. American Journal of Applied Sciences (2006) ISSN
5. Mazhari, S.A., Kumar, S.: Hybrid GA tuned RBF Based Neuro Fuzzy for Robotic Manipulator. International Journal of Electrical, Computer, and Systems Engineering (2008)
6. Wang, C.-H., Liu, H.-L., Lin, C.-T.: Dynamic Optimal Learning Rates of a Certain Class ofFuzzy Neural Networks and its Applications with Genetic Algorithm. IEEE Trans. on Systems, Man, and Cybernetics—Part B: CYBERNETICS 31(3) (June 2001)
7. Lin, C.-J., Chen, C.-H.: A compensation-based recurrent fuzzy neural network for dynamic system identification. European Journal of Operational Research 172, 696–715 (2006)
8. Wu, S.Q., Er, M.J.: Dynamic Fuzzy Neural Networks—A Novel Approach to Function Approximation. IEEE Trans. on System, Man, and Cybernetics—Part B: CYBERNETICS 30(2) (April 2000)
9. Er, M.J., Aung, M.S.: Adaptive Noise Cancellation Using Dynamic Fuzzy Neural Networks Algorithm. IFAC, Barcelona, Spain (2002)

Adaptive Robust NN Control of Nonlinear Systems

Guo-Xing Wen[1], Yan-Jun Liu[1], and C.L. Philip Chen[2]

[1] School of Sciences, Liaoning University of Technology,
Jinzhou, Liaoning, 121001, P.R. China
[2] Faculty of Science and Technology, University of Macau,
Av. Padre Tomás Pereira, S.J., Taipa, Macau, S.A.R., P.R. China
gxwen@live.cn, liuyanjun@live.com, philip.chen@ieee.org

Abstract. A direct adaptive neural networks (NNs) control based on the backstepping technique is proposed for uncertain nonlinear discrete-time systems in the strict-feedback form. The NNs are utilized to approximate unknown functions, and a stable adaptive neural backstepping controller is synthesized. The fact that all the signals in the closed-loop system are semi-globally uniformly ultimately bounded (SGUUB) is proven so that it is clear that the tracking error converges to a small neighborhood of zero by choosing the design parameters appropriately. Compared with the previous research for discrete-time systems, the proposed algorithm improves the robustness of the closed-loop system. Therefore, it ensures the feasibility of the control method.

Keywords: Neural networks, nonlinear systems, adaptive robust control.

1 Introduction

Since the universal approximation of the NNs and the fuzzy logic systems is proved [1,2], the NNs have become an active research topic and obtained widespread attention and they have been a powerful tool for stabilizing complex nonlinear dynamic systems [3-5].

Robustness in adaptive control has been an active topic of research in continuous-time [6-8].However, all these elegant methods in continuous-time domain are not directly applicable to discrete-time systems because the Lyapunov design for discrete-time becomes much more intractable than in the continuous-time and the noncausal problem in the controller design procedure via backstepping. So the linearity property of the derivative of a Lyapunov function in continuous-time is not present in the difference of Lyapunov function in the discrete-time. For several classes of discrete-time systems, some significant results are proposed in [9-13]. But, in [9-13], it is assumed that unknown bounds of the NN approximation error are less than bounded constants, if unknown bounds are larger than the assumed bounds, no performance of systems is guaranteed.

In this paper, we try to address an adaptive NN control of uncertain nonlinear systems in the discrete-time form. In the controller design, the NNs are used to approximate the unknown functions. Compared with previous study for the discrete-time system, the adaptive neural control design approach relaxes the restrictive

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 535–541, 2011.

assumptions that are usually made on the controlled nonlinear systems. Using the Lyapunov analysis method, all the signals in the closed-loop system are guaranteed to be SGUUB and the output tracks the reference signal to a bounded compact set.

2 System Dynamics and Preliminaries

Consider the following SISO discrete-time nonlinear systems

$$\begin{cases} \xi_i(k+1) = f_i\left(\bar{\xi}_i(k)\right) + g_i\left(\bar{\xi}_i(k)\right)\xi_{i+1}(k), i = 1,\cdots,n-1, \\ \xi_n(k+1) = f_n\left(\bar{\xi}_n(k)\right) + g_n\left(\bar{\xi}_n(k)\right)u(k), \\ y_1 = \xi_1(k), \end{cases} . \tag{1}$$

where $f_i\left(\bar{\xi}_i(k)\right)$, $g_i\left(\bar{\xi}_i(k)\right)$ are unknown functions; $\bar{\xi}_i(k) = \left[\xi_1(k),\cdots,\xi_i(k)\right]^T$, $u(k) \in R$ and $y_1 \in R$ are the state variable, the system input and the system output.

Assumption 1. The signs of $g_i\left(\bar{\xi}_i(k)\right), i = 1,2,\cdots,n$ are known and there exist constants $\underline{g}_i > 0$ and $\bar{g}_i > 0$ such that $\underline{g}_i \le \left|g_i\left(\bar{\xi}_i(k)\right)\right| \le \bar{g}_i, \forall \bar{\xi}_n(k) \in \Omega \subset R^n$.

The control objective is to design an adaptive NN controller so that: i) all the signals in the closed-loop are SGUUB, and ii) the output follows the reference signal $y_d(k)$ to a small compact set where $y_d(k) \in \Omega_y, \forall k > 0$ is a known bounded function with $\Omega_y := \{\chi \mid \chi = \xi_1\}$. In this paper, it assumes that $\underline{g}_i \le g_i\left(\bar{\xi}_i(k)\right) \le \bar{g}_i$.

3 Adaptive NN Control Design and Performance Analysis

We will design an adaptive controller $u(k)$ so that the system output y_1 follows $y_d(k)$, and simultaneously guarantees $\bar{\xi}_n(k) \in \Omega, \forall k > 0$ under the condition that $\bar{\xi}_n(0) \in \Omega$. With the aim of transformation procedure in [12], the system (1) can be transformed into

$$\begin{cases} \xi_1(k+n) = F_1\left(\bar{\xi}_n(k)\right) + G_1\left(\bar{\xi}_n(k)\right)\xi_2(k+n-1) \\ \vdots \\ \xi_n(k+1) = f_n\left(\bar{\xi}_n(k)\right) + g_n\left(\bar{\xi}_n(k)\right)u(k) \\ y_k = \xi_1(k) \end{cases} . \tag{2}$$

For convenience of analysis and discussion, for $i = 1,2,\cdots,n-1$, let

$$F_i(k) = F_i\left(\bar{\xi}_n(k)\right), G_i(k) = G_i\left(\bar{\xi}_n(k)\right)$$

$$f_n(k) = f_n\left(\bar{\xi}_n(k)\right), g_n(k) = g_n\left(\bar{\xi}_n(k)\right)$$

From the definition of $G_i\left(\bar{\xi}_n(k)\right)$, $G_i\left(\bar{\xi}_n(k)\right)$ satisfies $\underline{g}_i \le G_i\left(\bar{\xi}_n(k)\right) \le \bar{g}_i$. Before further going, let $k_i = k-n+i, i=1,2,\cdots,n-1$ for convenient description.

Step 1: For $\eta_1(k) = \xi_1(k) - y_d(k)$, its n th difference is given by

$$\eta_1(k+n) = F_1(k) + G_1(k)\xi_2(k+n-1) - y_d(k+n). \tag{3}$$

Considering $\xi_2(k+n-1)$ as a fictitious control for (3), if we choose

$$\xi_2(k+n-1) = \xi_{2d}^*(k) = -\frac{1}{G_1(k)}\left[F_1(k) - y_d(k+n)\right]$$

It is obvious that $\eta_1(k+n) = 0$. Since $F_1(k)$ and $G_1(k)$ are unknown, they are not available for constructing a fictitious control $\xi_{2d}^*(k)$. Thus, we can use NN to approximate $\xi_{2d}^*(k)$ and it has $\xi_{2d}^*(k) = W_1^{*T} S_1\left(z_1(k)\right) + \varepsilon_{z1}\left(z_1(k)\right)$ where $z_1(k) = \left[\bar{\xi}_n^T(k), y_d(k+n)\right]^T \in \Omega_{z1} \subset R^{n+1}$.

Assumption 2. On the compact set Ω_{zi}, $\varepsilon_{z1}\left(z_1(k)\right)$ satisfies $\left|\varepsilon_{zi}\left(z_i(k)\right)\right| \le \delta_i$ where $\delta_i > 0$ is unknown constant.

Remark 1. Most of the analytical results in the adaptive NN control literature make the key assumptions that the approximation error is bounded by some constants [9-13], if the approximation error is larger than the assumed bounds, no performance of systems can be guaranteed. Assumption 2 relaxes these conditions by requiring only that the approximation error is bounded.

$\hat{W}_i(k)$ and $\hat{\delta}_i(k)$ are used to denote the estimations of W_i^* and δ_i, and let $\tilde{W}_i(k) = \hat{W}_i(k) - W_i^*$, $\tilde{\delta}_i(k) = \hat{\delta}_i(k) - \delta_i$.

Choose the following direct adaptive fictitious control as

$$\xi_2(k+n-1) = \xi_{2f}(k) + \eta_2(k+n-1) = \hat{W}_1^T(k) S_1\left(z_1(k)\right) + \hat{\delta}_1(k) + \eta_2(k+n-1)$$

Choose the adaptation law in the following

$$\hat{W}_1(k+1) = \hat{W}_1(k_1) - \Gamma_1\left[S_1\left(z_1(k_1)\right)\eta_1(k+1) + \sigma_1\hat{W}_1(k_1)\right]$$

$$\hat{\delta}_1(k+1) = \hat{\delta}_1(k_1) - B_1\left[\eta_1(k+1) + \beta_1\hat{\delta}_1(k_1)\right]$$

Then, we have

$$\eta_1(k+n) = F_1(k) - y_d(k+n) + G_1(k)\left[\hat{W}_1^T(k) S_1\left(z_1(k)\right) + \hat{\delta}_1(k) + \eta_2(k+n-1)\right]$$

Adding and subtracting $G_1(k)\xi_{2d}^*(k)$ and noting the above equation, we have

$$\eta_1(k+n) = G_1(k)\Big[\hat{W}_1^T(k)S_1(z_1(k)) + \hat{\delta}_1(k) + \eta_2(k+n-1) - \xi_{2d}^*(k)\Big] + G_1(k)\xi_{2d}^*(k) + F_1(k) - y_d(k+n)$$

Further, we have

$$\eta_1(k+n) = G_1(k)\Big[\tilde{W}_1^T(k)S_1(z_1(k)) + \hat{\delta}_1(k) - \varepsilon_{z_1}(z_1(k)) + \eta_2(k+n-1)\Big]$$

Consider the Lyapunov function candidate as follows

$$V_1(k) = \frac{1}{\bar{g}_1}\eta_1^2(k) + \sum_{j=0}^{n-1}\tilde{W}_1^T(k_1+j)\Gamma_1^{-1}\tilde{W}_1(k_1+j) + \sum_{j=0}^{n-1}B_1^{-1}\tilde{\delta}_1^2(k_1+j)$$

Using the fact that

$$\tilde{W}_1^T(k_1)S_1(z_1(k_1)) = \eta_1(k+1)/G_1(k_1) - \hat{\delta}_1(k) + \varepsilon_{z_1} - \eta_2(k)$$

Based on the above equations, it can be obtained that

$$\begin{aligned}\Delta V_1 &= \frac{1}{\bar{g}_1}\left(\eta_1^2(k+1) - \eta_1^2(k)\right) + \tilde{W}_1^T(k+1)\Gamma_1^{-1}\tilde{W}_1(k+1) - \tilde{W}_1^T(k_1)\Gamma_1^{-1}\tilde{W}_1(k_1)\\ &\quad + B_1^{-1}\tilde{\delta}_1^2(k+1) - B_1^{-1}\tilde{\delta}_1^2(k_1)\\ &\le -\frac{1}{\bar{g}_1}\eta_1^2(k+1) - \frac{1}{\bar{g}_1}\eta_1^2(k) - 2\varepsilon_{z_1}\eta_1(k+1) + 2\eta_2(k)\eta_1(k+1) - 2\sigma_1\tilde{W}_1^T(k_1)\\ &\quad \hat{W}_1(k_1) + \left(S_1(z_1(k_1))\right)^T\Gamma_1 S_1(z_1(k_1))\eta_1^2(k+1) + 2\sigma_1\left(\hat{W}_1^T(k_1)\right)\Gamma_1 S_1(z_1(k_1))\\ &\quad \eta_1(k+1) + \sigma_1^2\left(\hat{W}_1^T(k_1)\right)\Gamma_1\hat{W}_1(k_1) + 2\delta_1\eta_1(k+1) - 2\beta_1\tilde{\delta}_1(k_1)\hat{\delta}_1(k_1)\\ &\quad + B_1\eta_1^2(k+1) + 2B_1\beta_1\hat{\delta}_1(k_1)\eta_1(k+1) + B_1\beta_1^2\hat{\delta}_1^2(k_1)\end{aligned}$$

Using the facts that

$$\left(S_1(z_1(k_1))\right)^T\Gamma_1 S_1(z_1(k_1)) \le \gamma_1 l_1,\ -2\varepsilon_{z_1}\eta_1(k+1) \le \frac{\gamma_1}{\bar{g}_1}\eta_1^2(k+1) + \frac{\bar{g}_1\varepsilon_{z_1}^2}{\gamma_1}$$

$$2\sigma_1\left(\hat{W}_1^T(k_1)\right)\Gamma_1 S_1(z_1(k_1))\eta_1(k+1) \le \frac{\gamma_1 l_1}{\bar{g}_1}\eta_1^2(k+1) + \bar{g}_1\sigma_1^2\gamma_1\left\|\hat{W}_1(k_1)\right\|^2$$

$$2\tilde{W}_1^T(k_1)\hat{W}_1(k_1) = \left\|\tilde{W}_1(k_1)\right\|^2 + \left\|\hat{W}_1(k_1)\right\|^2 - \left\|W_1^*(k_1)\right\|^2$$

$$2\eta_1(k+1)\eta_2(k) \le \frac{\gamma_1}{\bar{g}_1}\eta_1^2(k+1) + \frac{\bar{g}_1}{\gamma_1}\eta_2^2(k)$$

$$2B_1\beta_1\hat{\delta}_1(k_1)\eta_1(k+1) \le B_1^2\eta_1^2(k+1) + \beta_1^2\hat{\delta}_1^2(k_1)$$

$$2\tilde{\delta}_1(k_1)\hat{\delta}_1(k_1) = \tilde{\delta}_1^2(k_1) + \hat{\delta}_1^2(k_1) - \delta_1^2,\ 2\delta_1\eta_1(k+1) \le \frac{B_1}{\bar{g}_1}\eta_1^2(k+1) + \frac{\bar{g}_1}{B_1}\delta_1^2$$

we obtain

$$\Delta V_1 \le -\frac{\rho_1}{2\overline{g}_1}\eta_1^2(k+1)-\sigma_1(1-\sigma_1\gamma_1-\overline{g}_1\sigma_1\gamma_1)\left\|\hat{W}_1(k_1)\right\|^2-\frac{\omega_1}{2\overline{g}_1}\eta_1^2(k+1)$$
$$-\beta_1(1-\beta_1-B_1\beta_1)\hat{\delta}_1^2(k_1)-\frac{1}{\overline{g}_1}\eta_1^2(k)+\theta_1+\frac{\overline{g}_1}{\gamma_1}\eta_2^2(k)$$

where $\rho_1=1-4\gamma_1-2\gamma_1 l_1-2\overline{g}_1\gamma_1 l_1$, $\omega_1=1-2B_1-2B_1\overline{g}_1-2B_1^2\overline{g}_1$ and $\theta_1=\overline{g}_1\delta_1^2/B_1+\beta_1\delta_1^2+\sigma_1\left\|W_1^*(k_1)\right\|^2+\overline{g}_1\varepsilon_{z_1}^2/\gamma_1$. Choose the design parameters to satisfy $\gamma_1<1/(4+2l_1+2\overline{g}_1 l_1)$, $B_1<1/(2+2\overline{g}_1+2B_1\overline{g}_1)$ and $\sigma_1<1/(\gamma_1+\overline{g}_1\gamma_1)$, $\beta_1<1/(1-B_1)$.

Step $i(2\le i\le n-1)$ **:** Following the same procedure as in step 1, define $\eta_i(k)=\xi_i(k)-\xi_{if}(k_{i-1})$ and choose the Lyapunov function candidate

$$V_i(k)=\eta_i^2(k)/\overline{g}_i+\sum_{j=0}^{n-i}\tilde{W}_i^T(k_i+j)\Gamma_i^{-1}\tilde{W}_i(k_i+j)+\sum_{j=0}^{n-2}B_i^{-1}\tilde{\delta}_i^2(k_i+j)$$

We have

$$\Delta V_i \le -\frac{\rho_i}{\overline{g}_i}\eta_i^2(k+1)-\sigma_i(1-\sigma_i\gamma_i-\overline{g}_i\sigma_i\gamma_i)\left\|\hat{W}_i(k_i)\right\|^2-\frac{\omega_i}{\overline{g}_i}\eta_i^2(k+1)$$
$$-\beta_i(1-\beta_i-B_i\beta_i)\hat{\delta}_i^2(k_i)-\frac{1}{\overline{g}_i}\eta_i^2(k)+\theta_i+\frac{\overline{g}_2}{\gamma_2}\eta_{i+1}^2(k)$$

where ρ_i,ω_i and θ_i is defined as $\rho_i=1-4\gamma_i-2\gamma_i l_i-2\overline{g}_i\gamma_i l_i$, $\omega_i=1-2B_i-2B_i\overline{g}_i-2B_i^2\overline{g}_i$ and $\theta_i=\overline{g}_i\delta_i^2/B_i+\beta_i\delta_i^2+\sigma_i\left\|W_i^*(k_i)\right\|^2+\overline{g}_i\varepsilon_{z_i}^2/\gamma_i$. Choose the design parameters to satisfy $\gamma_i<1/(4+2l_i+2\overline{g}_i l_i)$, $B_i<1/(2+2\overline{g}_i+2B_i\overline{g}_i)$ and $\sigma_i<1/(\gamma_i+\overline{g}_i\gamma_i)$, $\beta_i<1/(1-B_i)$.

Step n**:** Following the same procedure as in Step i, we choose the direct adaptive controller and the adaptation laws as

$$u(k)=\hat{W}_n^T(k)S_n(z_n(k))+\hat{\delta}_n(k)$$
$$\hat{W}_n(k+1)=\hat{W}_n(k_n)-\Gamma_n\left[S_n(z_n(k_n))\eta_n(k+1)+\sigma_n\hat{W}_n(k_n)\right]$$
$$\hat{\delta}_n(k+1)=\hat{\delta}_n(k_n)-B_n\left[\eta_n(k+1)+\beta_n\hat{\delta}_n(k_n)\right]$$

and obtain

$$\Delta V_n \le -\frac{\rho_n}{\overline{g}_n}\eta_n^2(k+1)-\sigma_n\left(1-\sigma_n\gamma_n-\overline{g}_n\sigma_n\gamma_n\right)\left\|\hat{W}_n(k_n)\right\|^2-\frac{\omega_n}{\overline{g}_n}\eta_n^2(k+1)$$
$$-\beta_n\left(1-\beta_n-B_n\beta_n\right)\hat{\delta}_n^2(k_n)-\frac{1}{\overline{g}n}\eta_n^2(k)+\theta_n$$

where ρ_n, ω_n and θ_n are defined as $\rho_n = 1-2\gamma_n-2\gamma_n l_n-2\overline{g}_n\gamma_n l_n$, $\omega_n = 1-2B_n-2B_n\overline{g}_n-2B_n^2\overline{g}_n$ and $\theta_n = \frac{\overline{g}_n}{B_n}\delta_n^2+\beta_n\delta_n^2+\sigma_n\left\|W_n^*(k_n)\right\|^2+\frac{\overline{g}_n\varepsilon_{z_n}^2}{\gamma_n}$. Choose the design parameters to satisfy $\gamma_n < 1/\left(2+2l_n+2\overline{g}_n l_n\right)$, $B_n < 1/\left(2+2\overline{g}_n+2B_n\overline{g}_n\right)$ and $\sigma_n < 1/(\gamma_n+\overline{g}_n\gamma_n)$, $\beta_n < 1/\left(1-B_n\right)$.

Theorem 1. Consider the closed-loop system consisting of system (1), controller u and adaptation laws $\hat{W}_i(k_i)$ and $\hat{\delta}_i(k_i)$, under Assumptions 1 and 2, and the bounded initial condition, and there exist compact sets $\Omega_{y0} \subset \Omega_y, \Omega_{\varphi 0} \subset \Omega_\varphi$, then, all the signals in the closed-loop system are SGUUB and the tracking error can be made arbitrarily small.

Proof. The proof is similar to that of the theorem 1 in [12,13] and will be omitted.

4 Conclusion

By using the backstepping technique and the approximation property of the neural networks, we have proposed an adaptive control approach for a class of uncertain discrete-time nonlinear systems. The approach can improve robustness of the closed-loop system. The adaptive controllers were obtained based on Lyapunov stability theory and all the signals of the resulting closed-loop system were guaranteed to be SGUUB, and the tracking errors can be reduced to a small neighborhood of zero.

Acknowledgements

The work was supported by the National Natural Science Funds of China under grant 61074014, 60874056 and The Foundation of Educational Department of Liaoning Province L2010181 and China Postdoctoral Science Foundation Special 200902241 and The Chinese National Basic Research 973 Program 2011CB302801 and Macau Science and Technology Development Fund 008/2010/A1.

References

1. Park, J., Sandberg, I.W.: Universal approximation using radial-basis-function network. Neural Computation 3, 246–257 (1991)
2. Wang, L.X.: Fuzzy systems are universal approximators. In: Proc. IEEE Int. Conf. Fuzzy Systems, San Diego, CA, pp. 1163–1170 (1992)

3. Plett, G.L.: Adaptive inverse control of linear and nonlinear systems using dynamic neural networks. IEEE Trans. Neural Netw. 14(3), 360–376 (2003)
4. Tong, S.C., Li, Y.M.: Observer-based fuzzy adaptive control for strict-feedback nonlinear systems. Fuzzy Sets and Systems 160(12), 1749–1764 (2009)
5. Tong, S.C., Li, C.Y., Li, Y.M.: Fuzzy adaptive observer backstepping control for MIMO nonlinear systems. Fuzzy Sets and Systems 160(19), 2755–2775 (2009)
6. Liu, Y.J., Wang, W., Tong, S.C., Liu, Y.S.: Robust Adaptive Tracking Control for Nonlinear Systems Based on Bounds of Fuzzy Approximation Parameters. IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans 40(1), 170–184 (2010)
7. Liu, Y.J., Wang, W.: Adaptive Fuzzy Control for a Class of Uncertain Nonaffine Nonlinear Systems. Information Sciences 177(18), 3901–3917 (2007)
8. Li, T.S., Tong, S.C., Feng, G.: A novel robust adaptive fuzzy tracking control for a class of nonlinear MIMO systems. IEEE Trans. on Fuzzy Systems 18(1), 150–160 (2010)
9. Yang, C.G., Li, Y., Ge, S.S., Lee, T.H.: Adaptive Control of a Class of Discrete-Time MIMO Nonlinear Systems with Uncertain Couplings. International Journal of Control 83(10), 2020–2133 (2010)
10. Yang, C.G., Ge, S.S., Lee, T.H.: Output Feedback Adaptive Control of a Class of Nonlinear Discrete-Time Systems with Unknown Control Directions. Automatica 45(1), 270–276 (2009)
11. Yang, C.G., Ge, S.S., Xiang, C., Chai, T., Lee, T.H.: Output Feedback NN Control for two Classes of Discrete-time Systems with Unknown Control Directions in a Unified Approach. IEEE Transactions on Neural Networks 19(11), 1873–1886 (2008)
12. Ge, S.S., Li, G.Y., Lee, T.H.: Adaptive NN control for a class of strict-feedback discrete-time nonlinear systems. Automatica 39(5), 807–819 (2003)
13. Ge, S.S., Yang, C.G., Lee, T.H.: Adaptive predictive control using neural network for a class of pure-feedback systems in discrete time. IEEE Trans. Neural Netw. 19(9), 1599–1614 (2008)

A Generalized Online Self-constructing Fuzzy Neural Network

Ning Wang*, Yue Tan, Dan Wang, and Shaoman Liu

Marine Engineering College, Dalian Maritime University,
No.1 Linghai Road, Dalian 116026, China
n.wang.dmu.cn@gmail.com

Abstract. We propose a Generalized Online Self-constructing Fuzzy Neural Network (GOSFNN) which extends the ellipsoidal basis function (EBF) based fuzzy neural networks (FNNs) by permitting input variables to be modeled by dissymmetrical Gaussian functions (DGFs). Due to the flexibility and dissymmetry of left and right widths of the DGF, the partitioning made by DGFs in the input space is more flexible and more interpretable, and therefore results in a parsimonious FNN with high performance under the online learning algorithm. The geometric growing criteria and the error reduction ratio (ERR) method are incorporated into structure identification which implements an optimal and compact network structure. The GOSFNN starts with no hidden neurons and does not need to partition the input space *a priori*. In addition, all free parameters in premises and consequents are adjusted online based on the Extended Kalman Filter (EKF) method. The performance of the GOSFNN paradigm is compared with other well-known algorithms like ANFIS, OLS, GDFNN, SOFNN and FAOS-PFNN, *etc.*, on a benchmark problem of multi-dimensional function approximation. Simulation results demonstrate that the proposed GOSFNN approach can facilitate a more powerful and parsimonious FNN with better performance of approximation and generalization.

Keywords: Fuzzy neural network, Online self-constructing, Extended Kalman filter, Dissymmetrical Gaussian function.

1 Introduction

Similar to the well-known ANFIS [1], the traditional design of fuzzy neural networks (FNNs) is to assume that membership functions have been defined in advance and the number of fuzzy rules is determined *a priori* according to either expert knowledge or trial and error method [2], and the parameters are modified by the hybrid or BP learning algorithm [3,4] which is known to be slow and easy to be entrapped into local minima. A significant contribution was made

* This work is financially supported by National Nature Science Foundation of China (under Grant 51009017 and 61074017), and Foundamental Research Funds for the Central Universities of China (under Grant 2009QN025).

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 542–551, 2011.

by Platt [5] through the development of the resource-allocating network (RAN) that adds hidden units to the network based on the novelty of the new data in the sequential learning process. And it is followed by some improved works [6,7]. Another seminal work was proposed by Chen *et al.* [2] that an orthogonal least square (OLS) learning algorithm is used to conduct both structure and parameter identification. In addition, several evolving FNNs, i.e. DENFIS [8] *etc.*, have been reported in the fruitful field. Specifically focusing on variants of geometric growing criteria [5], some typical self-constructing paradigms have been proposed lately [9]. The Dynamic Fuzzy Neural Network (DFNN) based on RBF neural networks has been developed in [10] in which not only the parameters can be adjusted by the linear least square (LLS) but also the structure can be self-adaptive via growing and pruning criteria. In [11], the DFNN is extended to a generalized DFNN (GDFNN) by introducing the ellipsoidal basis function (EBF). Similar to the GDFNN, a self-organizing fuzzy neural network (SOFNN) [12] with a pruning strategy using the optimal brain surgeon (OBS) approach has been proposed to extract fuzzy rules online. In addition, a fast and accurate online self-organizing scheme for parsimonious fuzzy neural networks (FAOS-PFNN) [13] based on the Extended Kalman Filter (EKF) method has been proposed to accelerate the learning speed and increase the approximation accuracy via incorporating pruning strategy into new growth criteria. A convincing improvement based on the LLS method has been presented to enhance the accuracy and compactness [14]. In the context of membership functions of the input variables, the asymmetric Gaussian function (AGF) [15,16] has been presented to upgrade the learning ability and flexibility of the FNN.

Based on the brief overview of the development of FNNs, we find that it is difficult to consider the balance between the compact structure and the high performance. In order to solve this dilemma, an intuitive and straightforward method is to enhance the descriptive capability of membership functions which can partition well the input space. Motivated by this idea, a dissymmetrical Gaussian function (DGF) is presented to extend symmetric Gaussian functions by permitting the input signal to be modeled by the DGF in this paper. We present a generalized online self-constructing fuzzy neural network (GOSFNN) which implements a TSK fuzzy inference system. Simulation results indicate that the GOSFNN paradigm can online facilitate a more compact FNN with better performance of approximation and generalization.

2 Architecture of the GOSFNN

2.1 Generalized Ellipsoidal Basis Function

Definition 1 (DGF). *If any function satisfies the following condition*

$$DGF\left(x; c, \sigma(x)\right) = \exp\left(-\frac{(x-c)^2}{\sigma^2(x)}\right) \tag{1}$$

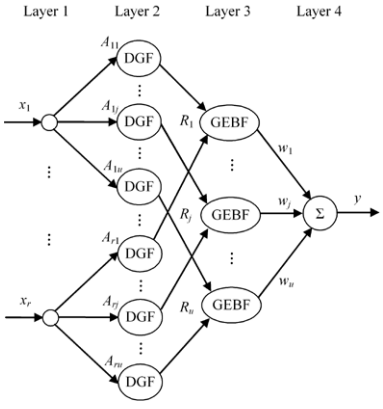


Fig. 1. Architecture of the GOSFNN

$$\sigma(x) = \begin{cases} \sigma_{right}, & x \geq c \\ \sigma_{left}, & x < c \end{cases} \tag{2}$$

where c, σ_{left} *and* σ_{right} *denote the center, left width and right width of the DGF, respectively. We call the function* $DGF(.)$ *dissymmetrical Gaussian function, where* $\sigma(.)$ *is called the dynamic width.*

Definition 2 (GEBF). *We call the function defined in (3) generalized ellipsoidal basis function.*

$$GEBF\left(\boldsymbol{X}; \boldsymbol{C}, \boldsymbol{\Sigma}(\boldsymbol{X})\right) = \prod_{i=1}^{n} DGF\left(x_i; c_i, \sigma_i(x_i)\right) \tag{3}$$

where $\boldsymbol{X} = [x_1, x_2, \cdots, x_n]^{\mathrm{T}}$ *and* $\boldsymbol{C} = [c_1, c_2, \cdots, c_n]^{\mathrm{T}}$ *denote input vector and center vector, respectively. And,* $\boldsymbol{\Sigma}(\boldsymbol{X}) = [\sigma_1(x_1), \cdots, \sigma_n(x_n)]^{\mathrm{T}}$ *is called dynamic width vector, where the dynamic width* $\sigma_i(x_i)$ *is defined by (2).*

2.2 Architecture of the GOSFNN

The GOSFNN shown in Fig.1 can be described by the following fuzzy rules:

$$\text{Rule } j: \text{ IF } x_1 \text{ is } A_{1j} \text{ and } ... \text{ and } x_r \text{ is } A_{rj} \text{ THEN } y = w_j,\ j = 1, 2, ..., u\ . \tag{4}$$

where A_{ij} is the fuzzy set of the ith input variable x_i in the jth fuzzy rule, r and u are the numbers of input variables and fuzzy rules, respectively. Let μ_{ij} be the corresponding membership function of the fuzzy set A_{ij} shown in layer 2, where the foregoing defined DGF is used to model fuzzy sets.

Layer 1: The nodes in this layer denote input variables.

Layer 2: Each node represents a possible membership function as follows:

$$\mu_{ij}(x_i) = DGF\left(x_i; c_{ij}, \sigma_{ij}(x_i)\right) \tag{5}$$

$$\sigma_{ij}(x_i) = \begin{cases} \sigma_{ij}^R, \ x_i \geq c_{ij} \\ \sigma_{ij}^L, \ x_i < c_{ij} \end{cases} \tag{6}$$

where c_{ij}, σ_{ij}^L and σ_{ij}^R are the center, left width and right width of the corresponding fuzzy set, respectively.

Layer 3: Each node represents a possible IF-part of fuzzy rules. The output of the jth rule R_j can be calculated as follows:

$$\varphi_j(\mathbf{X}) = GEBF\left(\mathbf{X}; \mathbf{C}_j, \boldsymbol{\Sigma}_j(\mathbf{X})\right) \tag{7}$$

where $\mathbf{X} = [x_1, x_2, \cdots, x_r]^{\mathrm{T}}$, $\mathbf{C}_j = [c_{1j}, c_{2j}, \cdots, c_{rj}]^{\mathrm{T}}$, and $\boldsymbol{\Sigma}_j = [\sigma_{1j}(x_1), \sigma_{2j}(x_2), \cdots, \sigma_{rj}(x_r)]^{\mathrm{T}}$ denote input vector, center vector and dynamic width vector.

Layer 4: This layer has single output node for multi-input and single-output (MISO) systems. The output is the weighted summation of incoming signals,

$$y(\mathbf{X}) = \sum_{j=1}^{u} w_j \varphi_j \tag{8}$$

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

3 Learning Scheme of the GOSFNN

For each observation $(\mathbf{X}^k, t^k), k = 1, 2, \cdots, n$, where n is the number of total training data pairs, $\mathbf{X}^k \in \mathbf{R}^r$ and $t^k \in \mathbf{R}$ are the kth input vector and the desired output, respectively.

3.1 Criteria of Rule Generation

1) System Error: The system error can be calculated as follows:

$$\|e^k\| = \|t^k - y^k\|, \ k = 1, 2, \cdots, n \tag{9}$$

If

$$\|e^k\| > k_e, \ k_e = \max\{e_{\max}\beta^{k-1}, e_{\min}\} \tag{10}$$

a new GEBF hidden neuron should be created for high performance. Otherwise, no new fuzzy rules will be recruited and only the parameters of the existing fuzzy rules will be updated. Here, k_e is a predefined threshold that decays during the learning process, where $e_{\max}$ is the maximum error chosen, $e_{\min}$ is the desired accuracy and $\beta \in (0, 1)$ is the convergence constant.

2) Input Partition: The distance $dist_{jk}$ between the new coming sample $\mathbf{X}^k = [x_{1k}, x_{2k}, \cdots, x_{rk}]^{\mathrm{T}}$ and the center $\mathbf{C}_j = [c_{1j}, c_{2j}, \cdots, c_{rj}]^{\mathrm{T}}$ of the jth GEBF unit can be obtained as follows:

$$dist_{jk}(\mathbf{X}^k) = \sqrt{(\mathbf{X}^k - \mathbf{C}_j)^{\mathrm{T}} \mathbf{S}_j^{-1} (\mathbf{X}^k)(\mathbf{X}^k - \mathbf{C}_j)}, \ j = 1, 2, \cdots, u \tag{11}$$

$$\mathbf{S}_j(\mathbf{X}^k) = diag\left(\boldsymbol{\Sigma}_j^2(\mathbf{X}^k)\right) = \begin{pmatrix} \sigma_{1j}^2(x_{1k}) & 0 & \cdots & 0 \\ 0 & \sigma_{2j}^2(x_{2k}) & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_{rj}^2(x_{rk}) \end{pmatrix} \tag{12}$$

where $\boldsymbol{\Sigma}_j(\mathbf{X}^k)$ and $\sigma_{ij}(x_{ik})$ are dynamic width vector and corresponding dynamic widths of the ith dimension in the jth GEBF unit for the kth observation and can be derived from (6).

For the kth observation, find the nearest GEBF unit given by

$$J = \arg \min_{1 \le j \le u} (dist_{jk}(\mathbf{X}^k)) \tag{13}$$

If

$$dist_{jk}(\mathbf{X}^k) > k_d,\ k_d = \max\{d_{\max}\gamma^{k-1}, d_{\min}\} \tag{14}$$

existing GEBFs cannot partition the input space well. A new GEBF unit should be considered for fine partitioning in the input space. Otherwise, no new fuzzy rules will be recruited and only the parameters of existing fuzzy rules will be updated. Here, k_d is a predefined threshold that decays during the learning process, where $d_{\max}$ and $d_{\min}$ are the maximum and minimum distance, respectively.

3) Generalization Capability: The error reduction ratio (ERR) [3] is used to calculate the significance of fuzzy rules. Consider (8) as a special case of the linear regression model which can be described in the following compact form:

$$\mathbf{T} = \boldsymbol{\Psi}\mathbf{W} + \mathbf{E} \tag{15}$$

where $\mathbf{T} = [t^1, t^2, \cdots, t^n]^{\mathrm{T}} \in \mathbf{R}^n$ is the desired output vector, $\mathbf{W} = [w_1, w_2, \cdots, w_u]^{\mathrm{T}} \in \mathbf{R}^u$ is the vector of weights, $\mathbf{E} = [e^1, e^2, \cdots, e^n]^{\mathrm{T}} \in \mathbf{R}^n$ is the error vector which is assumed to be uncorrelated with the regressors, and $\boldsymbol{\Psi} = [\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \cdots, \boldsymbol{\psi}_n] \in \mathbf{R}^{n \times u}$ is the output matrix of layer 3 given by

$$\boldsymbol{\Psi} = \begin{pmatrix} \varphi_{11} & \cdots & \varphi_{u1} \\ \vdots & \ddots & \vdots \\ \varphi_{1n} & \cdots & \varphi_{un} \end{pmatrix} \tag{16}$$

For the matrix $\boldsymbol{\Psi}$, if its row number is larger than the column number, we can transform it into a set of orthogonal basis vectors by QR decomposition,

$$\boldsymbol{\Psi} = \mathbf{PQ} \tag{17}$$

where the matrix $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_v] \in \mathbf{R}^{n \times u}$ has the same dimension as the matrix $\boldsymbol{\Psi}$ with orthogonal columns and $\mathbf{Q} \in \mathbf{R}^{u \times u}$ is an upper triangular matrix. Substituting (17) into (15) yields

$$\mathbf{T} = \mathbf{PQW} + \mathbf{E} = \mathbf{PG} + \mathbf{E} \tag{18}$$

where $\mathbf{G} = [g_1, g_2, \cdots, g_u]^{\mathrm{T}} = (\mathbf{P}^{\mathrm{T}}\mathbf{P})^{-1}\mathbf{P}^{\mathrm{T}}\mathbf{T} \in \mathbf{R}^u$ could be obtained by the linear least square (LLS) method. An ERR due to $\mathbf{p}_i$ is given by

$$err_i = \frac{(\mathbf{p}_i^{\mathrm{T}}\mathbf{T})^2}{\mathbf{p}_i^{\mathrm{T}}\mathbf{p}_i\mathbf{T}^{\mathrm{T}}\mathbf{T}},\ i = 1, 2, \cdots, u \tag{19}$$

In order to define the significance of each fuzzy rule, we propose a novel growth criterion termed *generalization factor (GF)* for checking the generalization capability of the GOSFNN.

$$GF = \sum_{i=1}^{u} err_i \tag{20}$$

If $GF < k_{GF}$, where k_{GF} is the threshold, the generalization capability is poor and therefore the fuzzy neural network needs more hidden neurons to achieve high generalization performance. Otherwise, no hidden nodes will be created.

3.2 Parameter Adjustment

Note that the following parameter learning phase is performed on the entire system after structure learning, regardless of whether all the hidden nodes are newly generated or are existent originally. The network parameter vector $\mathbf{W}_{EKF} = [w_1, \mathbf{C}_1^{\mathrm{T}}, \boldsymbol{\Sigma}\_left_1^{\mathrm{T}}, \boldsymbol{\Sigma}\_right_1^{\mathrm{T}}, \cdots, w_u, \mathbf{C}_u^{\mathrm{T}}, \boldsymbol{\Sigma}\_left_u^{\mathrm{T}}, \boldsymbol{\Sigma}\_right_u^{\mathrm{T}}]$ is adapted using the following EKF algorithm,

$$\mathbf{W}_{EKF}(k) \ = \ \mathbf{W}_{EKF}(k-1) + e^k \boldsymbol{\kappa}_k \ , \tag{21}$$

where $\boldsymbol{\kappa}_k$ is the Kalman gain vector given by

$$\boldsymbol{\kappa}_k \ = \ [R_k + \mathbf{a}_k^{\mathrm{T}}\mathbf{P}_{k-1}\mathbf{a}_k]^{-1}\mathbf{P}_{k-1}\mathbf{a}_k \ , \tag{22}$$

and $\mathbf{a}_k$ is the gradient vector and has the following form:

$$\begin{aligned}
&\mathbf{a}_k = \\
&\Big[\varphi_{1k}(\mathbf{X}^k), \varphi_{1k}(\mathbf{X}^k)\frac{2w_1}{\sigma_{11}^2(x_{1k})}(x_{1k} - c_{11}), \cdots, \varphi_{1k}(\mathbf{X}^k)\frac{2w_1}{\sigma_{r1}^2(x_{rk})}(x_{rk} - c_{r1}), \\
&\varphi_{1k}(\mathbf{X}^k)\frac{2sgn(c_{11} - x_{1k})w_1}{(\sigma_{11}^L)^3}(x_{1k} - c_{11})^2, \cdots, \\
&\varphi_{1k}(\mathbf{X}^k)\frac{2sgn(c_{r1} - x_{rk})w_1}{(\sigma_{r1}^L)^3}(x_{rk} - c_{r1})^2, \cdots, \\
&\varphi_{uk}(\mathbf{X}^k), \varphi_{uk}(\mathbf{X}^k)\frac{2w_u}{\sigma_{1u}^2(x_{1k})}(x_{1k} - c_{1u}), \cdots, \varphi_{uk}(\mathbf{X}^k)\frac{2w_u}{\sigma_{ru}^2(x_{rk})}(x_{rk} - c_{ru}), \\
&\varphi_{uk}(\mathbf{X}^k)\frac{2sgn(x_{1k} - c_{1u})w_u}{(\sigma_{1u}^L)^3}(x_{1k} - c_{1u})^2, \cdots, \\
&\varphi_{uk}(\mathbf{X}^k)\frac{2sgn(x_{rk} - c_{ru})w_u}{(\sigma_{ru}^L)^3}(x_{rk} - c_{ru})^2\Big]^{\mathrm{T}} \ ,
\end{aligned} \tag{23}$$

Table 1. Comparisons of the proposed GOSFNN with other algorithms

Algorithms	Rules	$APE_{trn}(\%)$	$APE_{chk}(\%)$
ANFIS	8	0.0043	1.066
OLS	22	2.43	2.56
GDFNN	10	2.11	1.54
SOFNN	9	1.1380	1.1244
FAOS-PFNN	7	1.89	2.95
GOSFNN	7	1.94	2.39

where $sgn(.)$ is the defined sign function given by

$$sgn(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{24}$$

and R_k is the variance of the measurement noise, and $\mathbf{P}_k$ is the error covariance matrix which is updated by

$$\mathbf{P}_k = [\mathbf{I} - \boldsymbol{\kappa}_k \mathbf{a}_k^{\mathrm{T}}]\mathbf{P}_{k-1} + Q_0 \mathbf{I}\,, \tag{25}$$

Here, Q_0 is a scalar which determines the allowed random step in the direction of gradient vector and $\mathbf{I}$ is the identity matrix. When a new hidden neuron is allocated, the dimensionality of the $\mathbf{P}_k$ increases to

$$\mathbf{P}_k = \begin{bmatrix} \mathbf{P}_{k-1} & \mathbf{0} \\ \mathbf{0} & P_0 \mathbf{I} \end{bmatrix}\,, \tag{26}$$

where P_0 is an estimate of the uncertainty in the initial values assigned to the parameters. The dimension of the identify matrix $\mathbf{I}$ is equal to the number of new parameters introduced by the new hidden unit.

4 Simulation Studies

In this section, the effectiveness and superiority of the GOSFNN is demonstrated on a benchmark problem of three-dimensional nonlinear function. Comparisons are made with other significant works such as ANFIS [1], OLS [2], GDFNN [11], SOFNN [12] and FAOS-PFNN [13], *etc.*

The multi-dimensional nonlinear system is given by

$$f(x_1, x_2, x_3) = (1 + x_1^{0.5} + x_2^{-1} + x_3^{-1.5})^2 \tag{27}$$

The training samples consisting of a total of 216 data points are randomly extracted from the input space $[1, 6]^3$ and the corresponding desired outputs can be derived from (27). The parameters used for the training are chosen as follows: $d_{\max} = 0.8$, $d_{\min} = 0.1$, $e_{\max} = 0.8$, $e_{\min} = 0.01$, $\beta = 0.99$, $\gamma = 0.95$, $k_{GF} = 0.99$, $P_0 = R_k = 1.0$ and $Q_0 = 0.11$. To compare the performance with

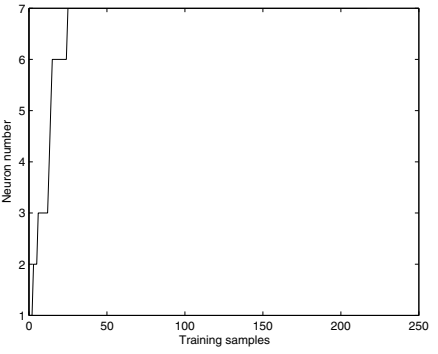


Fig. 2. Growth of neurons

other approaches, the performance index is chosen to be the same as that in [2], which is given by

$$APE = \frac{1}{n}\sum_{k=1}^{n}\frac{|t^k - y^k|}{|t^k|} \tag{28}$$

Another 125 data pairs are randomly selected from the same discourse of universe to check the generalization performance of the resulting fuzzy neural network. Simulation results are shown in Fig.2-4, from which we can see that there are only 7 fuzzy rules in the final OSFNN which can model satisfactorily the underlying multi-dimensional function. Comparisons of the proposed algorithm with ANFIS, OLS, GDFNN, SOFNN and FAOS-PFNN are listed in Table 1, which shows that the proposed GOSFNN obtains a parsimonious structure while the performance of approximation and generalization is considerably better than other algorithms. It should be noted that the ANFIS is a batch learning approach using the BP method rather than the sequential learning although it can obtain a more compact structure. Therefore, the proposed GOSFNN provides the best performance in the sense of high accuracy and generalization with compact structure.

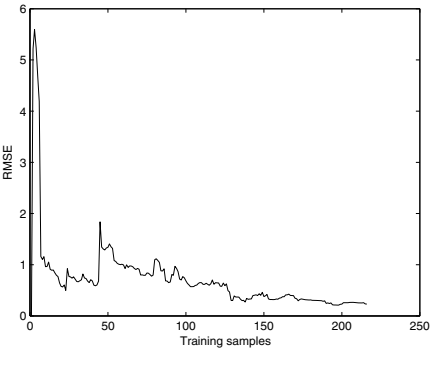


Fig. 3. Root mean squared error (RMSE) during online training

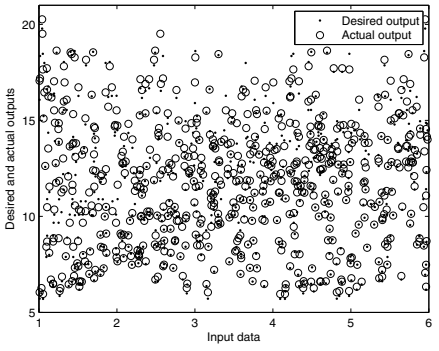


Fig. 4. Comparisons between desired and actual outputs.

5 Conclusions

In this paper, we present a Generalized Online Self-constructing Fuzzy Neural Network (GOSFNN) which implements a TSK fuzzy inference system. The generalized ellipsoidal basis function (GEBF) is introduced by defining a concept of the dissymmetrical Gaussian function (DGF) which releases the symmetry of the standard Gaussian function in each dimension of input variables. Consequently, the proposed GEBF makes the partitioning in the input space more flexible and more efficient, and therefore enhances the performance of the resulting fuzzy neural network. In the online learning process, criteria of rule generation are presented to identify the structure of the GOSFNN by creating GEBFs. The parameter estimation of the resulting GOSFNN is implemented by using the Extended Kalman Filter (EKF) method. The effectiveness and superiority of the proposed GOSFNN is demonstrated in multi-dimensional function approximation. Simulation results show that a compact fuzzy neural network with better generalization performance can be online self-constructed by the GOSFNN. Comprehensive comparisons with other popular approaches indicate that the overall performance of the GOSFNN is superior to the others in terms of parsimonious structure and high capability of approximation and generalization.

References

1. Jang, J.-S.R.: ANFIS: Adaptive-network-based fuzzy inference system. IEEE Trans. on Syst. Man and Cybern. 23, 665–684 (1993)
2. Chen, S., Cowan, C.F.N., Grant, P.M.: Orthogonal least squares learning algorithm for radial basis function network. IEEE Trans. on Neural Netw. 2(2), 302–309 (1991)
3. Chao, C.T., Chen, Y.J., Teng, C.C.: Simplification of fuzzy-neural systems using similarity analysis. IEEE Trans. on Syst. Man and Cybern. Part B Cybern. 26(2), 344–354 (1996)

4. Juang, C.-F., Lin, C.-T.: An on-line self-constructing neural fuzzy inference network and its applications. IEEE Trans. on Fuzzy Syst. 6(1), 12–32 (1998)
5. Platt, J.: A resource-allocating network for function interpolation. Neural Comput. 3, 213–225 (1991)
6. Kadirkamanathan, V., Niranjan, M.: A function estimation approach to sequential learning with neural networks. Neural Comput. 5, 954–975 (1993)
7. Lu, Y.W., Sundararajan, N., Saratchandran, P.: Performance evaluation of a sequential minimal radial basis function (RBF) neural network learning algorithm. IEEE Trans. on Neural Netw. 9(2), 308–318 (1998)
8. Kasabov, N., Song, Q.: DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction. IEEE Trans. Fuzzy Syst. 10, 144–154 (2002)
9. Mitra, S., Hayashi, Y.: Neuro-fuzzy rule generation: survey in soft computing framework. IEEE Trans. Neural Networks 11, 748–768 (2000)
10. Wu, S.-Q., Er, M.J.: Dynamic fuzzy neural networks - a novel approach to function approximation. IEEE Trans. Syst., Man, Cybern., B, Cybern. 30, 358–364 (2000)
11. Wu, S.-Q., Er, M.J., Gao, Y.: A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks. IEEE Trans. Fuzzy Systems 9, 578–594 (2001)
12. Leng, G., McGinnity, T.M., Prasad, G.: An approach for on-line extraction of fuzzy rules using a self-organising fuzzy neural network. Fuzzy Sets and Systems 150, 211–243 (2005)
13. Wang, N., Er, M.J., Meng, X.Y.: A fast and accurate online self-organizing scheme for parsimonious fuzzy neural networks. Neurocomputing 72, 3818–3829 (2009)
14. Wang, N., Er, M.J., Meng, X.Y., et al.: An online self-organizing scheme for parsimonious and accurate fuzzy neural networks. Int. Jour. Neural Systems 20(5), 389–405 (2010)
15. Hsu, C.-F., Lin, P.-Z., Lee, T.-T., Wang, C.-H.: Adaptive asymmetric fuzzy neural network controller design via network structuring adaptation. Fuzzy Sets and Systems 159, 2627–2649 (2008)
16. Velayutham, C.S., Kumar, S.: Asymmetric subsethood-product fuzzy neural inference system (ASuPFuNIS). IEEE Trans. Neural Networks 16, 160–174 (2005)

Adaptive Fuzzy Control of an Active Vibration Isolator

Naibiao Zhou[1], Kefu Liu[1], Xiaoping Liu[1], and Bing Chen[2]

[1] Faculty of Engineering, Lakehead University, Thunder Bay, Canada
[2] Institute of Complexity Science, Qingdao University, Qingdao, China

Abstract. This paper focuses on control of an active vibration isolator that possesses a strong nonlinearity and parameter uncertainty. An adaptive fuzzy controller is developed. A backstepping approach is employed to design the controller. A fuzzy logical system is used to approximate the unknown nonlinear function in the system. The developed controller guarantees the boundedness of all the signals in the closed-loop system. The unique feature of the developed controller is that only one parameter needs to be adapted online. The effectiveness of the controller is demonstrated by a computer simulation.

Keywords: Active vibration isolation, fuzzy control, adaptive control, backstepping control.

1 Introduction

Vibration isolation is very important for many applications. In general there are two types of vibration isolation: isolation of a device from a vibrating source and isolation of a vibrating source from its support. This study concerns the first case which is also referred to as base isolation. The objective of the base isolation is to reduce the displacement transmissibility. In [1], a novel active vibration isolator was developed. The study showed that with a proportional feedback, the closed-loop system has a narrow stability margin due to the time delay caused by the actuator. A phase compensation technique was used to tackle this problem. As the system possesses a strong nonlinearity that is difficult to be modeled accurately, it is desired to develop an adaptive approximation-based controller for it. Various approximation-based adaptive controllers have been proposed in the past. For example, neural network (NN) method has attracted considerable attention because of its inherent capability of approximating nonlinear functions [2-4]. When an NN is used as an approximator, a large number of NN nodes should be used in order to improve the approximation accuracy. As a result, a great number of parameters are required to be adapted online, which increases the learning time. To overcome this shortcoming, a novel adaptive neural control design was presented in [5]. The merit of the proposed controller is that the number of online adapted parameters is independent of the number of the NN nodes. In [6], the fuzzy logical systems were used as approximators for nonlinear time-delay systems. Following the design methodology of [5], a novel adaptive fuzzy control scheme was proposed. An advantage of the proposed design scheme is that the number of adaptive parameters is not more than the order of the system under

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 552–562, 2011.

consideration. In this study, following the design procedure of [6], an adaptive fuzzy controller is developed for the active vibration isolator under consideration.

2 Active Vibration Isolator and Dynamic Model

Figure 1 shows a schematic of the active vibration isolator developed in [1]. A steel beam (1) is used to support a permanent magnet (PM) block (2) that acts as an isolated mass denoted as m. The PM block dimension is $l \times w \times h = 25.4 \times 25.4 \times 29.0$ mm. The mass-beam assembly is placed between a pair of electromagnets (EMs) (3). The tension of the beam can be adjusted by screws (4). Both the beam supports and the EM supports are fastened to a base which rides on two linear guide carts sliding along a precision rail rack (6). The rail rack is fastened to a heavy rigid stand (7). A shaker (5) is used to excite the base through a stinger. Figure 2 illustrates a simplified model for the system and indicates the polarities of the PM and the EMs. Note that the polarities of the EMs vary according to the direction of the coil current. The base motion is denoted by y while the motion of the mass is denoted by x. The stiffness of the beam is represented by k_b while the stiffness of the magnetic spring due to the interaction between the PM and the EM cores is represented by k_m. Both k_b and k_m are inherently nonlinear.

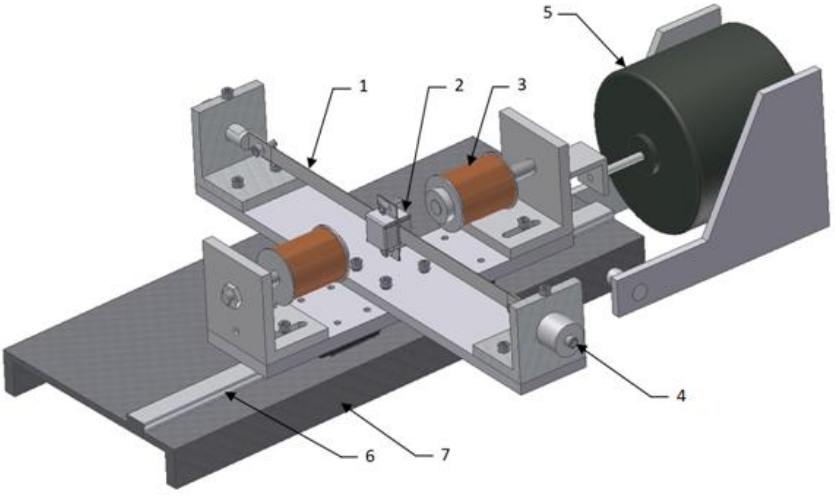


Fig. 1. Active vibration isolator

The equation governing the motion of the mass is given by

$$m\ddot{z} + c\dot{z} + f_b(z) + f_m(z) = -m\ddot{y} + F_c \tag{1}$$

where $z = x - y$ represents the relative displacement of the mass, $f_b(z)$ is the restoring force of the beam and $f_m(z)$ is the attracting force of the magnetic spring,

F_c is the actuating force due to the interaction between the PM and the energized EMs. As shown in Fig. 1(b), the actuator is formed by connecting two identical EMs in parallel. Therefore, the dynamics of the actuator is defined by

$$0.5L\dot{i} + 0.5Ri + k\dot{z} = e \tag{2}$$

where L and R are the inductance and resistance of one EM, e is the voltage that is applied to the EMs, the term $k\dot{z}$ represents the effect of back electromotive force (emf) induced by the motion of the mass with k as a proportional constant, and i is the total current applied to the actuator. The actuating force F_c can be approximated by

$$F_c = \gamma i \tag{3}$$

where γ is a proportional constant.

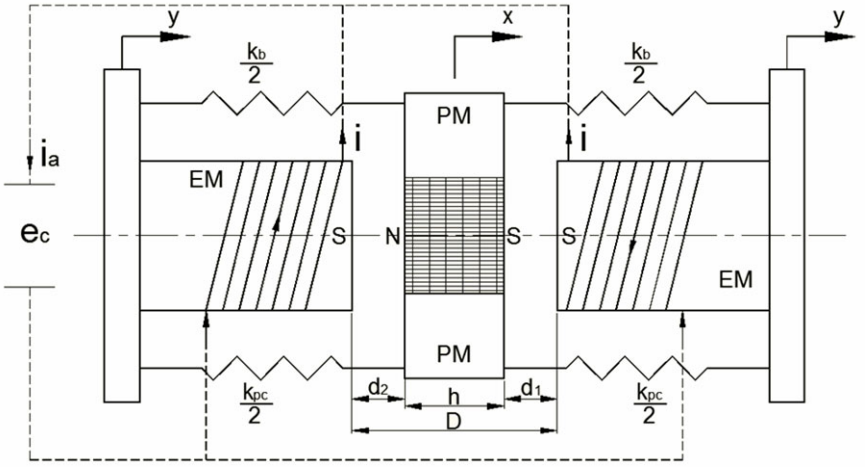


Fig. 2. Schematic of a simplified model for the active isolator

3 Adaptive Fuzzy Controller

By defining the system states as

$$x_1 = z, x_2 = \dot{z}, x_3 = i \tag{4}$$

the system of (1) and (2) can be written in a state space form:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= f(x_1, x_2) + A_{23}x_3 + d(t) \\ \dot{x}_3 &= -A_{32}x_2 - A_{33}x_3 + ge \end{aligned} \tag{5}$$

where $f(x_1, x_2) = -f_b(x_1)/m - f_m(x_1)/m - cx_2/m$, $A_{23} = \gamma/m$, $d(t) = -\ddot{y}(t)$ $A_{32} = 2k/L$, $A_{33} = R/L$, $g = 2/L$. The following assumptions are made

1. $f(x_1, x_2)$ is an unknown nonlinear smooth function with $f(0,0) = 0$. The parameters A_{23}, A_{32}, A_{33} and g are known positive constants.
2. $d(t)$ is an unknown external disturbance which satisfies $|d(t)| \le \bar{d}$ with $\bar{d}$ being a constant.

A fuzzy logic system consists of four parts: the knowledge base, the fuzzifier, the fuzzify inference engine, and the defuzzifier. With the singleton fuzzifier, product inference and center-average defuzzifier, the fuzzy logic system output can be expressed as

$$f(\chi) = W^T \Phi(\chi) \tag{6}$$

where $\chi = [\chi_1 \quad \chi_2 \quad \cdots \quad \chi_n]^T$ and $W^T = [w_1 \quad w_2 \quad \cdots \quad w_N]$ with w_i being the center of the output membership function for the i^{th} rule and $\Phi(\chi) = [\phi_1(\chi) \quad \phi_2(\chi) \quad \cdots \quad \phi_N(\chi)]^T$ with $\phi_i(\chi)$ defined as

$$\phi_i(\chi) = \prod_j^n \mu_{F_j^i}(\chi_j) \Big/ \sum_{i=1}^N \left[\prod_j^n \mu_{F_j^i}(\chi_j) \right] \tag{7}$$

where $\mu_{F_j^i}(\chi_j)$ is the membership function of F_j^i. It has been proven in [7] that the above fuzzy logic system is capable of uniformly approximating any continuous nonlinear function over a compact set Ω_χ with any degree of accuracy. Eq. (7) can also be considered to be the output of a three-layer neural network.

The main goal of this study is to design a controller for the system (5) so as to quickly attenuate the system response x_1, while all the signals in the closed-loop system remain bounded. Following the backstepping technique, the auxiliary variables are introduced

$$z_1 = x_1, \; z_2 = x_2 - \alpha_1, \; z_3 = x_3 - \alpha_2 \tag{8}$$

where α_1 and α_2 are virtual control variables. Through a Lyapunov-based design procedure given in Appendix, the virtual control signals and the real control signal are defined by

$$\alpha_1 = -k_1 z_1 \tag{9}$$

$$\alpha_2 = -\frac{g}{A_{23}}(k_2 + \frac{1}{2}) z_2 - \frac{g}{2A_{23}a^2} \hat{\theta} \Phi^T(Z) \Phi(Z) z_2 \tag{10}$$

$$e = -A_{32} x_2 / g - A_{33} x_3 / g - A_{23} z_2 / g - k_3 z_3 + \dot{\alpha}_2 / g \tag{11}$$

and the adaptive law is given by

$$\dot{\hat{\theta}} = \frac{r}{2a^2} z_2^2 \Phi^T(Z)\Phi(Z) - \sigma\hat{\theta} \tag{12}$$

where $Z = [z_1 \quad z_2]^T$ and k_1, k_2, k_3, a, r, and σ are positive design parameters, $\hat{\theta}$ is the estimate of a constant parameter defined as

$$\theta = \|W\|^2 / g \tag{13}$$

Theorem. The responses of the system (5) controlled by the control laws given in Eqs. (11) to (12) are globally uniformly ultimately bounded. The proof of the theorem is given in Appendix.

4 Computer Simulation

To examine the effectiveness of the developed controller, a computer simulation is conducted. Based on the identification result of [8], the nonlinear restoring force is approximated by a 5th-order polynomial

$$f_s = 754.0x_1 + 3.5\times10^8 x_1^3 + 3.5\times10^7 x_1^5 \tag{14}$$

the system parameters are taken as $m = 0.17$ kg, $c = 0.2831$ Ns/m, $\gamma = 2.46$ N/A, $R = 9.2$ Ohm, $L = 0.16$ H, $k = 0.525$ As/m.

For the fuzzy logic system, Gaussian function is used as the membership function or base function. Thus, for the i[th] rule and the z_j input universe of discourse, the membership function is defined as

$$\mu_{F_j^i}(z_j) = \exp\left(-0.5\left((z_j - c_j^i)/\sigma_j^i\right)^2\right) \tag{15}$$

where c_j^i and σ_j^i are the center and spread of the membership function, respectively. Thus for the i[th] element of $\Phi(Z)$ is defined by

$$\phi_i(Z) = \frac{\exp\left(-0.5\left((z_1 - c_1^i)/\sigma_1^i\right)^2 - 0.5\left((z_2 - c_2^i)/\sigma_2^i\right)^2\right)}{\sum_{i=1}^{N}\left[\exp\left(-0.5\left((z_1 - c_1^i)/\sigma_1^i\right)^2 - 0.5\left((z_2 - c_2^i)/\sigma_2^i\right)^2\right)\right]} \tag{16}$$

To evaluate the performance of the controller, two indices are used [9]. The following index measures the control performance

$$P = \left\|z_1^{con}\right\|_{rms} / \left\|z_1^{unc}\right\|_{rms} \tag{17}$$

where $\left\|z_1^{con}\right\|_{rms}$ and $\left\|z_1^{unc}\right\|_{rms}$ denote the root-mean-squared (rms) value for the system response with control and without control, respectively. The following index measures the strength of the control action

$$S = \left\|F_c\right\|_{rms} / \left\|m\ddot{y}\right\|_{rms} \tag{18}$$

where $\left\|F_c\right\|_{rms}$ denotes the rms value for the control force F_c and $\left\|m\ddot{y}\right\|_{rms}$ the rms value for the inertial force due to the base disturbance. Taft's earthquake record [9] is used as a testing disturbance. The initial conditions are chosen to be zero. The number of fuzzy rules is chosen to be $N = 7$. The centers and spreads of the membership functions are chosen according to possible ranges Δz_j of the auxiliary variables.

$$c_j^i = -\Delta z_j / 2 + (i-1)\Delta z_j / 6 \text{ and } \sigma_j^i = \sigma_j = 0.15\Delta z_j,\ i = 1, 2, \cdots, 7$$

respectively, In the simulation, the auxiliary variable ranges are chosen to be $\Delta z_1 = 0.005$ and $\Delta z_2 = 0.05$. The initial value for θ is chosen to be zero.

Table 1. Simulation results

case	k_1	k_2	k_3	a	r	σ	P	S
1	25	10	5	0.1	1000	1	0.2523	0.5755
2	50	10	5	0.1	1000	1	0.1565	0.5968
3	50	10	5	0.1	0	0	0.5218	0.4552
4	50	10	5	0.1	1500	1	0.0983	0.6119
5	50	0	5	0.1	1000	1	0.2642	0.5720
6	50	10	5	0.5	1000	1	0.6935	0.3213
8	50	10	5	0.1	1000	10	0.2803	0.5579

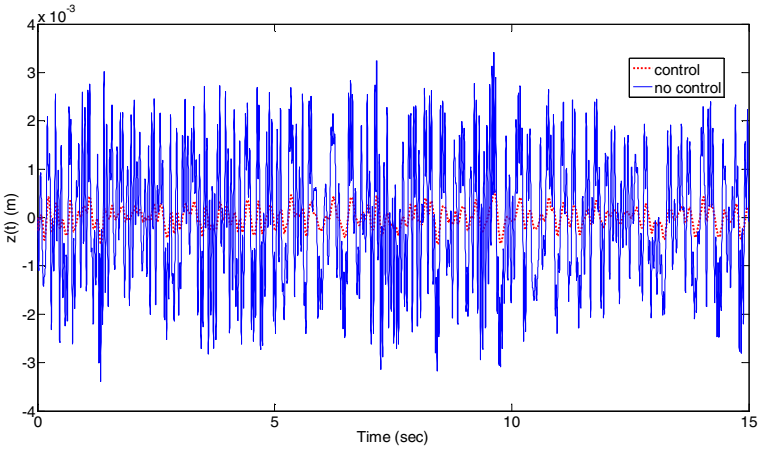


Fig. 3. Comparison of the open-loop response and the closed-loop response control by case 2

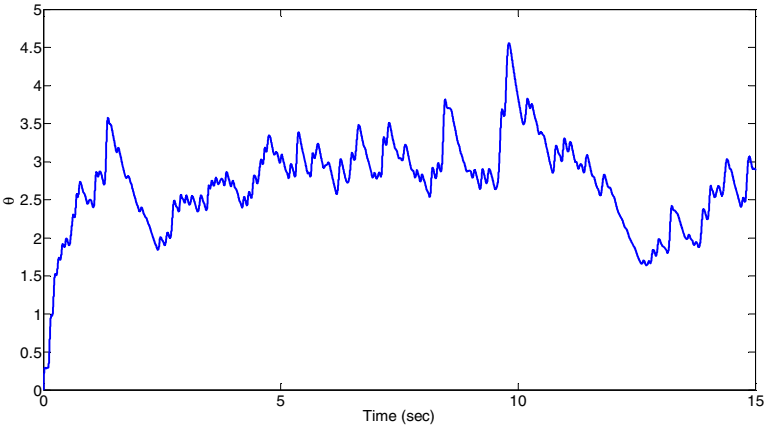


Fig. 4. The adaptive parameter $\hat{\theta}$ for control case 2

Table 1 lists some of the simulation results. Figs. 3 and 4 show the results with control case 2. From them, the following observations can be drawn. Among the cases considered, case 4 achieves the best vibration suppression at the price of the largest control effort used. Shown by case 3, without the adaptive control part, the performance deteriorates. A comparison of case 1 and case 2 indicates that an increase of the gain k_1 will improve the control performance while increasing the control effort. Shown by case 2 and case 6, the smaller the parameter a, the smaller the index P. Shown by cases 2, 4, and 7, increasing the parameter r will reduce the index P and increase the index S. The parameter σ affects the transient behavior of the adaptive parameter. An increase of the σ value will negatively influence the performance as shown by cases 2 and 8.

5 Conclusions

In the paper, an adaptive fuzzy controller is developed for an active vibration isolator. A fuzzy logical system is used to approximate the unknown nonlinear function. The unique feature of the developed controller is that only one parameter needs to be adapted online. A computer simulation has shown the effectiveness of the controller and the guidelines for the parameter tuning.

References

[1] Coppola, G., Liu, K.: Control of a unique active vibration isolator with a phase compensation technique and automatic on/off switching. J. of Sound and Vibration 329, 5233–5248 (2010)

[2] Chen, L., Narendra, K.: Nonlinear adaptive control using neural networks and multiple models. Automatica 37, 1245–1255 (2001)

[3] Tan, K., Huang, S., Lee, T.: Adaptive backstepping control for a class of nonlinear systems using neural network approximation. Int. J. of Robust and Nonlinear Control 14, 643–664 (2004)
[4] Gao, W., Selmic, R.: Neural network control of a class of nonlinear systems with actuator saturation. IEEE Trans. on Neural Networks 17, 147–156 (2006)
[5] Chen, B., Liu, X., Liu, K., Lin, C.: Novel adaptive neural control design for nonlinear MIMO time-delay systems. Automatica 45, 1554–1560 (2009)
[6] Wang, M., Chen, B., Liu, K., Liu, X., Zhang, S.: Adaptive fuzzy tracking control of nonlinear time-delay systems with unknown virtual control coefficients. Information Science 178, 4326–4340 (2008)
[7] Lee, H., Tomizuka, M.: Robust adaptive control using a universal approximator for SISO nonlinear systems. IEEE Trans. Fuzzy Sets 8, 95–106 (2000)
[8] Zhou, N., Liu, K.: A tunable high-static-low-dynamic stiffness vibration isolator. J. of Sound and Vibration 329, 1254–1273 (2010)
[9] Manosa, V., Ikhouane, F., Rodellar, J.: Control of uncertain non-linear systems via adaptive backstepping. J. of Sound and Vibration 280, 657–680 (2005)

Appendix A: Proof of the Theorem

Lemma 1. For a given $\varepsilon > 0$, any continuous function $f(\chi)$ defined on a compact set $\Omega_\chi \subset R^n$ can be written as:

$$f(\chi) = W^T \Phi(\chi) + \delta(\chi),\ |\delta(\chi)| \le \varepsilon \tag{A1}$$

where $\delta(\chi)$ is the approximation error. The term $W^T \Phi(\chi)$ may be viewed as an approximator with $\Phi(\chi)$ being a chosen base function vector and W being the weight vector.

Lemma 2. Let A and B be any row vector and column vector, respectively, with appropriate dimension then

$$AB \le \frac{1}{2\rho^2}\|A\|^2 + \frac{\rho^2}{2}\|B\|^2 \tag{A2}$$

where ρ is any nonzero real number.

Lemma 3. Let $\hat{\theta}$ denotes the estimate of a constant θ. Define the estimation error as $\tilde{\theta} = \theta - \hat{\theta}$ then

$$\dot{\tilde{\theta}} = -\dot{\hat{\theta}} \text{ and } \tilde{\theta}\hat{\theta} \le \frac{1}{2}\theta^2 - \frac{1}{2}\tilde{\theta}^2 \tag{A3}$$

Step 1. Consider a Lyapunov's function candidate as:

$$V_1 = \frac{1}{2} z_1^2 \tag{A4}$$

Differentiating V_1 yields:

$$\dot{V}_1 = z_1\dot{z}_1 = z_1(x_2 - \alpha_1 + \alpha_1) = z_1(z_2 - k_1 z_1) = -k_1 z_1^2 + z_1 z_2 \tag{A5}$$

Step 2. Choose a Lyapunov's function candidate as:

$$V_2 = V_1 + \frac{1}{2}z_2^2 + \frac{b}{2r}\tilde{\theta}^2 \tag{A6}$$

where $\tilde{\theta} = \theta - \hat{\theta}$ is parameter estimation error. Differentiating V_2 yields:

$$\dot{V}_2 = \dot{V}_1 + z_2\dot{z}_2 + \frac{b}{r}\tilde{\theta}\dot{\tilde{\theta}} = \dot{V}_1 + z_2(\dot{x}_2 - \dot{\alpha}_1) - \frac{b}{r}\tilde{\theta}\dot{\hat{\theta}} \tag{A7}$$

as $\dot{\tilde{\theta}} = -\dot{\hat{\theta}}$. From Eqs. (A5) and (5),

$$\begin{aligned}\dot{V}_2 &= -k_1 z_1^2 + z_1 z_2 + z_2(f(x_1, x_2) + A_{23}x_3 + d(t) - \dot{\alpha}_1) - \frac{b}{r}\tilde{\theta}\dot{\hat{\theta}} \\ &= -k_1 z_1^2 + z_2 d(t) + z_2(f(x_1, x_2) - \dot{\alpha}_1 + z_1) + A_{23}z_2 z_3 + A_{23}z_2\alpha_2 - \frac{b}{r}\tilde{\theta}\dot{\hat{\theta}}\end{aligned} \tag{A8}$$

Using Lemma 2 and Assumption 2,

$$z_2 d(t) \le \frac{1}{2\rho^2}z_2^2 + \frac{\rho^2}{2}d^2(t) \le \frac{1}{2\rho^2}z_2^2 + \frac{\rho^2}{2}\bar{d} \tag{A9}$$

Therefore

$$\begin{aligned}\dot{V}_2 &\le -k_1 z_1^2 + \frac{1}{2\rho^2}z_2^2 + \frac{\rho^2}{2}\bar{d} + z_2(f(x_1, x_2) - \dot{\alpha}_1 + z_1) + A_{23}z_2 z_3 + A_{23}z_2\alpha_2 - \frac{b}{r}\tilde{\theta}\dot{\hat{\theta}} \\ &= -k_1 z_1^2 + \frac{\rho^2}{2}\bar{d} + z_2(f(x_1, x_2) - \dot{\alpha}_1 + z_1 + \frac{1}{2\rho^2}z_2) + A_{23}z_2 z_3 + A_{23}z_2\alpha_2 - \frac{b}{r}\tilde{\theta}\dot{\hat{\theta}}\end{aligned}$$

Define $\bar{f}(Z) = f(x_1, x_2) - \dot{\alpha}_1 + z_1 + \frac{1}{2\rho^2}z_2$ where $Z = [z_1 \quad z_2]^T \subset R^2$, therefore,

$$\dot{V}_2 \le -k_1 z_1^2 + \frac{\rho^2}{2}\bar{d} + z_2\bar{f}(Z) + A_{23}z_2 z_3 + A_{23}z_2\alpha_2 - \frac{b}{r}\tilde{\theta}\dot{\hat{\theta}} \tag{A10}$$

By Lemma 1,

$$z_2\bar{f}(Z) = z_2 W^T\Phi(Z) + z_2\delta(Z) = \frac{W^T}{\|W\|}z_2\|W\|\Phi(Z) + z_2\delta(Z) \tag{A11}$$

Using Lemma 2, then

$$\left(\frac{W^T}{\|W\|}\right)\left(z_2\|W\|\Phi(Z)\right) \le \frac{1}{2}a^2 + \frac{1}{2a^2}z_2^2\|W\|^2\Phi^T(Z)\Phi(Z) \tag{A12}$$

$$z_2\delta(Z) \le \frac{1}{2}gz_2^2 + \frac{1}{2g}\delta^2(Z) \le \frac{1}{2}gz_2^2 + \frac{1}{2g}\varepsilon^2 \tag{A13}$$

Substitution of (A12) and (A13) into (A11) yields:

$$\begin{aligned} &z_2\bar{f}(Z) \le \frac{1}{2}a^2 + \frac{g}{2a^2}z_2^2\frac{\|W\|^2}{g}\Phi^T(Z)\Phi(Z) + \frac{1}{2}gz_2^2 + \frac{1}{2g}\varepsilon^2 \le \\ &\frac{g}{2a^2}z_2^2\theta\Phi^T(Z)\Phi(Z) + \frac{1}{2}a^2 + \frac{1}{2}gz_2^2 + \frac{1}{2g}\varepsilon^2 \end{aligned} \tag{A14}$$

By Eq. (10), $z_2A_{23}\alpha_2$ can be written as:

$$\begin{aligned} &z_2A_{23}\alpha_2 = z_2A_{23}\left(-\frac{g}{A_{23}}(k_2+\frac{1}{2})z_2 - \frac{g}{2A_{23}a^2}\hat{\theta}z_2\Phi^T(Z)\Phi(Z)\right) = \\ &-(k_2+\frac{1}{2})bz_2 - \frac{g}{2a^2}\hat{\theta}z_2^2\Phi^T(Z)\Phi(Z) \end{aligned} \tag{A15}$$

Substitution of (A14) and (A15) into (A10) yields

$$\dot{V}_2 \le -k_1z_1^2 - k_2gz_2^2 + A_{23}z_2z_3 + \frac{1}{2}\left(a^2 + \frac{\varepsilon^2}{g} + \rho^2\bar{d}\right) + \frac{g}{r}\left(\frac{r}{2a^2}z_2^2\Phi^T(Z)\Phi(Z) - \dot{\hat{\theta}}\right)$$

Substitution of Eq. (12) for $\dot{\hat{\theta}}$ yields:

$$\dot{V}_2 \le -k_1z_1^2 - k_2gz_2^2 + A_{23}z_2z_3 + \frac{1}{2}\left(a^2 + \frac{\varepsilon^2}{g} + \rho^2\bar{d}\right) + \frac{g\sigma}{r}\tilde{\theta}\hat{\theta} \tag{A16}$$

By Lemma 3,

$$\begin{aligned} &\dot{V}_2 \le -k_1z_1^2 - k_2gz_2^2 + A_{23}z_2z_3 + \frac{1}{2}\left(a^2 + \frac{\varepsilon^2}{g} + \rho^2\bar{d}\right) + \frac{g\sigma}{r}\left(\frac{1}{2}\theta^2 - \frac{1}{2}\tilde{\theta}^2\right) \\ &= -k_1z_1^2 - k_2gz_2^2 - \frac{g\sigma}{2r}\tilde{\theta}^2 + A_{23}z_2z_3 + \frac{1}{2}\left(a^2 + \frac{\varepsilon^2}{g} + \rho^2\bar{d} + \frac{g\sigma}{r}\theta^2\right) \end{aligned} \tag{A17}$$

Step 3. Choose a Lyapunov's function candidate as:

$$V_3 = V_2 + \frac{1}{2}z_3^2 \tag{A18}$$

Since this Lyapunov function contains all of the signals and parameter estimate error, it can be viewed as the total Lyapunov function for the entire system, namely

$$V = \frac{1}{2}z_1^2 + \frac{1}{2}z_2^2 + \frac{1}{2}z_3^2 + \frac{g}{2r}\tilde{\theta}^2 \tag{A19}$$

Its time derivative is given by

$$\dot{V} = \dot{V}_2 + z_3\dot{z}_3 = \dot{V}_2 + z_3\left(-A_{32}x_2 - A_{33}x_3 + ge - \dot{\alpha}_2\right) \tag{A22}$$

Substituting Eq. (11) and Eqs. (A9) and (A11) yields

$$\dot{V} \le -k_1 z_1^2 - k_2 g z_2^2 - k_3 g z_3^2 - \frac{g\sigma}{2r}\tilde{\theta}^2 + \frac{1}{2}\left(a^2 + \frac{\varepsilon^2}{g} + \rho^2\bar{d} + \frac{g\sigma}{r}\theta^2\right) = -pV + q \tag{A20}$$

where $p = \min\left(2k_1, 2k_2 g, 2k_3 g, \sigma\right)$ and $q = \frac{1}{2}\left(a^2 + \frac{\varepsilon^2}{g} + \rho^2\bar{d} + \frac{g\sigma}{r}\theta^2\right)$.

Eq. (20) implies that

$$\text{for } t \ge 0,\ V \le V(0)e^{-pt} + \frac{q}{p},\ \text{thus } \lim_{t\to\infty} V(t) \le \frac{q}{p} \tag{A21}$$

From (A21), V is shown to be uniformly bounded, which implies that z_1, z_2, z_3, and $\tilde{\theta}$ are bounded. Thus, the state variables x_1, x_2, x_3, and the estimated parameter $\hat{\theta}$ are also bounded. As a consequence the boundedness of the control e is obtained.

Fuzzy-Adaptive Fault-Tolerant Control of High Speed Train Considering Traction/Braking Faults and Nonlinear Resistive Forces

M.R. Wang[1], Y.D. Song[1], Q. Song[1], and Peng Han[2]

[1] Center for System Intelligence and Renewable Energy,
Beijing Jiaotong University, Beijing, 100044, China
ydsong@bjtu.edu.cn
[2] Chongqing Academy of Science and Technology
Chongqing, China

Abstract. High precision speed and position tracking control is important to ensure safe and reliable operation of high speed train. This paper presents a solution to achieve fault-tolerant control of train consisting of multiple vehicles with distributed traction and braking systems. A multiple point-mass model coupled with uncertain resistive forces (i.e. aerodynamic resistance, mechanical resistance, transient impacts, *etc.*) is utilized for control design and stability analysis. Traction and braking faults in the form of tracking power loss and/or braking capability loss are explicitly considered. To cope with the resultant dynamic model that contains actuator faults, uncertain in-train forces as well as resistive disturbances, a fuzzy-adaptive fault-tolerant control method is proposed. The salient feature of the developed control scheme lies in its independence of the precise dynamic model of the train. More specifically, there is no need for system parameter estimation, no need for fault detection and diagnosis, and no need for in-train force and resistive force estimation or approximation in designing and implementing the control scheme. The stable control algorithm is derived based on Lyapunov stability theory. Its effectiveness is confirmed with simulation verification.

Keywords: Fuzzy-adaptive fault-tolerated control, multiple point mass model, Lyapunov stability theory, in-train force.

1 Introduction

As an efficient massive transportation system, high speed train is experiencing rapid development worldwide during the past few years [1]-[4]. As the travel speed increases, safe and reliable operation naturally becomes an extremely important factor to consider in developing ATP/ATO (automatic train protection/automatic train operation) systems, where advanced control represents one of the crucial enabling technologies.

This paper is concerned with speed and position tracking control of high speed train. A multiple point-mass model is used to describe the dynamic behavior of the

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 563–573, 2011.

train where resistive forces and in-train forces between adjacent vehicles are explicitly considered. It should be mentioned that the in-train forces and resistive forces are nonlinear and uncertain [6], measuring or modeling such forces are extremely difficult in practice. The most typical way to around this is to linearize or approximate the nonlinear and uncertain impacts based on a prior designed speed [7],[14]. However, as the in-train forces and aerodynamic drag forces are proportional to the square of the travel speed, its influence on train's dynamic behavior becomes more significant when the speed increases to a higher level. Furthermore, the distributed traction/braking units of the train might experience faults during the system operation. Therefore it is of theoretical and practical importance to develop effective control schemes for speed and position tacking of the train that take into account uncertain resistive forces and in-train forces as well as actuation faults.

We present a fuzzy-adaptive fault-tolerant control scheme to account for all the above mentioned issues. A multiple point-mass model coupled with uncertain resistive forces (i.e. aerodynamic resistance, mechanical resistance, transient impacts, etc.) is utilized for control design and stability analysis. Traction and braking faults in the form of tracking power loss and/or braking capability loss are explicitly considered. The developed control scheme is essentially model-independent in that it does not need the precise dynamic model of the train. More specifically, the design and implementation of the control scheme do not involve system parameter estimation, fault detection and diagnosis, nor in-train force and resistive force estimation or approximation.

The rest of this paper is organized as follows. Section II describes multiple point-mass model of the train consisting of multiple vehicles. Section III gives a brief introduction of the fuzzy system, based on which the fuzzy adaptive and fault-tolerant control algorithm is developed. Section IV presents the simulation results to demonstrate the performance the proposed method. The paper is closed in Section V.

2 Modeling and Problem Statement

Consider a train consisting of n vehicles, as illustrated in Figure 1. The adjacent vehicles are connected by couplers [7], y_i (i=1, 2,..,n) describes the position of each vehicle, the various forces acting on each vehicle includes: pulling/braking force f_i, resistance force f_{di} ,interaction force between two adjacent vehicles f_{bi} and f_{bi-1} ,and other disturbing forces d_i. By Newton's law, the equations of motion for each vehicle can be established as follows,

$$m_i \ddot{y}_i = \lambda_i f_i - f_{di} + f_{bi-1} - f_{bi} + d_i \quad . \tag{1}$$

where m_i is the mass of each vehicle, and the resistive force $f_{di} = w_{1i} + w_{2i}$ consists of two parts: the basic resistance w_{1i} and additional resistance w_{2i}, which are of the form:

$$w_{1i} = a(i,t) + b(i,t)\dot{y}_i + c(i,t)\dot{y}_i^{\,2} \quad . \tag{2}$$

and

$$w_{2i} = f_{ri} + f_{ci} + f_{ti} \quad . \tag{3}$$

respectively. The resistive force for the i^{th} vehicle w_{1i} is caused by the mechanical friction and aerodynamic drag (the coefficients a, b and c are time-varying and unavailable precisely in general), f_{ri} is the ramp resistance related to the component of the gravity in the direction of the slope, f_{ci} denotes the curvature resistance, f_{ti} models the tunnel resistance (caused by the air in the tunnel), λ_i is the distributing parameter determining the power/braking effort of the i^{th} vehicle, f_{bi} represents the in-train force from the couplers connecting the adjacent vehicles. It is noted that modeling or measuring such force is extremely difficult in practice. The commonly used approach is to linearize f_{bi} or approximate its impact using [7],[14],[17]:

$$f_{bi} = b_i \Delta\dot{d}_i \left|\Delta\dot{d}_i\right| + (k_{0i} + k_{1i}\Delta\dot{d}_i^{\,2})\Delta\dot{d}_i \quad . \tag{4}$$

However, determining the coefficients b_i, k_{0i} and k_{1i} is a non-trivial task. In this work, we deal with this challenge, together with other anomaly impacts, by integrating fuzzy technique with adaptive control.

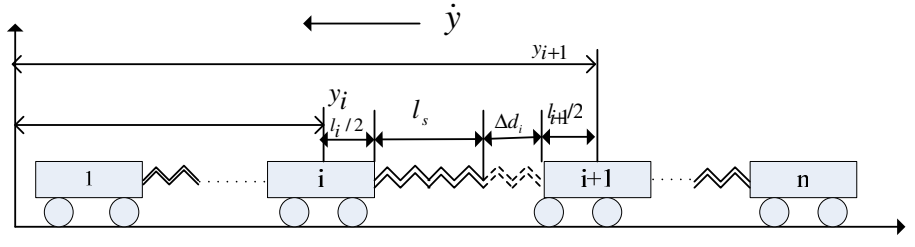


Fig. 1. Relative position and connection relationship among the vehicles.

Considering all the vehicles connected in the train, the following multiple vehicle dynamic model can be derived,

$$M\ddot{Y} = \Lambda F - F_d + F_b - F_b^{'} + D \quad . \tag{5}$$

where

$$M = \begin{bmatrix} m_1 & & & & \\ & m_2 & & & \\ & & . & & \\ & & & . & \\ & & & & m_n \end{bmatrix}, \quad \ddot{Y} = \begin{bmatrix} \ddot{y}_1 \\ \ddot{y}_2 \\ . \\ \vdots \\ \ddot{y}_n \end{bmatrix}, \quad F = \begin{bmatrix} f_1 \\ f_2 \\ . \\ \vdots \\ f_n \end{bmatrix}, \quad D = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_n \end{bmatrix}.$$

$$F_d = \begin{bmatrix} f_{d1} \\ f_{d2} \\ . \\ . \\ f_{dn} \end{bmatrix}, \quad F_b = \begin{bmatrix} f_{b0} \\ f_{b1} \\ . \\ . \\ f_{bn-1} \end{bmatrix}, \quad F'_b = \begin{bmatrix} f_{b1} \\ f_{b2} \\ . \\ . \\ f_{bn} \end{bmatrix}, \quad \Lambda = \begin{bmatrix} \lambda_1 & & & & \\ & \lambda_2 & & & \\ & & . & & \\ & & & . & \\ & & & & \lambda_n \end{bmatrix}$$

Notice that during system operation unexpected faults might occur, which are actually difficult to foresee and prevent. To account for this fact, we consider the situation that the powering/braking unit suffers from the fault of loss of effectiveness such that the dynamic behavior of the multiple vehicle train is governed by

$$M\ddot{Y} = \Lambda[\mu(t)F + P(t)] - F_d + F_b - F_b' + D \ . \tag{6}$$

$$\mu(t) = diag\{\mu_1(t), \mu_2(t), ...\mu_n(t)\} \ . \tag{7}$$

where $P(t)$ denotes a vector function corresponding to the portion of the control action produced by the actuator that is completely out of control, which might be time-varying, $\mu_i(t)$ is a time-varying scalar function reflecting the effectiveness of the powering/braking force F_i of each vehicle, called actuator efficiency or actuator "health indicator", which bears the following physical meaning:

- If $\mu_i(t) = 1$, the powering/braking system of the i^{th} vehicle is healthy.
- If $\mu_i(t) = 0$, the i^{th} vehicle totally loses its traction or braking capability.
- If $0 < \mu_i(t) < 1$, the i^{th} vehicle partially loses its traction or braking effectiveness.

In this work, we consider the case that $\mu_i(t) \in (0,1]$, namely, the actuators have faults but all are still functional. Also, it is worth noting that the mass of each vehicle m_i (therefore the total mass of the train M) is unavailable since the loads or the number of passengers on board each vehicle are different and uncertain in general.

The objective is to design the control input F to make the position and speed tracking errors of the train ($E = Y - Y^*$ and $\dot{E} = \dot{Y} - \dot{Y}^*$, where Y^* and $\dot{Y}^*$ are the desired position and speed, respectively) sufficiently small asymptotically (practical tracking) in the presence of possible actuation faults and uncertain system parameters as well as external disturbances.

To facilitate the control design, we define a filtered variable:

$$S = \dot{E} + BE \ . \tag{8}$$

where $B = diag(\beta_1,...,\beta_n)$ with $\beta_i > 0$ being a free design parameter chosen by the designer/user. Based on which (6) can be expressed as:

$$M\dot{S} = \Lambda\mu(.)F - [M(\ddot{Y}^* - B\dot{E}) + F_d - (T-I)F_{in} + D - \Lambda P(t)] . \tag{9}$$

where I is a unit matrix, T is an elementary matrix and F_{in} is the in-train force vector. Therefore, the problem of speed and position tracking can be addressed by stabilizing S , as detailed in next section.

3 Fuzzy Adaptive Fault-Tolerant Control Design

As the system involves nonlinearities and uncertainties arisen from resistive forces, in-train forces as well as actuator failures, model based control is no longer applicable. Here fuzzy technique is integrated with adaptive control to deal with the lumped uncertainties of the system. As the first step, we denote the lumped uncertainties by

$$R(.) = -M(\ddot{Y}^* - B\dot{E}) - F_d + (T-I)F_{in} - D + \Lambda P(t) \tag{10}$$

The next task is to replace $R(.)$ by specific formula of fuzzy systems and develop an adaptation law for adjusting the parameters in the fuzzy systems for the purpose of forcing the tracking error to converge to a small neighborhood of the origin.

3.1 Takagi-Sugeno Fuzzy Systems

A multiple-input single-output (MISO) fuzzy system is a nonlinear mapping from an input vector $X = [x_1, x_2.....,x_n]^T \in \Re^n$ to an output vector $y \in \Re$ [9]-[11]. For a zero-order Takagi-Sugeno fuzzy systems, the fuzzy rule base contains a collection of fuzzy IF-THEN rules of the form[12],[15]:

$$R^L : IF\,(x_1\ is\ F_1^L)\ and...and\ (x_n\ is\ F_n^L)$$
$$THEN: y = y^L$$

where F_i^L is the label of the fuzzy set corresponding to the variable x_i , for L=1, 2…N, and y^L is a constant parameter representing the consequent part of the fuzzy rule L.

$$y = \frac{\sum_{L=1}^{N} y^L (\prod_{i=1}^{n} \mu_{F_i^L}(x_i))}{\sum_{L=1}^{N} (\prod_{i=1}^{n} \mu_{F_i^L}(x_i))} \tag{11}$$

If we fix the $\mu^L_{F_i}$'s and view the y^L 's as adjustable parameters, then (11) can be written as $y = w^T \psi(x)$ (12)

where $w = (w_1, w_2 ... w_N)^T$ is the parameter vector, and $\psi(x) = (\psi_1(x), \psi_2(x) ... \psi_N(x))^T$ is the vector of fuzzy basis functions defined as:

$$\psi_j(x) \triangleq \frac{\prod_{i=1}^{n} \mu_{F_i^L}(x_i)}{\sum_{L=1}^{N} (\prod_{i=1}^{n} \mu_{F_i^L}(x_i))}, \quad j = (1, 2 ... N) \tag{13}$$

As shown in [8] that Gaussian basis functions do have the best approximation property. One of the choices for the Gaussian basis functions to characterize the membership functions is of the form:

$$\mu_{F^L_i}(x_i) = \exp(-\frac{1}{2}(\frac{x_i - c_i^L}{\sigma_i})^2) \tag{14}$$

The centers c_i are usually randomly selected over a grid of possible values for the vector x_i , and the parameter σ_i usually can be chosen as a constant for all the membership functions.

3.2 Fuzzy Approximator

Now the fuzzy system as defined in (12) is used to approximate the nonlinear function *R*(.),

$$R(.) = w^T \psi + \xi \tag{15}$$

where $\psi \in R^L$ is fuzzy basis function vector, ξ is the fuzzy approximation error and $w \in R^{L \times n}$ is the optimal (ideal) weight matrix, *L* is the total number of fuzzy membership function. According to the universal approximation theory, the fuzzy reconstruction error ξ can be made bounded by a suitably chosen fuzzy system, i.e., $\|\xi\| \le \rho < \infty$, where ρ is some unknown constant. In our study, the following membership functions are used,

$$\mu_{F^k_p}(e_p) = \exp(-(e_p - c_k)^2), \; k = 1, 2 ... L, \; p = 1, 2 \tag{16}$$

where e_1 stands for the position tracking error and e_2 stands for the velocity tracking error. The number of the Gaussian functions is *L*, and the basis function ψ_i can be expressed as:

$$\psi_{ik} = \frac{\exp(-(e_i - c_k)^2) * \exp(-(\dot{e}_i - c_k)^2)}{\sum_{k=1}^{L} (\exp(-(e_i - c_k)^2) * \exp(-(\dot{e}_i - c_k)^2))} \tag{17}$$

$$i = 1, 2...n; k = 1, 2...L$$

The input to the fuzzy unit is the velocity error $\dot{e}_i$ and the position error e_i.

3.3 Adaptive Laws and The Controller Design

The proposed fuzzy adaptive and fault-tolerant control is of the form

$$F = \Lambda^T(-k_0 S - \hat{w}^T\psi + U_r)$$
$$U_r = -\hat{\rho}\frac{(1+\|\hat{w}^T\psi\|)S}{\|S\|} \tag{18a}$$

with

$$\dot{\hat{w}} = \gamma_1 \psi S^T, \quad \dot{\hat{\rho}} = \gamma_2 \|S\|(1+\|\hat{w}^T\psi\|) \tag{18b}$$

where k_0 is a positive number chosen by the designer, $\hat{w}$ is the estimated value of w and $\hat{\rho}$ is the estimated value of ρ, $\gamma_1 > 0, \gamma_2 > 0$. U_r is to counteract the impact due to reconstruction error ξ and $\Lambda\mu(.)\Lambda^T$ as seen later.

Theorem 1

Consider the train dynamics as governed by (6) and (7). If the control scheme as given in (18) is applied, asymptotically stable position and speed tracking are ensured.

Proof

To show the stability of the control scheme, we first note that the closed loop error dynamics become:

$$M\dot{S} = -k_0\Lambda\mu(.)\Lambda^T S + \tilde{w}^T\psi + \eta + \Lambda\mu(.)\Lambda^T U_r \tag{19}$$

with $\eta = (I - \Lambda\mu(.)\Lambda^T)\hat{w}^T\psi + \xi$. It can be verified that:

$$\|\eta\| \le \|(I - \Lambda\mu(.)\Lambda^T)\|\|\hat{w}^T\psi\| + \|\xi\| \le \max\{\|(I - \Lambda\mu(.)\Lambda^T)\|, \|\xi\|\}(1+\|\hat{w}^T\psi\|) = \rho(1+\|\hat{w}^T\psi\|) \tag{20}$$

where $\rho = \max\{\|(I - \Lambda\mu(.)\Lambda^T)\|, \|\xi\|\}$ is some constant, which is unknown in general because both $\mu(.)$ and ξ are unavailable (although bounded).Now define a constant $0 < \varepsilon < \min \lambda_i$, where λ_i is the eigenvalue of the matrix $\Lambda\mu\Lambda^T$ (such constant does

exist because the matrix $\Lambda\mu\Lambda^T$ is symmetric and positive definite). Consider the Lyapunov function candidate:

$$V = \frac{1}{2}S^T MS + \frac{1}{2\gamma_1}(w-\hat{w})^T(w-\hat{w}) + \frac{1}{2\gamma_2\varepsilon}(\rho-\varepsilon\hat{\rho})^2 \tag{21}$$

It follows that:

$$\begin{aligned}\dot{V} &= S^T M\dot{S} + \frac{1}{\gamma_1}tr\left((-\dot{\hat{w}})^T(w-\hat{w})\right) + \frac{1}{\gamma_2}(-\dot{\hat{\rho}})(\rho-\varepsilon\hat{\rho}) \\ &= S^T\left\{\Lambda\mu(t)F + R(.)\right\} + tr\left((-S\psi^T)(w-\hat{w})\right) \\ &\quad + \left(-\|S\|(1+\|\hat{w}^T\psi\|)\right)(\rho-\varepsilon\hat{\rho}) \\ &= S^T\left\{\Lambda\mu(t)\Lambda^T\left[-k_0 S - \hat{w}^T\psi + U_r\right] + R(.)\right\} \\ &\quad + tr\left((-S\psi^T)(w-\hat{w})\right) + \left(-\|S\|(1+\|\hat{w}^T\psi\|)\right)(\rho-\varepsilon\hat{\rho})\end{aligned} \tag{22}$$

Using (18) (19) and (21), and with certain computation, it is not difficult to show that $\dot{V} \le -k_0\varepsilon\|S\|^2$, thus $S \in L_2 \cap L_\infty$, therefore $\dot{E}$, $E \in L_\infty$ from Equation (8). Then it is readily known that $\dot{S} \in L_\infty$. By Barbalat lemma it is concluded that $\lim_{t\to\infty} S = 0$, therefore $\dot{E} \to 0$, $E \to 0$ as $t \to \infty$ by the definition of S, which completes the proof.

It is noted that when S tends to zero, the control component U_r might involve chattering. A simple yet effective method to avoid this is as follows.

Theorem 2

Consider the train dynamics as governed by (6) and (7). If the following control scheme is applied, ultimately uniformly bounded stable position and speed tracking are ensured.

$$F = \Lambda^T(-k_0 S - \hat{w}^T\psi + U_r)$$

$$U_r = -\hat{\rho}\frac{(1+\|\hat{w}^T\psi\|)S}{\|S\|+\delta_0}$$

$$\dot{\hat{w}} = \gamma_1\psi S^T, \quad \dot{\hat{\rho}} = -\delta_1\hat{\rho} + \gamma_2\frac{\|S\|^2(1+\|\hat{w}^T\psi\|)}{\|S\|+\delta_0}$$

where $\delta_0 > 0$ and $\delta_1 > 0$ designer parameters related to tracking precision and updating rate.

4 Simulation

To verify the effectiveness of the proposed control scheme, simulation study is conducted. A train with 8 vehicles is simulated. The total mass of the eight vehicles is 345000kg. The actuator failure variable μ_i is set as a random function, taking value

between (0,1).The distribution matrix is B=diag([1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5]). Other control parameters are chosen as: $\gamma_1 = 5, \gamma_2 = 4, k_0 = 1000000$. A total of 10 fuzzy membership functions are used, where a series of c_k chosen for simulation are:

$c_1 = 0.1, c_2 = 0.2, c_3 = 0.4, c_4 = 0.6, c_5 = 0.8,$

$c_6 = -0.8, c_7 = -0.6, c_8 = -0.4, c_9 = -0.2, c_{10} = -0.1$

The goal is to make the actual velocity $\dot{y}$ track the desired velocity $\dot{y}^*$ and the actual position y track the desired position y^* with high precision.

The simulation results are depicted in Fig. 2- Fig. 4. As indicated from Fig. 2 and Fig. 3, position tracking and speed tracking are achieved with the proposed control. The position tracking error is shown in Fig. 4, one can observe the tracking error is small.

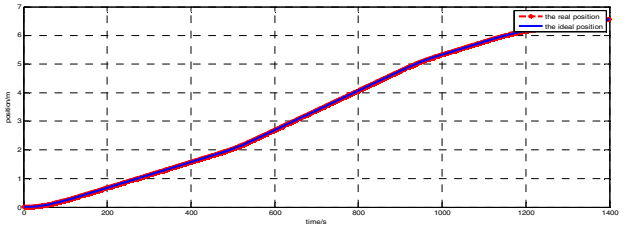


Fig. 2. Position tracking in 1400s

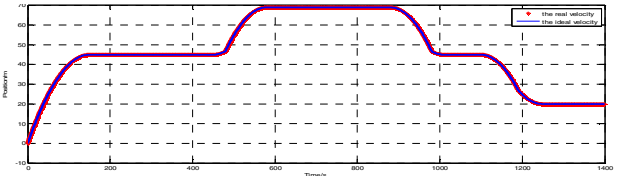


Fig. 3. Velocity tracking in 1400s

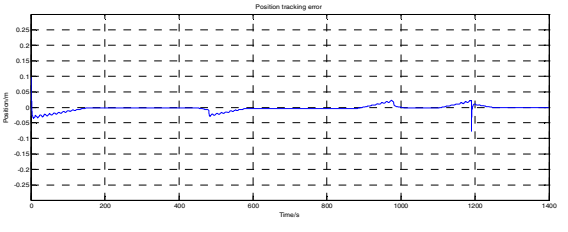


Fig. 4. The position tracking error

5 Conclusion

In this paper, a fuzzy-adaptive fault-tolerant control scheme based on multiple point mass model is presented for high speed train system in the presence of nonlinearity and uncertainty. Adaptive control algorithms are derived without using any explicit information on the system parameters. The adaptive control scheme guarantees that all signals involved are bounded and the system velocity and speed asymptotically track the corresponding desired trajectory. The computer simulation results show that the fuzzy-adaptive fault-tolerant controller is able to maintain good control precision.

Acknowledgement

This work was supported in part by National Natural Science Foundation of China under the grant No. 60974052, Program for Changjiang Scholars and Innovative Research Team in University (IRT0949), and Beijing Jiaotong University Research program (RCS2008ZT002, 2009JBZ001, and 2009RC008).

References

1. Song, Q., Song, Y.D.: Robust and adaptive control of high speed train systems. In: The 22th Chinese Control and Decision Conference, Xuzhou, China (May 2010)
2. Song, Q., Song, Y.D.: Adaptive control and optimal power/brake distribution of high speed trains with uncertain nonlinear couplers. In: Proceedings of the 29th Chinese Control Conference, Beijing, China (July 2010)
3. Fan, L.L., Song, Y.D.: On Fault-tolerant Control of Dynamic Systems with Actuator Failures and External Disturbances. Acta Automatica Sinica 36(11), 1620–1625 (2010)
4. Labiod, S., Boucherit, M.S.: Indirect fuzzy adaptive control of a class of SISO nonlinear system. The Arabian Journal for Science and Engineering 31(1B)
5. Liangji, H., Tao, T.: The analysis and design of the subway train's ATO system. Journal of Northern Jiaotong University 26(3) (June 2002)
6. Jie, G., Wenping, Z.: Investigation of Dynamic Modeling Method of Valve Train. In: 2nd International Conference on Mechanical and Electronics Engineering (2010)
7. Yang, C., Sun, Y.: Robust cruise control of high speed train with hardening/softening nonlinear coupler. In: Proc. American Control Conference, pp. 2200–2204 (1999)
8. Han, H., Su, C.-Y., Stepanenko, Y.: Adaptive control of a class of nonlinear system with nonlinearly parameterized fuzzy approximators. IEEE Transaction On Fuzzy Systems 9(2) (April 2001)
9. He, X., Tong, S., Zhang, W.: Fuzzy Adaptive Robust Fault-Tolerant Control for Uncertain Nonlinear Systems Based on Small-Gain Approach. In: Chinese Control and Decision Conference (2009)
10. Li, Y., Tong, S.: Adaptive Fuzzy Fault-Tolerant Control for Uncertain Nonlinear Systems. In: The 3rd International Conference on Innovative Computing Information and Control (2008)
11. Li, P., Yang, G.-H.: Adaptive Fuzzy Control of Unknown Nonlinear Systems with Actuator Failures for Robust Output Tracking. In: American Control Conference, Westin Seattle Hotel, Seattle, Washington, USA, June 11-13 (2008)

12. Hu, J.-m., Wang, Y.-h., Wang, X.: Fuzzy adaptive control for a class of nonlinear systems. In: Chinese Control and Decision Conference (2009)
13. Eryurek, E., Upadhyaya, B.R.: Fault-Tolerant Control and Diagnostics for Large-Scale systems. IEEE Control Systems (2005)
14. Guan, Y., Zhang, Y., Chen, X.: Application of Robust Fault-tolerant Control in Satellite Attitude control system. In: 3rd International Symposium on Systems and Control in Aeronautics and Astronautics (ISSCAA), June 8-10 (2010)
15. Hojati, M., Gazor, S.: Hybrid Adaptive Fuzzy Identification and Control of Nonlinear Systems. IEEE Transaction on Fuzzy Systems 10(2) (April 2002)
16. Wang, G.X.: Stable adaptive fuzzy control of nonlinear systems. IEEE Trans. Fuzzy System 1, 146–155 (1993)
17. Zhuan, X., Xia, X.: Optimal Scheduling and Control of Heavy Trains Equipped With Electronically Controlled Pneumatic Braking Systems. IEEE Transactions on Control Systems Technology 15(6), 1159–1166 (2007)

Robust Cascaded Control of Propeller Thrust for AUVs

Wei-lin Luo[1] and Zao-jian Zou[2]

[1] College of Mechanical Engineering and Automation,
Fuzhou University, Fujian 350108, China
[2] School of Naval Architecture, Ocean and Civil Engineering,
Shanghai Jiao Tong University, Shanghai 200240, China

Abstract. Robust neural-network controller of propeller thrust is proposed for autonomous underwater vehicles (AUVs). The cascaded plant consists of the dynamics of surge motion of an AUV, that of the propeller axial flow, that of the propeller shaft and that of the electrically-driven circuit. Uncertainties including modeling errors and external disturbances are taken into account simultaneously. A hybrid control strategy is proposed to the cascaded system with uncertainties. An on-line robust neural-network is used to compensate the modeling errors while L2-gain design is used to suppress the external disturbances. By backstepping method, the terminated control input to the thrust system is obtained. Design of the controller with L2-gain performance observes the recursive Lyapunov function method, which guarantee the uniformly ultimately bounded stability of tracking system. Simulation results demonstrate the validity of controller proposed.

Keywords: underwater vehicle; uncertainties; control of propeller thrust; neural-network; L2-gain.

1 Introduction

Focus has been increasingly directed to the accurate modeling and control of the autonomous underwater vehicles (AUVs). However, nonlinear dynamics and uncertainties of the underwater system make it challenging to achieve robust and accurate tracking and positioning. To accomplish appointed complicated underwater missions, many advanced control schemes have been developed to deal with the uncertainties. Healey and Lienard proposed sliding mode variable structure method to multi-degree-of-freedom AUVs [1]. Fossen applied adaptive control to underwater models with uncertain parameters [2]. Mu synthesis [3] and LMI synthesis [4] were used and compared with the sliding mode control. Neural networks controllers were proposed to the trajectory following of underwater vehicles [5]. An adaptive approach for motion control of underwater vehicles was proposed in the presence of external disturbance and uncertain parameters [6].Prasanth Kumar et al. applied a time delay control law to the trajectory control of AUVs [7]. Some work paid attention to the dynamics of actuators of AUVs. Fossen and Blanke used parameter adaptive control methods to the speed control of cascaded AUVs [8]. Pivano et al. applied state estimation method to the controller of a cascaded underwater vehicle [9].

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 574–582, 2011.

For a precise plant, satisfying performance can be obtained by using classic control strategies such as linearization feedback control, etc. However, some modeling errors are inevitable, such as parameter errors, high-order modes ignored and unmodelled dynamics of the underwater vehicles. The unmodelled dynamics perhaps result from thrust and torque losses caused by viscous drag effects, cross-coupling drag, varying wake with turn or sway, air suction and interaction between the thruster and the hull etc [10]. Moreover, external disturbances should be considered. This uncertainty may refer to some unknown random noises from mechanical or electrical equipments, or the environmental forces such as currents. Usually, both unmodelled dynamics and external disturbance cannot be modeled by mathematical expressions.

This paper presents a hybrid controller. H-infinity control and neural network (NN) control are both used. Uncertainties including modeling errors and external disturbances are taken into account for a cascaded system simultaneously. The controller design observes the recursive Lyapunov function method to guarantee the robustness and uniformly ultimate bounded stability of the error tracking system.

2 Problem Formulations

The plant consists of the dynamics of surge motion of an AUV, that of the propeller axial flow, that of the propeller shaft and that of the electrically-driven circuit.

2.1 AUV and Propeller Axial Flow Dynamics

Without loss of generality, the surge motion of an underwater vehicle equipped with a single propeller aft of hull is considered, which is directly related to propeller thrust. And it is assumed that the propeller is driven by a DC motor. The surge motion is

$$(m - X_{\dot{u}})\dot{u} - X_u u - X_{u|u|} u|u| = (1 - t_p)T, \tag{1}$$

where u is the surge velocity; m is the mass of the underwater vehicle; $X_{\dot{u}}$ is the added mass; X_u and $X_{u|u|}$ are the linear and nonlinear hydrodynamic derivatives respectively; t_p is the thrust deduction coefficient; T is the thrust force.

Considering the affecting effect caused by the propeller flow [8], one has

$$m_f \dot{u}_p + d_{f0} u_p + d_f |u_p|(u_p - u_a) = T, \tag{2}$$

where u_p is the propeller axial flow velocity; m_f is the mass of the control volume; d_{f0} is the linear damping coefficient; d_f is the quadratic damping coefficient; u_a is the advance velocity to the propeller.

Taking account of the propeller axial flow dynamics [8], the quasi-steady thrust and torque can be calculated as

$$T = T_{n|n|} n|n| - T^0_{|n|u_a} |n| u_p, \tag{3}$$

$$Q = Q_{n|n|} n|n| - Q^0_{|n|u_a} |n| u_p, \tag{4}$$

where the coefficients are

$$T_{n|n|} = \rho D^4 \alpha_0, T_{|n|u_a} = \rho D^3 \alpha_1, Q_{n|n|} = \rho D^5 \beta_0, Q_{|n|u_a} = \rho D^4 \beta_1, T^0_{|n|u_a} = \frac{T_{|n|u_a}}{1+a}, Q^0_{|n|u_a} = \frac{Q_{|n|u_a}}{1+a},$$

where ρ is the water density; D is the propeller diameter; $\alpha_0, \alpha_1, \beta_0$ and β_1 are the constants in the linearly regressive formulas of thrust coefficient K_T and torque coefficient K_Q respectively; a is a ratio determined by u_p and u_a.

2.2 Actuator Dynamics

Assume that the propeller is driven by a DC motor. The dynamics of propeller shaft and that of the electrically-driven circuit are given as [11]

$$J_m \dot{n} + K_n n + Q = KI, \tag{5}$$

$$L\dot{I} + RI + Kn = u_e, \tag{6}$$

where J_m is the inertia moment; n is the propeller revolution; K_n is the damping coefficient; K is the conversion coefficient; I is the electrical current; L is the armature inductance; R is the resistance; u_e is the applied voltage.

2.3 Cascaded System with Uncertainties

Taking uncertainties into account, a plant of propeller thrust is the cascaded system

$$\left.\begin{aligned} &(m - X_{\dot{u}})\dot{u} - X_u u - X_{u|u|} u|u| + \Delta_1 + w = (1-t_p)T \\ &J_m \dot{n} + K_n n + Q + \Delta_2 = KI \\ &L\dot{I} + RI + Kn = u_e \end{aligned}\right\}, \tag{7}$$

where Δ_1 and Δ_2 are the modeling errors, and Δ_2 refers to torque losses [9]. w is a bounded external disturbance signal. And the first equation of (7) is coupling with (2).

As seen, the terminated control input is u_e, while both I and n are interim state variables. In this paper, the assumed object is to design appropriate u_e, so that the actual surge speed u can track the desired speed u_d well.

3 Controller Design

Above all, without loss of generality, two assumptions are given as [A1] The desired speed u_d is differentiable and the system is controllable if $w = 0$ and $u(t_0) = u_d(t_0)$.

[A2] The external disturbance is bounded in norm as $\|w\| \le \overline{w}, \forall t \ge 0, \exists \overline{w} > 0$, where $\|\cdot\|$ denotes Euclidian norm.

3.1 Feedback Linearization Design

To certain system, the feedback linearization control can guarantee the good performance of close-loop tracking system. Hence, to the certainties in the cascaded system (7), several desired controllers are given first of all

$$T_d = [(m - X_{\dot{u}})\dot{u}_d - X_u u_d - X_{u|u|}|u|u_d + u_1]/(1 - t_p), \tag{8}$$

$$I_d = (J_m \dot{n}_d + K_n n_d + Q_d + u_2)/K, \tag{9}$$

$$u_e = RI_d + Kn_d + u_3, \tag{10}$$

where T_d is the desired thrust force; I_d is the desired electrical current; u_1, u_2 and u_3 are auxiliary controllers. For desired controls, denoting several tracking errors as

$$e = u_d - u, \varepsilon_1 = T_d - T, \varepsilon_2 = n_d - n, \varepsilon_3 = I_d - I .$$

The cascaded system (7) can be reduced to the following error system

$$\left.\begin{aligned} \dot{e} &= [X_u e + X_{u|u|}|u|e + \Delta_1 + w + (1 - t_p)\varepsilon_1 - u_1]/(m - X_{\dot{u}}) \\ \dot{\varepsilon}_2 &= [-K_n \varepsilon_2 - a_1 \varepsilon_1 - a_2(|n_d|n_d - |n|n) + K\varepsilon_3 + \Delta_2 - u_2]/J_m \\ \dot{\varepsilon}_3 &= (-R\varepsilon_3 - K\varepsilon_2 + L\dot{I}_d - u_3)/L \end{aligned}\right\}, \tag{11}$$

Obviously, appropriate designs of three auxiliary controllers, i.e. u_1, u_2 and u_3, answer for the performance of the tracking system. They can be obtained by using the method of Lyapunov recursive function method.

3.2 Lyapunov Recursive Function Method

Lyapunov recursive function method is used to the controller design. By using this method, the controller design complies with a systematic and simple procedure. Meanwhile, the robustness and stability of the tracking system can be guaranteed [12].

A positive definite Lyapunov function candidate is given as for the system (11)

$$V_0 = \frac{1}{2}(m - X_{\dot{u}})e^2 + \frac{1}{2}J_m \varepsilon_2^2 + \frac{1}{2}L\varepsilon_3^2 . \tag{12}$$

Note that the error ε_1 isn't involved in the candidate function. Its stability can be guaranteed if the error e and ε_2 converge.

The derivative of V_0 satisfies

$$\begin{aligned} \dot{V}_0 \le\ & e[X_{u|u|}|u|e + \Delta_1 + w + (1 - t_p)\varepsilon_1 - u_1] \\ & + \varepsilon_2[-a_1\varepsilon_1 - a_2(|n_d|n_d - |n|n) + \Delta_2 - u_2] + \varepsilon_3(L\dot{I}_d - u_3) . \end{aligned} \tag{13}$$

Because the mathematical expressions of uncertainties Δ_1, Δ_2 and w are unknown, neural-networks and L_2-gain design are used to compensate these uncertainties.

3.3 L2-Gain Design

To external disturbance w, an evaluation signal z is given as

$$\int_0^\tau \|z\|^2 dt \le \int_0^\tau \gamma^2 \|w\|^2 dt + \gamma_0, \tag{14}$$

to guarantee robust performance [12], where γ and γ_0 are small positive constants.

Let $z = re$ $(r > 0)$, incorporating the L_2-gain index into $\dot{V}_0$ yields

$$\begin{aligned} \dot{V}_0 + \|z\|^2 - \gamma^2 \|w\|^2 \le e[X_{u|u|}|u|e + \Delta_1 + \frac{1}{4\gamma^2}e + r^2 e + (1-t_p)\varepsilon_1 - u_1] \\ + \varepsilon_2[-a_1\varepsilon_1 - a_2(|n_d|n_d - |n|n) + \Delta_2 - u_2] + \varepsilon_3(L\dot{I}_d - u_3), \end{aligned} \tag{15}$$

3.4 Neural-Network Identifiers

Uncertainties in (15), Δ_1 and Δ_2, will be compensated by NN. And because it is tedious and difficult to obtain the explicit expression of $\dot{I}_d$, it is also identified by NN. In this paper, a two-layer feedforward NN is applied. As pointed out, this NN is a universal approximation of nonlinear functions with any accuracy provided the activation function is selected as basic or squashing one and appropriate number of the hidden layer nodes exist [13].

Let three nonlinear functions be approximated by NN

$$X_{u|u|}|u|e + \Delta_1 + (1-t_p)\varepsilon_1 = \boldsymbol{W}_1^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{h}_1) + \eta_1, \tag{16}$$

$$-a_1\varepsilon_1 - a_2(|n_d|n_d - |n|n) + \Delta_2 = \boldsymbol{W}_2^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{h}_2) + \eta_2, \tag{17}$$

$$L\dot{I}_d = \boldsymbol{W}_3^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{h}_3) + \eta_3, \tag{18}$$

where $\boldsymbol{W}_i$ $(i = 1,2,3)$ is the so-called ideal weight vector and satisfies $\|\boldsymbol{W}_i\|_F \le \boldsymbol{W}_{iM}$ $(\boldsymbol{W}_{iM} > 0)$; $\|\cdot\|_F$ denotes the Frobenius norm; $\boldsymbol{\Phi}(\cdot)$ is the activation function of hidden layer; $\boldsymbol{h}_i$ is the preprocessed input vector; η_i is the reconstruction error and will satisfy $\|\eta_i\| \le \eta_{iN}$ $(\eta_{iN} > 0)$.

The three auxiliary controllers can be designed as

$$u_1 = \boldsymbol{W}_{1e}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{h}_1) + e/4\gamma^2 + r^2 e + \lambda_1(m - X_{\ddot{u}})e, \tag{19}$$

$$u_2 = \boldsymbol{W}_{2e}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{h}_2) + \lambda_2 J_m \varepsilon_2, \tag{20}$$

$$u_3 = \boldsymbol{W}_{3e}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{h}_3) + \lambda_3 L\varepsilon_3, \tag{21}$$

where λ_i is a positive control gain; $\boldsymbol{W}_{ie}$ is the updated weight vector designed as

$$\dot{\boldsymbol{W}}_{1e} = k_1 \boldsymbol{\Phi}(\boldsymbol{h}_1) e - k_2 \|\boldsymbol{\xi}\| \boldsymbol{W}_{1e}, \tag{22}$$

$$\dot{\boldsymbol{W}}_{2e} = k_3 \boldsymbol{\Phi}(\boldsymbol{h}_2) \varepsilon_2 - k_4 \|\boldsymbol{\zeta}\| \boldsymbol{W}_{2e}, \tag{23}$$

$$\dot{\boldsymbol{W}}_{3e} = k_5 \boldsymbol{\Phi}(\boldsymbol{h}_3) \varepsilon_3 - k_6 \|\boldsymbol{\varsigma}\| \boldsymbol{W}_{3e}, \tag{24}$$

where $k_{1\sim6}$ are positive constants; and three general error vectors are introduced as $\boldsymbol{\xi} = [e \quad \varepsilon_2]^{\mathrm{T}}$, $\boldsymbol{\zeta} = [e \quad \varepsilon_1 \quad \varepsilon_2]^{\mathrm{T}}$, $\boldsymbol{\varsigma} = [e \quad \varepsilon_1 \quad \varepsilon_2 \quad \varepsilon_3]^{\mathrm{T}}$.

With the auxiliary controllers, the three desired controllers can be calculated as

$$\begin{aligned} T_d = \frac{1}{(1-t_p)} [&(m - X_{\dot{u}}) \dot{u}_d - X_u u_d - X_{u|u|} |u| u_d \\ &+ \boldsymbol{W}_{1e}^{\mathrm{T}} \boldsymbol{\Phi}(\boldsymbol{h}_1) + \frac{1}{4\gamma^2} e + r^2 e + \lambda_1 (m - X_{\dot{u}}) e], \end{aligned} \tag{25}$$

$$I_d = \frac{1}{K} (J_m \dot{n}_d + K_n n_d + a_1 T_d + a_2 n_d |n_d| + \boldsymbol{W}_{2e}^{\mathrm{T}} \boldsymbol{\Phi}(\boldsymbol{h}_2) + \lambda_2 J_m \varepsilon_2), \tag{26}$$

$$u_e = R I_d + K n_d + \boldsymbol{W}_{3e}^{\mathrm{T}} \boldsymbol{\Phi}(\boldsymbol{h}_3) + \lambda_3 L \varepsilon_3. \tag{27}$$

Note that no explicit expression of the desired propeller revolution is given. A low-pass filter is usually applied to the desired propeller revolution [8]

$$\ddot{n}_d + 2\omega_f \dot{n}_d + \omega_f^2 n_d = \omega_f^2 n_d', \tag{28}$$

where ω_f is the cut-off frequency and n_d' is obtained from the equation (3)

$$n_d' = \left[T_{|n|u_a}^0 u_p + sign(T_d) \sqrt{\left| \left(T_{|n|u_a}^0 u_p \right)^2 + 4 T_{n|n|} T_d \right|} \right] / 2 T_{n|n|}. \tag{29}$$

4 Stability Analyses

Substituting equations (16)-(21) into (15) yields

$$\begin{aligned} \dot{V}_0 + \|z\|^2 - \gamma^2 \|w\|^2 \le &-\lambda_1 (m - X_{\dot{u}}) e^2 - \lambda_2 J_m \varepsilon_2^2 - \lambda_3 L \varepsilon_3^2 + e \tilde{\boldsymbol{W}}_1^{\mathrm{T}} \boldsymbol{\Phi}(\boldsymbol{h}_1) \\ &+ e\eta_1 + e \tilde{\boldsymbol{W}}_2^{\mathrm{T}} \boldsymbol{\Phi}(\boldsymbol{h}_2) + e\eta_2 + \varepsilon_3 \tilde{\boldsymbol{W}}_3^{\mathrm{T}} \boldsymbol{\Phi}(\boldsymbol{h}_3) + \varepsilon_3 \eta_3, \end{aligned} \tag{30}$$

where $\tilde{\boldsymbol{W}}_i = \boldsymbol{W}_i - \boldsymbol{W}_{ie}$ is the weight error vector.

To guarantee the robustness of NN, a step Lyapunov function candidate is given

$$V_1 = V_0 + \frac{1}{2k_1} tr\{\tilde{\boldsymbol{W}}_1^{\mathrm{T}} \tilde{\boldsymbol{W}}_1\} + \frac{1}{2k_3} tr\{\tilde{\boldsymbol{W}}_2^{\mathrm{T}} \tilde{\boldsymbol{W}}_2\} + \frac{1}{2k_5} tr\{\tilde{\boldsymbol{W}}_3^{\mathrm{T}} \tilde{\boldsymbol{W}}_3\}. \tag{31}$$

Introduce a constant $0 \le \lambda_0 \le 1$ and let

$$\alpha_0 = \min\{(1-\lambda_0)\lambda_1, (1-\lambda_0)\lambda_2, (1-\lambda_0)\lambda_3, (1-\lambda_0)k_2 \|\boldsymbol{\xi}\|, (1-\lambda_0)k_4 \|\boldsymbol{\zeta}\|, (1-\lambda_0)k_6 \|\boldsymbol{\varsigma}\|\}.$$

One has

$$\begin{aligned} \dot{V}_1 + \|z\|^2 - \gamma^2 \|w\|^2 \le -2\alpha_0 V_1 + \|\boldsymbol{\varsigma}\|(B - b_1 \|\boldsymbol{\varsigma}\| + \frac{k_2}{4k_1\lambda_0} W_{1M}^2 \\ + \frac{k_4}{4k_3\lambda_0} W_{2M}^2 + \frac{k_6}{4k_5\lambda_0} W_{3M}^2), \forall B > 0, \forall b_2 > b_1 > 0. \end{aligned} \tag{32}$$

To guarantee the right-hand side of the above inequality negative, if only it holds

$$\|\boldsymbol{\varsigma}\| \ge \frac{B}{b_1} + \frac{k_2}{4k_1\lambda_0} W_{1M}^2 + \frac{k_4}{4k_3\lambda_0} W_{2M}^2 + \frac{k_6}{4k_5\lambda_0} W_{3M}^2 . \tag{33}$$

It can be achieved by appropriate parameters. Usually, larger $\lambda_1, \lambda_2, \lambda_3, k_1, k_3, k_5$, against smaller k_2, k_4, k_6, will improve the tracking accuracy of the control system.

5 Simulation Results

The parameters of vehicle model and DC-motor are from references [8] and [11]. The desired surge speed is assumed $u_d = \sin 0.2t$ and initial deviation exists, $u(0) = 0.5$. The controller gains are given $\lambda_1 = \lambda_2 = \lambda_3 = 1$ and parameters of the neural network weights are given as: $k_1 = 5 \times 10^3$, $k_2 = 25$, $k_3 = 50$, $k_4 = 0.5$, $k_5 = 0.5$, $k_6 = 0.05$. The initial weights are set zero; the activation function of the hidden layer is selected as sigmoid function. The preprocessed inputs to hidden layer are given

$$\boldsymbol{h}_1 = \begin{bmatrix} (u - u_p) & e & \varepsilon_2 & 1 \end{bmatrix}^{\mathrm{T}}, \boldsymbol{h}_2 = \begin{bmatrix} e & \varepsilon_1 & \varepsilon_2 & 1 \end{bmatrix}^{\mathrm{T}}, \boldsymbol{h}_3 = \begin{bmatrix} e & \varepsilon_1 & \varepsilon_2 & \varepsilon_3 & 1 \end{bmatrix}^{\mathrm{T}}.$$

The parameters of the L_2-gain index are $\gamma = 0.02, r = 0.1$; the modeling error is assumed as $\Delta_1 = 50 \sin e$; the torque losses Δ_2 is assumed 5% of Q; the external disturbance is assumed as a random normally distributed noise constrained in [-150N +150N]; The simulation results are shown in figures 1 to 3.

As it can be seen from the simulation results, not only the tracking errors but also the updated weights of neural-networks are uniformly ultimately bounded, under the condition of external disturbance and initial deviation from desired. The robustness and stability are both guaranteed, tracking accuracy as well.

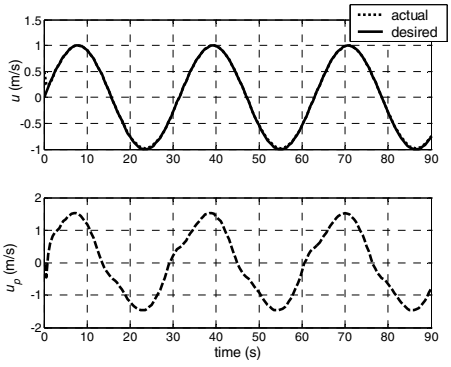


Fig. 1. Surge velocity and axial flow velocity

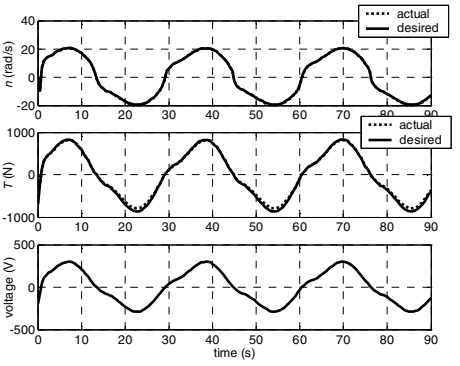


Fig. 2. Control inputs

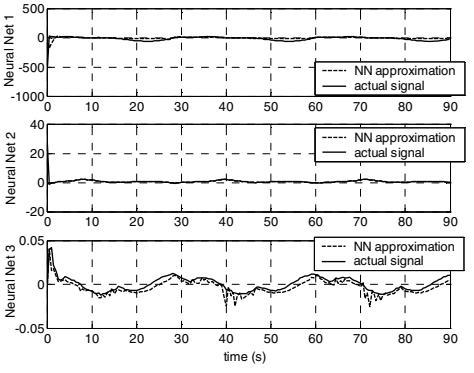


Fig. 3. Approximation abilities of neural-networks

6 Conclusions

Study is presented with respect to the propeller thrust control of an AUV. Uncertainties, including modeling error and external disturbances are taken into account. Neural-networks and L_2-gain design are used to compensate the uncertainties respectively. The uniformly ultimately bounded stabilities of the tracking errors and the neural-network weights errors are guaranteed by using the Lyapunov recursive function method. An appropriately selected set of parameters in the controller improves tracking accuracy. Simulation results have demonstrated the validity of the controller proposed. More efforts will be devoted to expand this method to the control of a cascaded underactuated underwater vehicle.

Acknowledgments. This work was supported by the National Natural Science Foundation of P.R. China (No. 51079031; 50979060) and the Natural Science Foundation of Fujian Province of P.R. China (No. 2010J01004).

References

1. Healey, A.J., Lienard, D.: Multivariable Sliding-mode Control for Autonomous Diving and Steering of Unmanned Underwater Vehicles. IEEE J. Oc. Eng. 18(3), 327–339 (1993)
2. Fossen, T.I.: Guidance and Control of Ocean Vehicles. John Wiley & Sons, New York (1994)
3. Campa, G., Innocenti, M.: Robust Control of Underwater Vehicles: Sliding Mode Control vs. Mu Synthesis. In: IEEE Conference on Oceans, pp. 1640–1644. IEEE Press, New York (1998)
4. Innocenti, M., Campa, G.: Robust Control of Underwater Vehicles: Sliding Mode Control vs. LMI Synthesis. In: American Control Conference, pp. 3422–3426 (1999)
5. Vukic, Z., Borovic, B., Beale, G.O.: Trajectory Following Neuro Controller for Submarine Maneuvering in Horizontal and Vertical Plane. In: 9th Mediterranean Conference on Control and Automation, paper No.112 (2001)
6. Antonelli, G., Caccavale, F., Chiaverini, S., Fusco, G.: A Novel Adaptive Control Law for Underwater Vehicles. IEEE Trans. on Cont. Sys. Tech. 11(2), 221–232 (2003)
7. Prasanth Kumar, R., Dasgupta, A., Kumar, C.S.: Robust Trajectory Control of Underwater Vehicles. Oc. Eng. 34(5-6), 842–849 (2007)
8. Fossen, T.I., Blanke, M.: Nonlinear Output Feedback Control of Underwater Vehicle Propeller Using Feedback Form Estimation Axial Flow Velocity. IEEE J. Oc. Eng. 25(2), 241–255 (2000)
9. Pivano, L., Johansen, T.A., Smogeli, N., Fossen, T.I.: Nonlinear Thrust Controller for Marine Propellers in Four-quadrant Operations. In: American Control Conference, pp. 900–905 (2007)
10. Blank, M., Lindegaard, K.P., Fossen, T.I.: Dynamic Model for Tthrust Generation of Marine Propellers. In: MCMC 2000, pp. 363–368 (2000)
11. Indiveri, G., Zanoli, S.M., Parlangeli, G.: DC Motor Control Issues for UUVs. In: 14th Mediterranean Conference on Control and Automation (2006)
12. Ishii, C., Shen, T.L., Qu, Z.H.: Lyapunov Recursive Design of Robust Adaptive Tracking Control with L2-gain Performance for Electrically-driven Robot Manipulators. Int. J. Con. 74(8), 811–828 (2001)
13. Kwan, C., Lewis, F.L., Dawson, D.M.: Robust Neural-network Control of Rigid-link Electrically Driven Robots. IEEE Trans. Neu. Net. 9(4), 581–588 (1998)

A Developmental Learning Based on Learning Automata

Xiaogang Ruan[1], Lizhen Dai[1], Gang Yang[2], and Jing Chen[1]

[1] Institute of Artificial Intelligence and Robots,
College of Electronic and Control Engineering,
Beijing University of Technology,
Beijing, 100124, P.R. China
[2] Intelligent Systems Institute,
College of Electronic Information and Control Engineering,
Beijing University of Technology, Beijing, 100124, P.R. China
adrxg@bjut.edu.cn, alice.dai2011@gmail.com

Abstract. This paper presents a new method for developmental robot based on a learning automaton. This method can be considered as active learning to select the best action in order to quickly adapt environment. The new model built in the framework of learning automata theory is an abstract and formal mathematical tool to describe the cognitive behavior or cognitive development mechanisms and provide an effective logical structure for the design of cognitive and development robots. The model reflects the design principles of cognitive development robotics. The direction of cognition and development is towards entropy minimization and simulation results verify its effectiveness.

Keywords: Learning Automaton, reward, active learning, developmental robot.

1 Introduction

The late 60s and the early 70s of last century, Stanford Research Institute developed a mobile robot called Shakey [1], having the ability of logical reasoning and behavior planning, which is not only considered the first intelligent robot, but also be regarded as the birth symbol of intelligent robotics.

In a sense, the robot lack of cognitive ability is not truly intelligent robot. In 1997, Professor Brooks from Artificial Intelligence Laboratory of MIT proposed the concept of cognitive robotics [2]. Cognitive robotics is aimed to give the robot cognitive ability that makes the robot form and develop knowledge and skills independently and gradually through cognitive in the process of interaction with the environment.

Doctor Chatila [3], EU COGNIRON project coordinator, director of French National Scientific Research Center System Structure and Analysis Laboratory, considered that learning is the basic elements of cognitive robots. In recent years, how the robot form and develop their own ability to solve problems independently and progressively was gradually emerged from cognitive robotics.

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 583–590, 2011.

Thus developmental robotics was derived. Here, the development does not refer to the body's development, but the mental or cognitive development of robot that can also be said to be knowledge and capacity development of robot.

Developmental robot imitates the development process of human brain and human mental that requires the robot learn the knowledge independently to complete various tasks in real environment and organize the knowledge in memory system. Researchers have proposed many different development models [4]. Weng [5] proposed IPCA + HDR tree model which mainly includes two basic algorithms: the incremental principal component analysis (IPCA) algorithm and hierarchical discriminating regression (HDR) tree algorithm. The output of the former is the latter's input. It can make appropriate response to the changing environmental in real time and is well applied to real-time development and autonomous incremental learning for robot. But this model lacks the ability of high-level decision-making and task-distinguishing, so it is difficult to complete more complex tasks. For these reasons, Tan et al. [6] proposed task-oriented developmental learning (TODL) model on the basis of this model. The model learning for the tasks can make the robot have the ability to handle multiple tasks, which improves the performance greatly. Blank et al. [7] put forward a hierarchical development model based on extraction and prediction mechanism, where extraction mechanism is achieved by self-organizing map (SOM), and prediction mechanism adopts a simple return network (SRN). The main deficiency of this model is complex structure, high-level decision-making action is too large, and lack ability of planning specific goals and tasks. Schema model is a developmental model proposed by Stojanov [8], and the thought comes mainly from the 20th century's greatest developmental psychologists Piagetian theories of cognitive development. This model imitates the development processes of human cognitive well and has strong robustness and adaptability. But convergence speed will be affected when the perception states are too much for it will greatly increase computation time.

Itoh et al. [9] implemented a new behavior model into the Emotion Humanoid Robot WE-4R and the robot could select suitable behavior according to the situation within a predefined behavior list. However, the robot can have just one recognition system in response to a stimulus although humans retrieve various memories in response to a stimulus according to their mood. We have presented a compute model with probabilistic automata [10]. Then we improved it and developed a new developmental model to simulate a bionic autonomous learning process.

2 Development on Psychology

Developmental psychology is concerned not only with describing the characteristics of psychological change over time, but also seeks to explain the principles and internal workings underlying these changes. Psychologists have attempted to better understand these factors by using models. Developmental models are sometimes computational, but they do not need to be. A model must simply

account for the means by which a process takes place. This is sometimes done in reference to changes in the brain that may correspond to changes in behavior over the course of the development. Computational accounts of development often use either symbolic, connectionist, or dynamical systems models to explain the mechanisms of development.

Piaget was one of the influential early psychologists to study the development of cognitive abilities. His theory suggests that development proceeds through a set of stages from infancy to adulthood and that there is an end point or goal. Modern cognitive development has integrated the considerations of cognitive psychology and the psychology of individual differences into the interpretation and modeling of development [11]. Specifically, the neo-Piagetian theories of cognitive development showed that the successive levels or stages of cognitive development are associated with increasing processing efficiency and working memory capacity. These increases explain progression to higher stages, and individual differences in such increases by same-age persons explain differences in cognitive performance. Other theories have moved away from Piagetian stage theories, and are influenced by accounts of domain-specific information processing, which posit that development is guided by innate evolutionarily specified and content-specific information processing.

3 Experimental Model of Developmental Robot Based on Learning Automata

To some extent, a learning automaton is a life model, but it is also a calculation model from the mathematic perspectives. The learning automata have strict mathematical definition. Defining: a learning automata is a eight-tupe: LA=$(t, \Omega, S, \Gamma, \delta, \varepsilon, \eta, \Psi)$.

(1) $t \in \{0, 1, \cdots, n_t\}$ is discrete time, t=0 is initial time.
(2) $\Omega = \{\alpha_k \mid k = 0, 1, 2, \cdots, n_\Omega\}$ is the set of all actions of the model.
(3) $S = \{s_i \mid i = 0, 1, 2, \cdots, n_s\}$, is the set of all states of the model.
(4) $\Gamma = \{r_{ik(p)} \mid p \in P; i \in \{0, 1, 2, \cdots, n_s\}; k \in \{0, 1, \cdots, n_\Omega\}\}$, is the set of all action rules of the model. Random rule $r_{ik(p)} : s_i \to \alpha_k(p)$ means that LA implements action $\alpha_k \in \Omega$ in accordance with the probability $p \in P$, when its state is $s_i \in S$, $p = p_{ik} = p(\alpha_k \mid s_i)$ is a probability of action α_k in the state s_i, P represents the set of p_{ik}.
(5) $\delta : S(t) \times \Omega(t) \to S(t+1)$, is the state transition function. At the time $t+1$ the state $s(t+1) \in S$ is confined by the state $s(t) \in S$ and action $a(t) \in \Omega$ at time t, which has no relationship with the state and action before time t.
(6) $\varepsilon : S \to E = \{\varepsilon_i \mid i = 0, 1, 2, \cdots, n_s\}$, is orientation function of LA, $\varepsilon_i = \varepsilon(s_i) \in E$ is orientation of state $s_i \in S$.
(7) $\eta : \Gamma(t) \to \Gamma(t+1)$ is developmental learning law, which is defined as

$$\eta : \begin{cases} \text{IF } s(t) = s_i,\ \alpha(t) = \alpha_k,\ s(t+1) = s_j \\ \text{THEN} \begin{cases} p_{ik}(t+1) = p_{ik}(t) + \Delta \\ p_{iu}(t+1) = p_{iu}(t) - \Delta\xi,\ \forall u \neq k \end{cases} \end{cases} \text{ and } \begin{cases} \Delta = \phi(\overline{\varepsilon}_{ij}), 0 \leq p_{ik} + \Delta \leq 1 \\ \xi = p_{iu}(t) / \sum_{v \neq k} p_{iv}(t) \end{cases}$$

There $\overrightarrow{\varepsilon}_{ij} = \varepsilon(s_j) - \varepsilon(s_i)$; $\phi(x)$ is monotonic increasing function, satisfied $\phi(x) \equiv 0$ if $x = 0$; $p_{ij}(t)$is a probability of action α_k in the state s_i at time t. Regulate action rule $r_{ik}(p) \in \Gamma$ and action probability $p \in P$ will change. Suppose LA implement action $\alpha(t) \in \Omega$ in the state $s(t)$ at time t, at time $t+1$ LA is in the state $s(t+1)$, according to the developmental rule, if $\varepsilon(s(t+1)) - \varepsilon(s(t)) < 0$ then $p(\alpha(t) \mid s(t))$ will decrease, vice versa, if $\varepsilon(s(t+1)) - \varepsilon(s(t)) > 0$ then $p(\alpha(t) \mid s(t))$ will increase. There Δ has a relationship with orientation ε, the greater the orientation values the better the results of actions. At the same time $\sum_k p_{ik}(t) = 1$ should be satisfied.

(8) $\Psi : P \times E \to R^+$ is action entropy of LA, R^+ is the set of positive real number. At time t action entropy of LA is $\Psi(t)$ which means the sum of action entropy under the conditions of s_i at the moment t, it is given by:

$$\Psi(t) = \Psi(\Omega(t) \mid S) = \sum_{i=0}^{n_s} p_i \Psi_i(t) = \sum_{i=0}^{n_s} p(s_i) \Psi_i(\Omega(t) \mid s_i).$$

And $\Psi_i(t)$ is the action entropy of LA at the moment t, namely:

$$\Psi_i(t) = \Psi_i(\Omega(t) \mid S_i) = -\sum_{k=1}^{n_\Omega} p_{ik} \log_2 p_{ik} = -\sum_{k=1}^{n_\Omega} p(\alpha_k \mid s_i) \log_2 p(\alpha_k \mid s_i).$$

So, $\Psi(t) = -\sum_{i=0}^{n_s} p(s_i) \sum_{k=1}^{n_\Omega} p(\alpha_k \mid s_i) \log_2 p(\alpha_k \mid s_i)$.

If the action entropy of LA tends to become smaller and smaller, and in time $t \to \infty$ tend to be the minimum, then action entropy of LA is convergent. The system self-organization process is a process of drawing information, draw negative entropy and eliminating uncertainty.

4 The Structure and Learning Mechanism of Learning Automata

For the shaking hands experiment of development robot, the structure of learning automata is shown in Fig.1. The environment is a relative concept which is on behalf of a system interacting with learning system. It is providing external information of the dynamic behavior for the learning system. Learning system obtains information from the environment, processes information online, and finally achieves the desired objectives. Each stage of the learning process requires two steps: step 1, the learning automaton selects a specific output behavior α_i; Step 2, learning system obtains the behavioral response from the environment for the first step and automatically updates the action probability p_i according to the response of α_i, which will affect the choice of future behavior.

Automating the process of learning can be summarized as follows: at each learning stage, the automaton chooses a random act of behavior from the limited options according to certain probability distribution and outputs to the environment. Then the environment returns a reward or punishment signal to the automaton as the response to the choice of corresponding actions. Automaton updates its probability of the action selection according to the reaction from

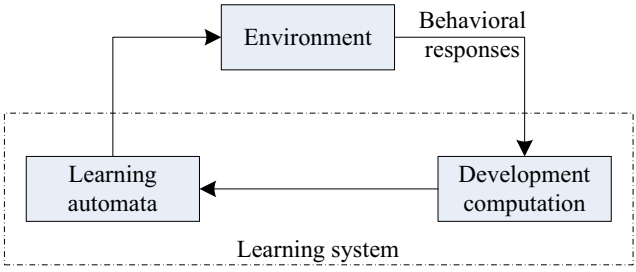


Fig. 1. Structure of learning system based on learning automata

environment, and chooses a new behavior in accordance with the modified action selection probability. At the beginning, reward/punishment probabilities of all actions are unknown, so the learning automata select individual behavior according to uniform distribution (i.e., equal probability) initially. However, with repeated exchanges with the environment, automata get to know the environment characteristics of reward and punishment, and eventually tend to select the behavior of large rewarded probability or small punished probability with a larger probability. The developmental learning process runs recursively in accordance with the following procedural steps:

Step 1: initialization. Set $t = 0$, initial state is $s(0)$, learning rate is α, initial action probability is $p_{ik} = 1/n_{\Omega}$ $(i = 0, 1, \cdots, n_S; k = 0, 1, \cdots, n_{\Omega})$

Step 2: chose action. According to developmental rule, LA selects an action $\alpha(t) \in \Omega$ randomly.

Step 3: implement action.

Step 4: observe state. According to the state transition function $\delta : S(t) \times \Omega(t) \rightarrow S(t+1)$, the result of state transition is fully capable of observing.

Step 5: development learning. LA implements action at time t, not only the state of LA shifts, the implemental probability of various actions in the next moment updates as:

$$\eta : \begin{cases} p_{ik}(t+1) = p_{ik}(t) + \Delta \\ p_{iu}(t+1) = p_{iu}(t) - \Delta\xi, \ u \neq k \end{cases}$$

Step 6: calculate action entropy. By the action entropy formula

$$\Psi(t) = -\sum_{i=0}^{n_s} p(s_i) \sum_{k=1}^{n_{\Omega}} p(\alpha_k \mid s_i) \log_2 p(\alpha_k \mid s_i)$$

Step 7: recursive transfer. If $t + 1 \leq T_f$, then $t = t + 1$ and go to step 2; or else end.

5 Simulation Results and Analysis

We implemented the developmental model into a robot which quickly increased robot need and restricted behaviors except for shaking hands. In the experiment, the robot repeated the action of shaking hands with us and we reacted to it following the procedure shown in Fig.2. We set the experimental condition as follows: the initial state is randomly given non-stroking s_0 or non-hitting state s_1; the initial probabilities distribution of shaking two hands respectively is $P = (p_R, p_L) = (0.5, 0.5)$; the learning coefficient is $\alpha = 0.01$.

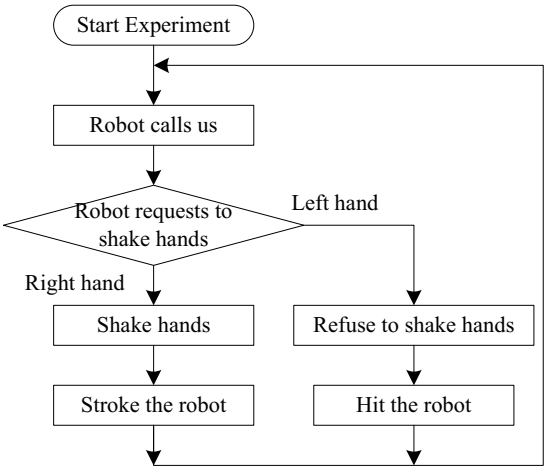


Fig. 2. Basic action patterns and setups

We obtained the robot's action probability as in Fig.3. We showed the probability of the handshake with the left hand using a vertical dotted line, and the probability of the right handshake using a vertical solid line. As a result, the action probability of the right handshake increased while that of the left handshake decreased as the experiment progressed.

Next, we counted the number of behaviors every 250 steps as shown in Fig.4 in order to confirm the behavior tendency. As for learning, the number of right handshakes exceeded the number of those made with the left hand during the first 250 steps. Then, the number of left handshakes increased little, and the robot did not shake hands with the left hand at the last. Thus we confirmed that the robot could autonomously change its behavior tendency according to its experiences.

In addition, the action entropy shown in Fig.5 proves action entropy of LA is convergent for the value of entropy has been reduced and finally reduced to zero.

The experimental results show that the robot is able to change its behavior autonomously through online interactions with the environment, and a learned action can be easily updated to meet the changing requirements.

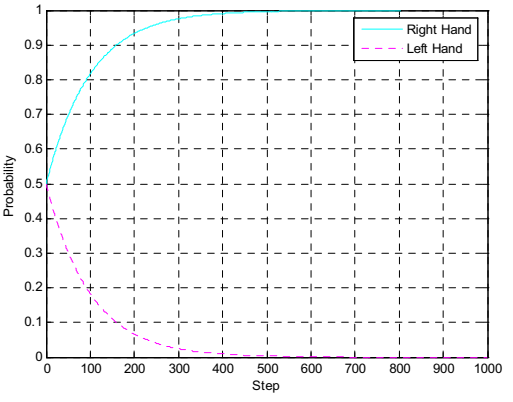


Fig. 3. Curve of action probability

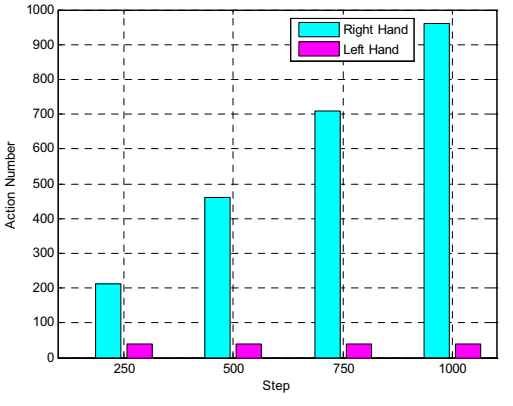


Fig. 4. Action Number

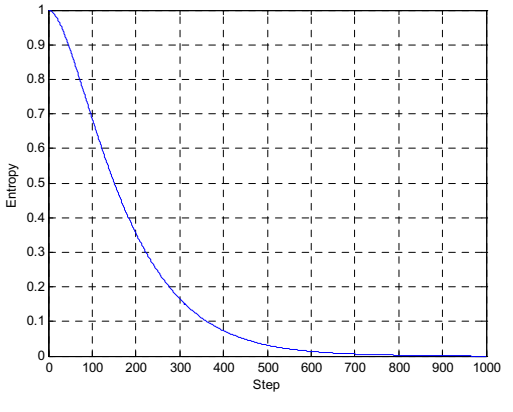


Fig. 5. Curve of action entropy

6 Conclusion

In this paper, we presented a developmental model for developmental robots based on learning automata, where the robot could select and output its behavior according to its action probability and tropism. We confirmed that the robot could autonomously change its behavior tendency according to its experiences through experimental evaluations implementing the model into a robot. In the future, we will apply the model into robots to realize autonomous exploration.

Acknowledgments. This work is supported by NSFC (60774077), 863 programs of China (2007AA04Z226), BMEC and BNSF (KZ200810005002), BNSF (4102011) and NSFC (61075110).

References

1. Raphael, B.: The robot 'Shakey' and 'his' successors. Computers and People 25, 7–21 (1976)
2. Brooks, R.A.: From earwigs to humans. Robotics and Autonomous Systems 20, 291–304 (1997)
3. Chatila, R.: Robot mapping: An introduction. Springer Tracts in Advanced Robotics, Robotics and Cognitive Approaches to Spatial Mapping, vol. 38, pp. 9–12 (2008)
4. Yu, H.-l., Zhu, C.-m., Liu, H.-b., Gu, G.-c., Shen, J.: A survey on developmental robotics. CAAI Transaction on Intelligent System 2(4), 34–39 (2007) (in chinese)
5. Weng, J.: Developmental robotics: theory and experiments. International Journal of Humanoid Robotics 1(2), 199–236 (2004)
6. Tan, K.C., Chen, Y.J., Tan, K.K.: Task-oriented developmental learning for humanoid robots. IEEE Trans. on Industry Electronics 52(3), 906–914 (2005)
7. Blank, D., Kumar, D., Meeden, L.: Bringing up robot: fundamental mechanisms for creating a self-motivated, self-organizing architecture. Cybernetics and Systerns 36(2), 125–150 (2005)
8. Stojanov, G.: Petitag: A case study in developmental robotics. In: The First International Workshop on Epigenetic Robotics, IEEE Press, Lund (2001)
9. Itoh, K., Miwa, H.: Behavior model of humanoid robot based on operant conditioning. In: 5th IEEE-RAS International Conference on Humanoid Robots, pp. 220–225. IEEE Press, Los Alamitos (2005)
10. Ruan, X., Cai, J.: Skinner-Pigeon experiment based on probabilistic automata. In: 2009 Global Congress on Intelligent System, pp. 578–581 (2009)
11. Demetriou, A.: Cognitive development. In: Demetriou, A., Doise, W., van Lieshout, K.F.M. (eds.) Life-Span Developmental Psycholog, pp. 179–269 (1998)

Meddler, Agents in the Bounded Confidence Model on Flocking Movement World

Shusong Li, Shiyong Zhang, and Binglin Dou

Department of Computing & Information Technology,
Fudan University,
ShangHai, P.R. China
li_shusong@126.com, szhang@fudan.edu.cn, binglin.dou@gmail.com

Abstract. We present and analyze a model of Opinion Dynamics and Bounded Confidence on the Flocking movement world. There are two mechanisms for interaction. The theorem of 'Flocking' limits the agent's movement around the world and 'Bounded Confidence' chooses the agents to exchange the opinion. We introduce some special agents with different character into the system and simulate the opinion formation process using the proposed model. The results show the special agent can change the dynamics of system with small population. The infector shortens convergence time; the extremist leads to asymmetry polarization or deflection consensus; the leader change dynamics of system from consensus to polarization; and the meddler make sure that the final state becomes asymmetry polarization.

Keywords: Multi-agent system, Opinion Dynamics, Flocking, Social Network, Computer Simulation.

1 Introduction

We are interested in modeling, simulating and analyzing the opinion formation dynamics of social scenarios where individuals can benefit from pervasive sources of information [1]. Such scenarios include: public opinion on Internet; advertisement in Electronic Commerce; damage spreading in information system, etc. The study of such processes is of great importance, providing advice to risk assessment, management and disaster recovery in consensus emergency and large-scale technological incidents.

The *Bounded Confidence* model is one of the most famous opinion formation dynamics models. It is based on continuous opinions which agent's opinion adjustments only proceed when opinion difference is below a given threshold. An early formulation of such a model was given by Deffuant in 2000 [2]. Another source of opinion dynamics is the work by Hegselmann and Krause in 2002 [3]. The two models differ in their proposed communication structure but lead to opinion clustering in a similar way. Both of models show one of three types of result: consensus, polarization and fragmentation [4]. In recent years, a lot of interest has in the fields of formation of public and the opinion dynamics in the social network. It is obvious that the

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 591–600, 2011.

"classical" BC (*Bounded Confidence*) model is not efficient for social systems, so there are some well-suited models in recent study. Most extension of these models is based on the communication structure with different network topology. The influence of this structure, especially of small-world networks, on such models has been studied [5]. We find the knowledge of *Complex Network* is helpful to improve the BC model, but most of models extend the communication structure in the static space. We want reform the BC model in a dynamic space and combine the BC model with *Flocking* movement by the Multi-Agent System.

Flocking is a form of collective behavior of large number of interacting agents with a common group objective [6]. We can find this kind of phenomena in many systems, such as birds, fish, penguins, ant colonies, bees and crowds [7]. In nature, flocks are examples of self-organized networks of mobile agents capable of coordinated group behavior. The flocking characters include decentralization , adaptability, self-organization and robustness [8]. In 1986, Reynolds introduced three heuristic rules that led to creation of the first computer animation of flocking[9].

1) *Flock Centering: attempt to stay close to nearby flockmates;*
2) *Collision Avoidance: avoid collisions with nearby flockmates;*
3) *Velocity Matching: attempt to match velocity with nearby flockmates.*

The model presented in this paper is an extension of the BC model in a dynamic space. In the model, we add one variable to the agent, "*eyeshot*", which limits the set of neighbors that the agents can interaction. And there is another mechanism to system, "*Flocking*", which limits the agent's movement. The purpose of this paper is to present results about continuous opinion dynamics when agents have different character in the system with same reaction mechanism. We simulate and analyses the model with different type agents, especially the "extremism" agent (individuals with a very low uncertainty and an opinion located at the extremes of the initial opinion distribution)[5]. We then discuss the observed behavior of our model.

2 The BC Models on the Flocking Movement World

2.1 Bounded Confidence Algorithm with Local Communication

There are a population of N agents i with continuous opinions x_i in a two-dimensionality space. Each agent has an *eyeshot* value r_i and a confidence function $w_i(j,t)$. The number ε is called a *Bounded Confidence*.

When each agent moves in the space, it only can influence the agents in its eyeshot area. It means that the communication of the agents is local and limited by the *eyeshot*. We assume that any pair of agent i and agent j has a distance function $d(i,j,t)$. It is the value of distance between the agents in the space at time t. The set of neighbors for agent i is defined by

$$Neighbors_i(j,t) = \{ j \mid d(i,j,t) \le r_j, 1 \le j \le n, j \ne i \} \tag{1}$$

When the process of opinion adjustment begins, agent i communicates with every agent in set of $Neighbors_i(j,t)$. We collect the $w_i(j,t)$ of agent j which is the member of $Calculate_i(j,t)$.

$$Calculate_i(j,t) = \{ j \parallel x_i - x_j \mid < \varepsilon_i, 1 \le j \le m, j \ne i \} \tag{2}$$

We suppose that the agent i has several neighbors, and k is the size of $Calculate_i(j,t)$. The confidence function can be described by:

$$\sum_{j=1}^{k} w_i(j,t) + w_i(i,t) = 1 \tag{3}$$

Agent i's opinion x_i changes at time $t+1$ by weighting each received opinion at time t with the confidence in the corresponding source. Opinions are adjusted according to:

$$x_i(t+1) = w_i(i,t)x_i(t) + \sum_{j=1}^{k} w_i(j,t)x_j(t) \tag{4}$$

When the process of opinion adjustment ends, the agents choose a direction according to *Flocking* algorithm and continue their movement.

2.2 Flocking Movement

Each agent has a repulsive value r_radius_i and a gravitation value g_radius_i. When agents move in the space, they can influence the agents in its gravitation radius. The set of flocking group for agent i is defined by:

$$flocking_i(j,t) = \{ j \mid d(i,j,t) \le g\_radius_i, 1 \le j \le n, j \ne i \} \tag{5}$$

Agent i has a value of $heading_i(t)$ which is its movement direction at time t. When the opinion adjustment ends, agent chooses the neighbor h which the $d(i,h,t)$ is minimum. If the $d(i,h,t)$ is between the r_radius_i and g_radius_i, the agent i will change the direction according to formula (6)-(7).

$$direction_a(t) = \sum_{j=1}^{k} heading_j(t)/k - heading_i(t) \quad j \in flocking_i(j,t) \tag{6}$$

The formula makes sure that the speed of agents in a flocking group is alignment in the end. It is speed consensus algorithms.

$$direction_c(t) = \sum_{j=1}^{k} (heading_j(t) - heading_i(t))/k - heading_i(t) \quad j \in flocking_i(j,t) \tag{7}$$

The $direction_c(t)$ achieves the rule of cohesion. According to this formula, the distance between agents in the flocking group will come to steady.

If the $d(i,h,t)$ is smaller than the r_radius_i, the agent i will turn to a new direction by formula (8). It makes sure that the agents will not be collisions with neighbors in the same flocking group.

$$direction_s(t) = heading_i(t) - heading_h(t) \tag{8}$$

When we achieve the value of *direction(t)*, the agents will turn to a new direction according to the algorithms. Every agent adjusts its direction in the same time, moves one step and waits to adjust the opinions.

3 Simulation and Analysis

In this section we will study the model by means of simulations. There are hundreds of agents moving with flocking track on a square lattice of size 71*71[10]. The world is closed and edge connected. At each time step the agents readjust their opinion and direction according to the formulas. There are four essential parameters in the simulations: *Bounded Confidence*, *eyeshot*, *gravitation radius* and *repulsive radius*. We research the influence of these parameters in the process of opinion exchange and acquire some result of this mode.

3.1 Agent with Same Characters

At the beginning, we assume that the entire agents in the system are ordinary with the same characters and we achieve some conclusion by the analysis of the results in our previous studies.

First, the *Bounded Confidence* is the most important element that influences the final state of model. With the *bounded confidence* increasing, the dynamics of system will change from fragmentation to polarization, and become consensus in the end. Second, the *eyeshot* of agents does not influence the dynamics of system, but the convergence time will shorten when the area of *eyeshot* expands. Third, the *gravitation radius* not only influences the movement of agents in the flocking but also changes the dynamics of opinions. When the *gravitation radius* increases, the final state of opinions may become fragmentation, and the convergence time of flocking will shorten. The last, the *repulsive radius* only influences the convergence time of flocking. The convergence time will be prolonged when the *repulsive radius* enhances.

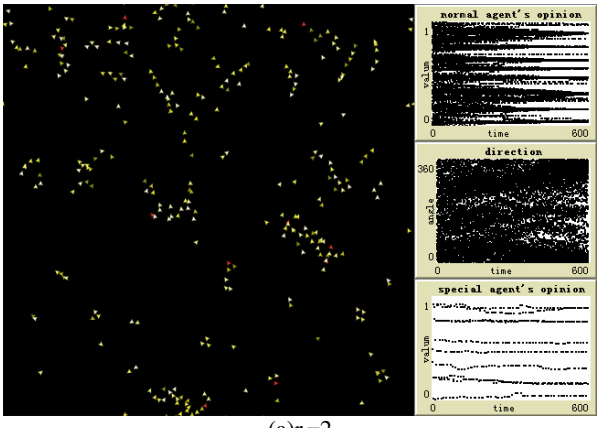
The above conclusion is based on the assumptions that agents have the same character in the system. But in practice the individual should have different attribute (e.g. ability, belief, power, etc). We want to know what the impact of the few special agents upon the dynamics of system, so the next section is focused on the assumptions of this: The different agents have different characters.

3.2 Agent with Different Characters

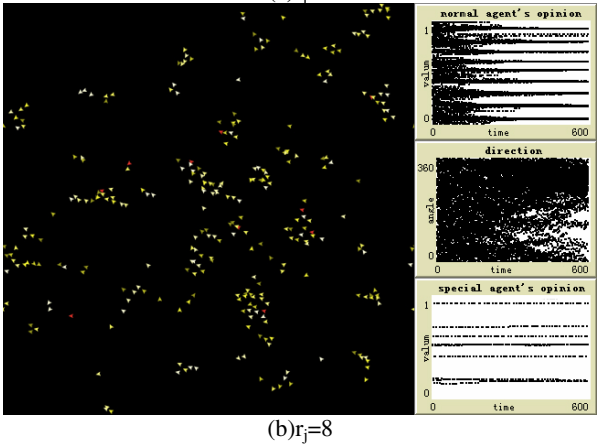
We now introduce some special agents with different character into our system. We suppose that these individuals have the larger ability, belief and power with small population.

3.2.1 Infector Agent

The different agents may have different *eyeshot*. It means that the influence between agents will be not symmetrical in the process of opinion adjustment. The reason might be the social sphere of individuals is different in the real world.



(a)r_j=2



(b)r_j=8

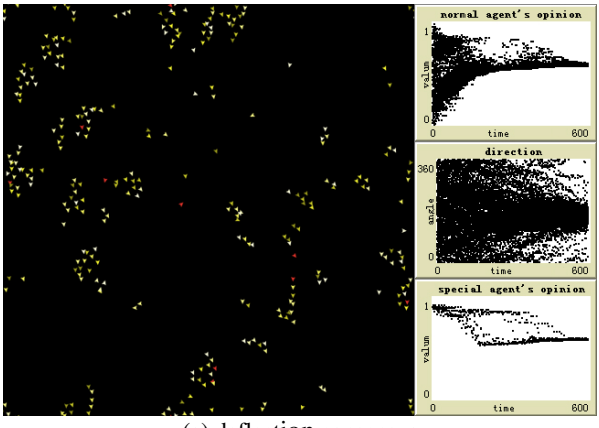
Fig. 1. The dynamics of opinions with different *eyeshot* of infector agent(300 agents, *bounded confidence* of ε_i=0.05, g_radius_i=2, r_radius_i=1, 600 steps)

There are some special agents that have larger *eyeshot* in this phase simulation. We call it *infector agent*. It has the uniform *bounded confidence* and *gravitation radius* as normal agents, and the population of it is small. With the simulation parameter N=300, ε_i=0.05, g_radius_i=2, r_radius_i=1, the normal agent's *eyeshot* r_i=2 and the infector's population n=10, we define r_j as the *eyeshot* of infector (Fig.1.).

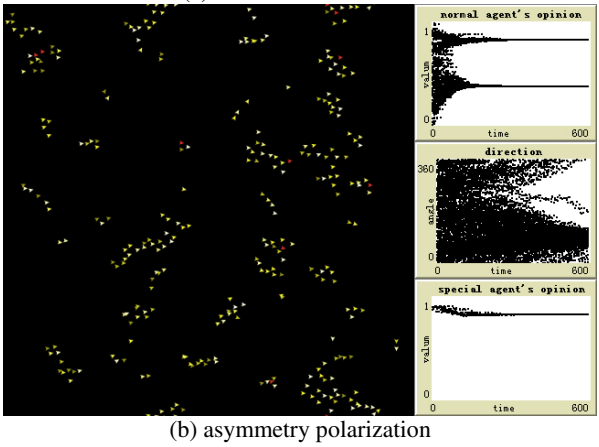
We can find that the change of convergence time is not distinct with exiguous infector. But the time will decrease when the eyeshot of infector increases.

3.2.2 Extremist Agent

Bounded confidence is the most powerful element that influences the opinions dynamics of system. In fact the agents' self-confidence is not same, so they should have different *bounded confidence*. The individual with strong belief will not change their mind easily and usually have a clear opinion. We assume that the extremist agents start with an extreme opinion, adhere to the opinion they already have and adjust their opinion according as a tiny *bounded confidence*.



(a) deflection consensus



(b) asymmetry polarization

Fig. 2. The dynamics of opinions with some extremist agents in the system (300 agents, *bounded confidence* of ε_i=0.3, ε_j=0.05, opinion of extremist $x_j(0)$=0.9, *eyeshot* r_i=2, g_radius_i=2, r_radius_i=1, 600 steps)

We introduce some extremist agents into the population. Their *eyeshot* and *gravitation radius* are not different with normal agents. The simulation results from Figure 2 are based on randomly generated start distribution of 300 agents with the same *eyeshot* r_i=2, g_radius_i=2, r_radius_i=1. The normal agent's *bounded confidence* ε_i=0.3, the extremist's *bounded confidence* ε_j=0.05, population n=10 and opinion $x_j(0)$=0.9. There are two types of convergence: deflection consensus and asymmetry polarization (Fig.2.).

We repeat the simulation 200 times, the state of asymmetry polarization happened 58 times. Most of the result is deflection consensus, and the survival opinion usually between the mean value of agents' initialization value and the extremist agents'. In the normal condition, agents' opinion should converge at the mean value of all agents'. But the system ends up in two distinct states when we introduce extremist agents. In the deflection consensus state, extremist's opinion deflects the initialization value and attracts the normal agent's opinions. The system's final opinion will

converge at one value. And in the asymmetry polarization situation, the dynamics of opinions should converge at polarizations. One pole of polarization is the extremist attracts the normal agents to cluster, and the other is the self-organizing by the remnant normal agents. Convergence time of two poles is same and the population of cluster is different. The opinion adjustment of extremist in the asymmetry polarization is thinner than in the deflection consensus.

By analysis of the result, we can conclude that the dynamics of opinion will be inscrutability if the other parameter (e.g. *eyeshot*, *gravitation radius*) keeps the value in the system. Because the dynamics of opinion is based on the situation (e.g. location, direction, opinion) of the agents at the beginning and the initial state of agents is stochastic.

3.2.3 Leader Agents

The agents may have different *gravitation radius*. It means that the power of influence in the movement will be not symmetrical. The reason might be the status of individuals is different in the real world.

There are some special agents that have larger *gravitation radius* in this phase simulation. We call it *leader agent*. It has the uniform *bounded confidence* and *eyeshot* as normal agents. With the simulation parameter N=300, *eyeshot* r_i=2, r_radius_i=1, the normal agent's g_radius_i=2 and the leader's population n=10, g_radius_i=6. We simulate the model with the different *bounded confidence* ε_i.

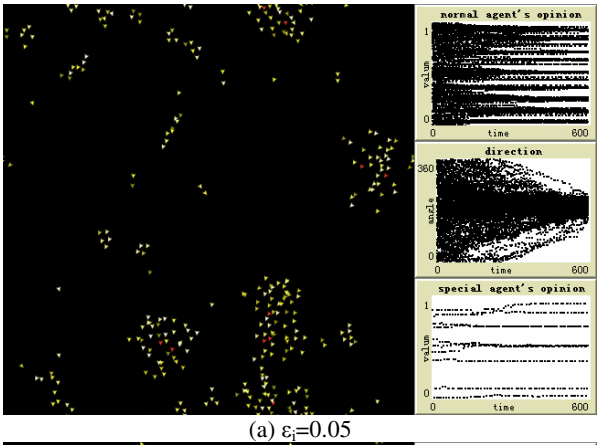
As is shown in the figure 3, the dynamics of normal agents' opinions conforms to leader's opinions. With the quite small population of leader agents the dynamics has notable change and the final state becomes polarization, when the *bounded confidence* is beyond the threshold of consensus.

By analysis the result we can find that a small quantity of leader agents can bring the normal agents into flocking movement in the short time. The leader attracts the agents to cluster with the same speed in the space. Among the clusters, the communication can't touch each other beyond the *eyeshot*. But the agents will adjust their mind very efficient in the same group. The leader's eyeshot is the same as normal agent that is why the dynamics of them are coincident. But the convergence time of flocking will be shorter than opinion adjustment when the leader exists, so the dynamics of system will become polarization even if the *bounded confidence* is beyond the threshold of consensus.

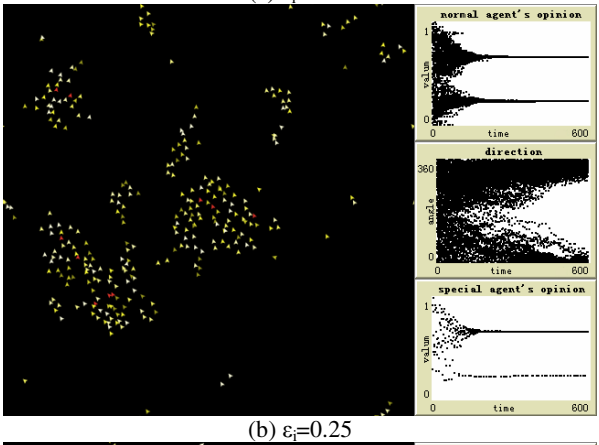
3.2.4 Meddler Agents

The agent with the characters of infector, leader and extremist we call it 'meddler' agent. It has the small *bounded confidence*, extreme opinion, large *eyeshot* and large *gravitation radius*, while it still adjusts its opinions by the uniform algorithm. The reason for such scenarios is for the individuals with great interpersonal relationships and powerful personal influence in the real world.

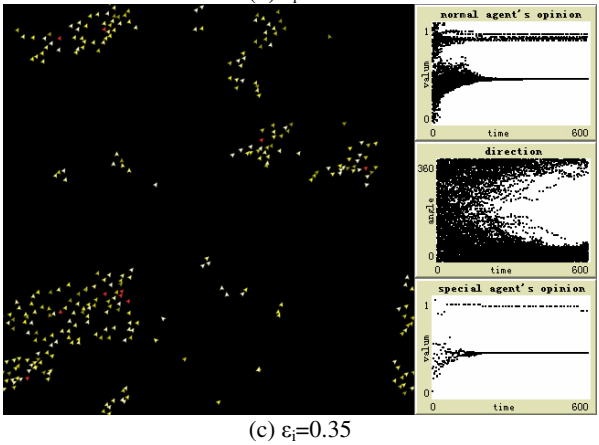
The final state of simulations will be asymmetry polarization or deflection consensus when the extremist agent exists. We conclude that the dynamics of opinion will be inscrutability if the other parameters (e.g. *eyeshot*, *gravitation radius*) keep the value in the system, but it can be forecasted when we increase the extremist agent's *eyeshot* and *gravitation radius*. We randomly generate a start distribution of 300 agents and introduce some meddler agents with different *eyeshot* and *gravitation radius* into our system. For each of these parameters we repeat the simulation 200 times. The statistic data is shown in the table 1.



(a) ε_i=0.05



(b) ε_i=0.25



(c) ε_i=0.35

Fig. 3. The dynamics of opinions with some leader agents in the system (300 agents, *eyeshot* r_i=2, r_radius_i=1, normal agent's g_radius_i=2, leader agent's g_$radius_i$=6, 600 steps)

Table 1. The count of asymmetry polarization happened in the 200 times(300 agents, normal agent's *bounded confidence* ε_i=0.3, *eyeshot* r_i=2, g_radius_i=2, r_radius_i=1)

Meddler's character	eyeshot r_i=2 g_radius=2	eyeshot r_i=4 g_radius=4	eyeshot r_i=6 g_radius=6
times of polarization	58/200	169/200	200/200

In this table we count the state of asymmetry polarization happened in the 200 times test. The meddler agent changes the probability of opinions dynamics occurred. When the radius of eyeshot and gravitation increases, the final state of system will be inclined to the asymmetry polarization. We find that the meddler attracts normal agents to cluster very quickly, and the other agents will cluster according to the algorithm by themselves. *Eyeshot* and *gravitation radius* is the key that influence the probability of asymmetry polarization occurred.

4 Conclusion and Future Work

The model presented in this paper is an extension of BC (*Bounded Confidence*) model in a dynamic space. We simulate and analyses the model with different type agents: infector, extremist, leader and meddler. The results show that the special agent can change the dynamics of system with small population.

The infector with large *eyeshot* can shorten the convergence time of opinion. The dynamics of system will be asymmetry polarization or deflection consensus when the extremist agent exists, and the state occurs in random. The leader with large *gravitation radius* can influence the final state of system, which will become polarization even if the *bounded confidence* is beyond the threshold of consensus. The meddler agent with all characters of three special agents can influence the probability of opinions dynamics occurred. The final state is one hundred percent asymmetry polarization when the *eyeshot* and *gravitation radius* increase enough.

This system is more complex and realistic than the classic BC model. We can use it to simulate the process of public opinion on the Internet and damage spreading in the information system, etc. there are still some issues we want to solve. How can we lead to a consensus that focuses on the topic we want or avoid a damage spreading by the means of interference? In future work, we will make a deep study on the influence of different type agents in this system.

References

1. Ramirez-Cano, D., Pitt, J.: Follow the leader: profiling agents in an opinion formation model of dynamic confidence and individual mind-sets. In: Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology, Hong Kong, China (2006)
2. Deffuant, G., Neau, D., Amblard, F., Weisbuch, G.: Mixing beliefs among interacting agents. Advances in Complex Systems 3(1-4) (2000)
3. Hegselmann, R., Krause, U.: Opinino Dynamics and Bounded Confidence models, analysis, and simulation. Journal of Artificial Societies and Social Simulation 5(3) (2002)

4. Mckeowwn, G., Sheehy, N.: Mass media and Polarisation processes in the bounded confidence model of opinion dynamics. Journal of Artificial Societies and Social Simulation 9(1) (2006)
5. Amblard, F., Deffuant, G.: The role of network topology on extremism propagation with relative agreement opinion dynamics. Physica A 343 (2004)
6. Olfati-Saber, R.: Flocking for multi-agent dynamic systems: algorithms and theory. IEEE Trans. Automat. Contr. 51 (2005)
7. Han, J.: Soft control on collective behavior of a group of autonomous agents by a shill agent. Jrl. Syst. Sci. & Complexity 19 (2006)
8. Gazi, V., Passino, K.M.: Stability Analysis of Social Foraging Swarms. IEEE Transactions on Systems, Man and Cybernetics 34 (2004)
9. Reynolds, C.W.: Flocks, herds, and schools: A distributed behavioral model. Computer Graphics 21 (1987)
10. Netlogo [EB/OL] (2007), `http://ccl.northwestern.edu/netlogo`
11. Fortunato, S.: Damage Spreading and Opinoin Dynamics on Scale Free Networks. Physica A 384 (2004)

Statistical Optimal Control Using Neural Networks

Bei Kang and Chang-Hee Won*

College of Engineering, CSNAP Laboratory, 1947 N. 12th Street,
Philadelphia, PA 19122, USA
cwon@temple.edu

Abstract. In this paper, we investigated statistical control problems using *n*-th order cost cumulants. The *n*-th order statistical control is formulated and solved using a Hamilton-Jacobi-Bellman (HJB) partial differential equation. Both necessary and sufficient conditions for *n*-th cumulant statistical control are derived. Statistical control introduces an extra degree of freedom to improve the performance. Then, the neural network approximation method is applied to solve the HJB equation numerically. This gives a statistical optimal controller. We apply statistical optimal control to a satellite attitude control application. This illustrates that neural network is useful in solving an *n*-th cumulant HJB equation, and the statistical controller improves the system performance.

Keywords: statistical control, neural networks, cost cumulants, stochastic optimization.

1 Introduction

The statistical optimal control method optimizes cumulants of the cost function of a stochastic system. For a stochastic system, the cost function is a random function because the states are random. This cost function is characterized by cumulants. For example, the first cumulant is the mean of the cost function. The second cumulant is the variance, which is a measure that tells how much the data deviate from the mean. The third cumulant, skewness, measures the departure from symmetry of the cost distribution. In statistical control, we shape the distribution of the cost function by optimizing cumulants, which will further affect the system performance.

Statistical control is closely related to adaptive dynamic programming (ADP). In typical ADP, we find the controller such that the expected value (first cumulant) of the utility function is optimized [1]. This idea can be generalized to the optimization of the higher order cumulants of the utility functions. This is the concept behind statistical optimal control [2, 3].

A statistical optimization problem is formulated using a Hamilton-Jacobi-Bellman (HJB) equation [4]. The *n*-th cost cumulant HJB equation generation procedure has been presented in [5]. However, general necessary and sufficient conditions for *n*-cost cumulant statistical control have not been published. We present HJB equations for the *n*-th order cost cumulant statistical control. The second cost cumulant case

* This work is supported in part by the National Science Foundation grant, ECCS-0969430.

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 601–610, 2011.

was presented in [6]. Generalizing this to *n*-th cost cumulant is the main results of this paper.

HJB equations are notoriously difficult to solve both numerically and analytically, except for a few special cases such as linear-quadratic-Gaussian case. For general nonlinear systems and higher-order cumulant problems, there is no straight forward way to solve the HJB equation. Even for a linear system, the higher order cumulant problem will yield a nonlinear controller [5]. For general stochastic systems, previous work in [4, 7] solves the HJB equation for higher cumulant of the linear system. For a nonlinear stochastic system, however, there is no effective way to solve the corresponding HJB equation. In deterministic systems, a neural network method is used to approximate the value function of the HJB equation [8]. The neural network approximation method is similar to power series expansion method which was introduced by Al'brekht in [9] to approximate the value function in infinite-time horizon. The approximated value function and the optimal controller are determined by finding coefficients of the series expansion. Another application of using neural network to solve HJB equation was developed by P. V. Medagam *et al.*, who proposed a radial basis function neural network method for the output feedback system [10]. All the above works, however, deal with deterministic systems and not stochastic systems. In this paper, we develop a neural network method for the stochastic systems. We solve HJB equations for arbitrary order cost cumulant minimization problems.

In the next section, we formulate the statistical control problem using HJB equations. Then we will discuss the solution of the HJB equation using neural network methods in Section 3. The examples are given in Section 4. Section 5 concludes the paper and proposes the future work.

2 Problem Formulation and HJB Equation

The stochastic system that we study has the following dynamics,

$$dx(t) = f\left(t, x(t), u(t)\right)dt + \sigma(t, x(t))dw(t), \tag{1}$$

where $t \in T = [t_0,\ t_f]$, $x(t) \in \mathbb{R}^n$ is a random state variable which is independent of $w(t)$, $x(t_0) = x_0$, and $w(t)$ is a d-dimensional Gaussian random process with zero mean and covariance of $W(t)dt$. The control action is defined as $u(t) = k\left(t, x(t)\right)$.

The cost function associated with (1) is given by the following form,

$$J(t, x(t); k) = \psi(x(t_f)) + \int_{t_0}^{t_f} L\left(s, x(s), k\left(s, x(s)\right)\right)ds. \tag{2}$$

where $\psi(x(t_f))$ is the terminal cost, L is a positive definite function which is continuously differentiable and satisfies polynomial growth condition. Let $Q_0 = [t_0,\ t_f] \times \mathbb{R}^n$ and $\bar{Q}_0$ be its closure. We also consider an open set $Q \subset Q_0$. The feedback control law $k\left(t, x(t)\right)$ and f, σ are assumed to satisfy the Lipschitz

condition and the linear growth condition. We want to find an optimal controller for system (1), which minimizes the *n*-th cumulant of the cost function (2).

We introduce a backward evolution operator $O_{(k)}$, which is defined as

$$O_{(k)} = \frac{\partial}{\partial t} + \left\langle f\left(t,x,k(t,x)\right), \frac{\partial}{\partial x} \right\rangle + \frac{1}{2} tr\left(\sigma(t,x) W \sigma(t,x)' \frac{\partial^2}{\partial x^2} \right),$$

where $\left\langle f\left(t,x,k(t,x)\right), \frac{\partial}{\partial x} \right\rangle = \sum_{i=1}^{n} f_i(t,x,k(t,x)) \frac{\partial}{\partial x_i}$,

$$tr\left(\sigma(t,x) W \sigma(t,x)' \frac{\partial^2}{\partial x^2} \right) = \sum_{i,j=1}^{n} \left(\sigma(t,x) W \sigma(t,x)' \right)_{i,j} \frac{\partial^2}{\partial x_i \partial x_j}.$$

The necessary and sufficient conditions for the *n*-th cost cumulant minimization problem are analyzed using HJB equations. The results are given in the following theorems.

Theorem 1. (*n-th cumulant HJB equation*) Let $V_1, V_2, \ldots, V_{n-1} \in C_p^{1,2}(Q) \bigcap C(\overline{Q})$ be admissible cumulant cost functions. Assume the existence of an optimal control law $k = k^*_{V_k \mid M} \in K_M$ and an optimum value function $V_n^*(t,x) \in C_p^{1,2}(Q) \bigcap C(\overline{Q})$. Then the minimal *n*-th cumulant cost function $V_n^*(t,x)$ satisfies the following HJB equation.

$$0 = \min_{k \in K_M} \left\{ O_{(k)}\left[V_n^*(t,x)\right] + \frac{1}{2} \sum_{s=1}^{n-1} \frac{n!}{s!(n-s)!} \left(\frac{\partial V_s(t,x)}{\partial x} \right)' \sigma(t,x) W \sigma(t,x)' \left(\frac{\partial V_{n-s}(t,x)}{\partial x} \right) \right\}.$$

for $(t,x) \in \overline{Q}_0$, with the terminal condition $V_n^*(t_F,x) = 0$.

Proof. Omitted for brevity. See [10].

Theorem 2. (*verification theorem*) Let $V_1(t,x), V_2(t,x), .., V_{n-1}(t,x) \in C_p^{1,2}(\overline{Q}) \bigcap C(\overline{Q})$ be an admissible cumulant cost function. Let $V_n^*(t,x) \in C_p^{1,2}(\overline{Q}) \bigcap C(\overline{Q})$ be a solution to the partial differential equation

$$0 = \min_{k \in K_M} \left\{ O_{(k)}\left[V_n^*(t,x)\right] + \frac{1}{2} \sum_{s=1}^{n-1} \frac{n!}{s!(n-s)!} \left(\frac{\partial V_s(t,x)}{\partial x} \right)' \sigma(t,x) W \sigma(t,x)' \left(\frac{\partial V_{n-s}(t,x)}{\partial x} \right) \right\}.$$

with zero terminal condition, then, $V_n^*(t,x)$ is less than or equal to the *n*-th cumulant of the cost $J\left(t,x,k(t,x)\right)$ for all $k \in K_M$ and $(t,x) \in Q$. If in addition, there is a k^* satisfies the following equation,

$$V_n(t,x,k^*) = \min_{k \in K_M} \left\{ V_n(t,x,k) \right\},$$

then $V_n^*(t,x) = V_n(t,x,k^*)$, which equals to the minimal n-th cumulant of the cost $J\left(t,x,k(t,x)\right)$ and k^* is the optimal controller.

Proof. Omitted for brevity. See [11].

3 Solution of HJB Equation Using Neural Networks

HJB equation (3) is difficult to solve directly, especially for the nonlinear systems. In this section, we use a neural network method to approximate the value functions in (3), and apply neural network method to find the solution of the HJB equations.

A number of neural network input functions, $\delta_i(x)$, which are state dependent, are multiplied by the corresponding weights, $w_i(t)$ which are time dependent, and are summed up to produce an output function, $V_L(x,t)$. This output function will be the approximated value function of the corresponding HJB equation. Sandberg proved that the uniform continuous function can be approximated by the summation of some basis functions multiplied by the time-varying weights [12, 13].

In this paper, we assume that the value functions of the HJB equation are uniformly continuous and therefore can be approximated by the combination of basis functions. And the neural network functions are used as the basis functions. In this chapter, we use a polynomial series expansion $\overline{\delta}_L(x) = \left\{ \delta_1(x), \delta_2(x), \cdots, \delta_L(x) \right\}'$ as the neural network input functions. By determining the time dependent weights of the series expansion represented by the vector $\overline{w}_L(t) = \{ w_1(t), w_2(t), ..., w_L(t) \}'$, we find the output functions, which is the approximation to the value function $V_{nL}^*(t,x) = \overline{w}_L'(t)\overline{\delta}_L(x) = \sum_{i=1}^{L} w_i(t)\delta_i(x)$. Here the short bars above the $w_L(t)$ and $\delta_L(x)$ stress the fact that $w_L(t)$ and $\delta_L(x)$ are vectors. To keep the notation simple, we omit the bar above the vectors *x*, *u*, and *dw*. We use notation $V_{nL}^*(t,x)$ instead of $V_n^*(t,x)$ to emphasize the difference between the approximated value function and the original value function.

Because the polynomial series expansion is pre-defined and known, the only unknown in the approximated HJB equation is the weight or coefficients of the series expansion. These weights are time dependent. Thus, solving the HJB partial differential equation problem becomes finding the weight of the series expansion problem.

From [11], the optimal controller k^* has the following form.

$$k^* = -\frac{1}{2} R^{-1} B' \left(\frac{\partial V_1}{\partial x} + \gamma_2 \frac{\partial V_2}{\partial x} + \gamma_3 \frac{\partial V_3}{\partial x} + ... + \gamma_{n-1} \frac{\partial V_{n-1}}{\partial x} + \gamma_n \frac{\partial V_n^*}{\partial x} \right).$$

We substitute k^* back into the HJB equation from the first to n-th cumulant. Then, we use neural network series expansion to approximate each HJB equation. Similar to the previous section, we assume that the terminal conditions for $V_1, V_2, ..., V_n$ are zero. To distinguish the weights for the different value functions, we define the weights for $V_{iL}(x,t)$ as $\overline{w}_{iL}(t)$. For example, we use $V_{1L}(x,t) = \overline{w}_{1L}'(t)\overline{\delta}_{1L}(x)$ to approximate $V_1(x,t)$, $V_{2L}(x,t) = \overline{w}_{2L}'(t)\overline{\delta}_{2L}(x)$ to approximate $V_2(x,t), ...,$ and $V_{nL}(x,t) = \overline{w}_{nL}'(t)\overline{\delta}_{nL}(x)$ to approximate $V_n^*(x,t)$. Then, we use the neural network approximations and substitute

$V_{1L}(x,t) = \overline{w}_{1L}'(t)\overline{\delta}_{1L}(x)$, $V_{2L}(x,t) = \overline{w}_{2L}'(t)\overline{\delta}_{2L}(x)$, ..., $V_{nL}(x,t) = \overline{w}_{nL}'(t)\overline{\delta}_{nL}(x)$ and apply the method of weighted residual [10]. Then we obtain

$$\begin{aligned}
\dot{\overline{w}}_{1L}(t) = & -\left\langle \overline{\delta}_{1L}(x), \overline{\delta}_{1L}(x)\right\rangle_\Omega^{-1} \left\langle \nabla \overline{\delta}_{1L}(x) g(x), \overline{\delta}_{1L}(x)\right\rangle_\Omega \overline{w}_{1L}(t) \\
& + \frac{1}{4}\left\langle \overline{\delta}_{1L}(x), \overline{\delta}_{1L}(x)\right\rangle_\Omega^{-1} A_L \cdot \overline{w}_{1L}(t) - \sum_{i=2}^{n} \frac{\gamma_i^2}{4}\left\langle \overline{\delta}_{1L}(x), \overline{\delta}_{1L}(x)\right\rangle_\Omega^{-1} B_{iL} \cdot \overline{w}_{iL}(t) \\
& - \left\langle \overline{\delta}_{1L}(x), \overline{\delta}_{1L}(x)\right\rangle_\Omega^{-1} \left\langle l, \overline{\delta}_{1L}(x)\right\rangle_\Omega - \frac{1}{2}\left\langle \overline{\delta}_{1L}(x), \overline{\delta}_{1L}(x)\right\rangle_\Omega^{-1} C_L \\
& - \sum_{i=2, j=2; i\neq j}^{n} \frac{\gamma_i \gamma_j}{2}\left\langle \overline{\delta}_{1L}(x), \overline{\delta}_{1L}(x)\right\rangle_\Omega^{-1} B_{iL} \cdot \overline{w}_{jL}(t),
\end{aligned} \tag{4}$$

$$\begin{aligned}
\dot{\overline{w}}_{2L}(t) = & -\left\langle \overline{\delta}_{2L}(x), \overline{\delta}_{2L}(x)\right\rangle_\Omega^{-1} \left\langle \nabla \overline{\delta}_{2L}(x) g(x), \overline{\delta}_{2L}(x)\right\rangle_\Omega \overline{w}_{2L}(t) \\
& + \frac{1}{2}\left\langle \overline{\delta}_{2L}(x), \overline{\delta}_{2L}(x)\right\rangle_\Omega^{-1} A_L \cdot \overline{w}_{2L}(t) + \sum_{i=2}^{n} \frac{\gamma_i}{2}\left\langle \overline{\delta}_{2L}(x), \overline{\delta}_{2L}(x)\right\rangle_\Omega^{-1} B_{iL} \cdot \overline{w}_{2L}(t) \\
& - \frac{1}{2}\left\langle \overline{\delta}_{2L}(x), \overline{\delta}_{2L}(x)\right\rangle_\Omega^{-1} D_L - \left\langle \overline{\delta}_{2L}(x), \overline{\delta}_{2L}(x)\right\rangle_\Omega^{-1} E_L \cdot \overline{w}_{1L}(t),
\end{aligned} \tag{5}$$

$$\vdots$$

$$\begin{aligned}
\dot{\overline{w}}_{nL}(t) = & -\left\langle \overline{\delta}_{nL}(x), \overline{\delta}_{nL}(x)\right\rangle_\Omega^{-1} \left\langle \nabla \overline{\delta}_{nL}(x) g(x), \overline{\delta}_{nL}(x)\right\rangle_\Omega \overline{w}_{nL} \\
& + \frac{1}{2}\left\langle \overline{\delta}_{nL}(x), \overline{\delta}_{nL}(x)\right\rangle_\Omega^{-1} A_L \cdot \overline{w}_{nL} + \sum_{i=2}^{n} \frac{\gamma_i}{2}\left\langle \overline{\delta}_{nL}(x), \overline{\delta}_{nL}(x)\right\rangle_\Omega^{-1} B_{iL} \cdot \overline{w}_{nL} \\
& - \frac{1}{2}\sum_{i=1}^{n-1} \frac{n!}{i!(n-i)!}\left\langle \overline{\delta}_{nL}(x), \overline{\delta}_{nL}(x)\right\rangle_\Omega^{-1} F_{iL} \cdot \overline{w}_{(n-i)L} - \frac{1}{2}\left\langle \overline{\delta}_{nL}(x), \overline{\delta}_{nL}(x)\right\rangle_\Omega^{-1} G_L \cdot \overline{w}_{nL}.
\end{aligned} \tag{6}$$

where quantities A_L, B_{iL}, C_L, D_L, E_L, F_{iL}, G_L, are defined as follows,

$$A_L = \sum_{s=1}^{L} w_{1s}(t) \left\langle \nabla \overline{\delta}_{1L}(x) B(t,x) R^{-1} B'(t,x) \nabla \delta_{1s}(x), \overline{\delta}_{1L}(x) \right\rangle_{\Omega},$$

$$B_{iL} = \sum_{s=1}^{L} w_{is}(t) \left\langle \nabla \overline{\delta}_{iL}(x) B(t,x) R^{-1} B'(t,x) \nabla \delta_{is}(x), \overline{\delta}_{1L}(x) \right\rangle_{\Omega},$$

$$C_L = \left\langle tr\left(\sigma W \sigma' \nabla\left(\nabla \overline{\delta}_{1L}'(x) \overline{w}_{1L}(t)\right)\right), \overline{\delta}_{1L}(x) \right\rangle_{\Omega},$$

$$D_L = \left\langle tr\left(\sigma W \sigma' \nabla\left(\nabla \overline{\delta}_{2L}'(x) \overline{w}_{2L}(t)\right)\right), \overline{\delta}_{2L}(x) \right\rangle_{\Omega},$$

$$E_L = \sum_{s=1}^{L} w_{1s}(t) \left\langle \nabla \overline{\delta}_{1L}(x) \sigma W \sigma' \nabla \delta_{1s}(x), \overline{\delta}_{2L}(x) \right\rangle_{\Omega},$$

$$F_{iL} = \sum_{s=1}^{L} w_{is}(t) \left\langle \nabla \overline{\delta}_{(n-i)L}(x) \sigma W \sigma' \nabla \delta_{is}(x), \overline{\delta}_{nL}(x) \right\rangle_{\Omega},$$

$$G_L = \left\langle tr\left(\sigma W \sigma' \nabla\left(\nabla \overline{\delta}_{nL}'(x) \overline{w}_{nL}(t)\right)\right), \overline{\delta}_{nL}(x) \right\rangle_{\Omega}.$$

Therefore, by knowing the terminal condition of the above ordinary differential equations, we solve the *n*-th cumulant neural network approximations along with other cumulant constraint equations.

Remark. In order to solve the *n*-th cumulant neural network equation, we need to solve all *n*-th cumulant neural network equations simultaneously. By using neural network method, we convert the partial differential HJB equations into the neural network ordinary differential equations of (4) to (6). Then, we solve these approximated ordinary differential equations for $\overline{w}_{1L}(t)$ to $\overline{w}_{nL}(t)$ and determine the corresponding Lagrange multipliers γ_2 to γ_n. We will show the applications of the neural network method in the next section.

4 Satellite Attitude Control Applications

In this example, we study the statistical control for a linearized satellite attitude control model. The satellite attitude control system that we consider is for a low earth orbit satellite, KOMPSAT [14, 15]. The satellite attitude is controlled by four reaction wheels and four thrusters. We have the following system dynamics,

$$\dot{\bar{x}} = Ax + Bu + E\xi,$$

where matrices *A, B,* and *E* are defined in [14], ξ is a Brownian motion with variance matrix $W = 0.01 I_{3\times 3}$. We assume the terminal cost is zero. The cost function is given in the following quadratic form,

$$J(t, x(t); k) = \int_{t_0}^{t_f} \left[x'(s) Q x(s) + k'(s, x(s)) R k(s, x(s)) \right] ds.$$

Here, we simulate the three cost cumulant control. Because we use Lagrange multiplier method to derive the second and third cost cumulant HJB equations, we

solve the HJB equations by assigning different values for the Lagrange multipliers γ_2 and γ_3. We compare the system performance with the minimal first, second, and third cumulant statistical controls. The results are presented in the following figures.

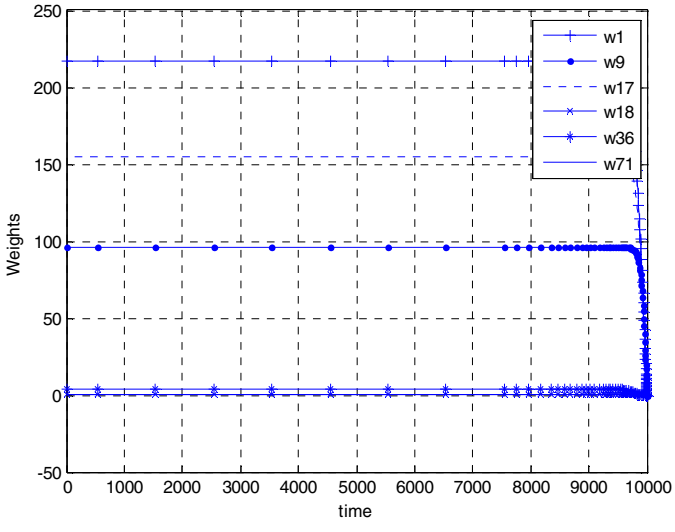


Fig. 1. Neural network weights evolution with respect to time

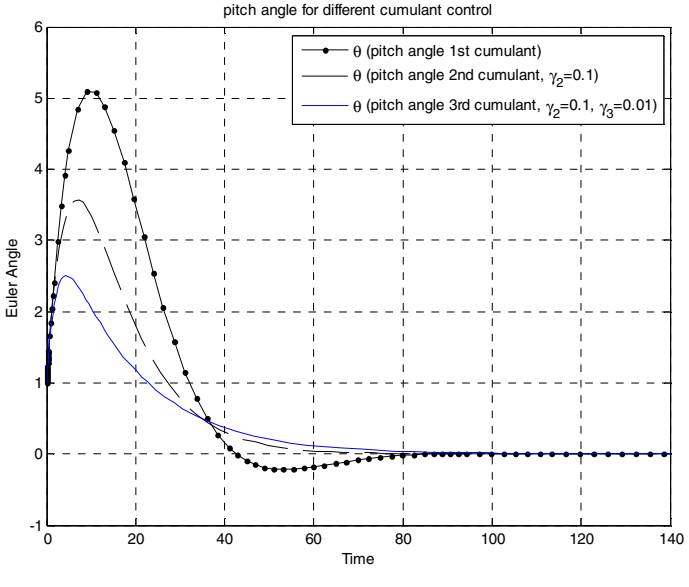


Fig. 2. Pitch angle θ for the first, second and third cumulant statistical control

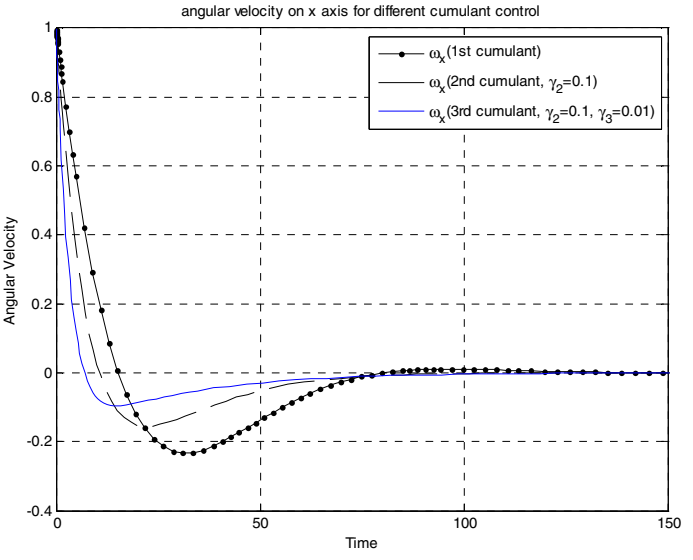


Fig. 3. Angular velocity of *y* axis for the first, second, and third cumulant statistical control

Fig. 1 shows the neural network weights trajectories when we calculate the backward integration. There are 168 weights used, out of which six are shown in the figure, in this example to yield an accurate neural network approximation for the first three cost cumulants HJB equations. Because most of the weights approach zero in the equilibrium state, we only show several representative weights which ended up with nonzero values. From Fig. 2, we note that the pitch angle trajectory for the third cumulant statistical control performs better than the second cumulant statistical control from the overshoot and settling time point of view. Moreover, the second cumulant statistical control has better performance than the first cumulant statistical control. For the angular velocity in Fig. 3, the trajectory of the third cumulant statistical control has less undershoot and smaller settling time than the trajectory of the second cumulant statistical control, which has less undershoot and smaller settling time than the first cumulant trajectory. Thus, the third cumulant statistical control has the best performance. However, in Fig. 4, the speed of the reaction wheel 1 under the third cumulant statistical control is much larger than the second and first cumulant statistical control. Moreover, the third cumulant statistical control settles the trajectory faster than the second and first cumulant statistical control. Therefore, it is shown that the third cumulant statistical control has better performance, i.e. settling time and angular velocity, but it requires much more actuation (reaction wheel speed) at the initial stage (reaction wheel speed).

Remark. Although the simulation results in this section show that the higher order cumulant statistical control generates better state trajectories than the lower cumulant statistical control, this is not always the case. The theory, however, shows that by applying different cumulant statistical control, we obtain different system performances. Even for the same order cumulant statistical control, we obtain

different results by assigning different values to the Lagrange multipliers. These properties demonstrate that we added an extra design freedom to the traditional mean cost control (minimal first cumulant statistical control) by using higher order cumulant statistical controls.

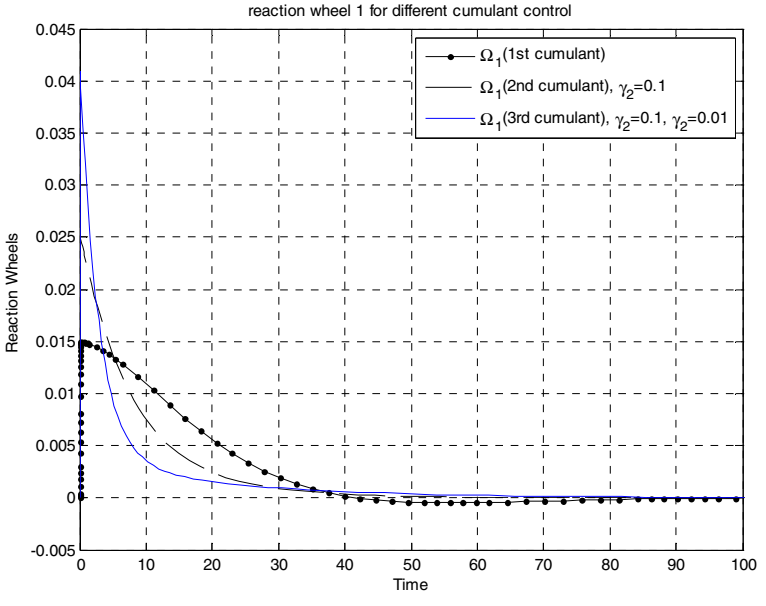


Fig. 4. Speed of reaction wheel one for the first, second, and third cumulant statistical control

5 Conclusions and Future Work

In this paper, we analyzed the statistical optimal control problem using cost cumulant approach. We investigate a method to minimize the different orders of the cost cumulants of the system cost. The control of different cumulants leads to different shapes of the distribution of the cost function. We developed the *n*-th cumulant control method which minimizes the cumulant of any order for a given stochastic system. The HJB equation for the *n*-th cumulant minimization is derived as necessary conditions of the optimality. The verification theorem, which is a sufficient condition, for the *n*-th cost cumulant case is also presented in this paper. We used neural network approximation method to solve HJB equations. Neural network approximation converts the partial differential HJB equation into the ordinary differential equation and is solved numerically. The Lagrange multiplier method is used with the neural network method to solve multiple HJB equations together to determine the optimal *n*-th cumulant controller. Then, a linear satellite attitude control example is given. The results of three different cost cumulant controls are presented and discussed. Statistical control improves the performance of a stochastic system.

References

1. Si, J., Barto, A., Powell, W., Wunsch II, D.: Handbook of Learning and Approximate Dynamic Programming. IEEE Press Series on Computational Intelligence. IEEE Press, Los Alamitos (2004)
2. Sain, M.K.: Control of Linear Systems According to the Minimal Variance Criterion—A New Approach to the Disturbance Problem. IEEE Transactions on Automatic Control AC-11(1), 118–122 (1966)
3. Sain, M.K., Liberty, S.R.: Performance Measure Densities for a Class of LQG Control Systems. IEEE Transactions on Automatic Control AC-16(5), 431–439 (1971)
4. Sain, M.K., Won, C.-H., Spencer Jr., B.F., Liberty, S.R.: Cumulants and Risk-Sensitive Control: A Cost Mean and Variance Theory with Application to Seismic Protection of Structures. In: Filar, J.A., Gaitsgory, V., Mizukami, K. (eds.) Advances in Dynamic Games and Applications, Annals of the International Society of Dynamic Games, vol. 5, pp. 427–459. Birkhäuser, Boston (2000)
5. Won, C.-H.: Nonlinear n-th Cost Cumulant Control and Hamilton-Jacobi-Bellman Equations for Markov Diffusion Process. In: Proceedings of 44th IEEE Conference on Decision and Control, Seville, Spain, pp. 4524–2529 (2005)
6. Kang, B., Won, C.-H.: Nonlinear second cost cumulant control using Hamilton-Jacobi-Bellman equation and neural network approximation. In: The Proceeding of American Control Conference, Baltimore, pp. 770–775 (2010)
7. Pham, K.D.: Statistical Control Paradigm for Structure Vibration Suppression. Ph.D Dissertation, University of Notre Dame, Notre Dame (2004)
8. Chen, T., Lewis, F.L., Abu-Khalaf, M.: A neural network solution for fixed-final time optimal control of nonlinear systems. Automatica 43, 482–490 (2007)
9. Al'brekht, E.G.: On the Optimal Stabilization of Nonlinear Systems. PMM—Journal of Applied Mathematics and Mechanics 25, 1254–1266 (1961)
10. Medagam, P.V., Pourboghrat, F.: Optimal Control of Nonlinear Systems using RBF Neural Network and Adaptive Extended Kalman Filter. In: Proceedings of the American Control Conference, St. Louis (2009)
11. Kang, B.: Statistical Control using Neural Network Methods with Hierarchical Hybrid Systems. Ph.D. dissertation. Temple University, Philadelphia (2010)
12. Sandberg, W.: Notes on Uniform Approximation of Time-Varying Systems on Finite Time Intervals. IEEE Transactions on Circuits and Systems-1: Fundamental Theory and Applications 45(8) (1998)
13. Finlayson, B.A.: The Method of Weighted Residuals and Variational Principles. Academic Press, New York (1972)
14. Lee, J.-H., Diersing, R.W., Won, C.-H.: Satellite Attitude Control Using Statistical Game Theory. In: Proceedings of the American Control Conference, Seattle, Washington (2008)
15. Won, C.-H.: Comparative Study of Various Control Methods for Attitude Control of a LEO Satellite. Aerospace Science and Technology (5), 323–333 (1999)

Adaptive Kernel-Width Selection for Kernel-Based Least-Squares Policy Iteration Algorithm

Jun Wu, Xin Xu, Lei Zuo, Zhaobin Li, and Jian Wang

Institute of Automation, National University of Defense Technology,
Changsha, 410073, P.R. China
aresnudt@yahoo.com.cn

Abstract. The Kernel-based Least-squares Policy Iteration (KLSPI) algorithm provides a general reinforcement learning solution for large-scale Markov decision problems. In KLSPI, the Radial Basis Function (RBF) kernel is usually used to approximate the optimal value-function with high precision. However, selecting a proper kernel-width for the RBF kernel function is very important for KLSPI to be adopted successfully. In previous research, the kernel-width was usually set manually or calculated according to the sample distribution in advance, which requires prior knowledge or model information. In this paper, an adaptive kernel-width selection method is proposed for the KLSPI algorithm. Firstly, a sparsification procedure with neighborhood analysis based on the l_2-ball of radius ε is adopted, which helps obtain a reduced kernel dictionary without presetting the kernel-width. Secondly, a gradient descent method based on the Bellman Residual Error (BRE) is proposed so as to find out a kernel-width minimizing the sum of the BRE. The experimental results show the proposed method can help KLSPI approximate the true value-function more accurately, and, finally, obtain a better control policy.

Keywords: reinforcement learning, sparsification, least-squares, gradient descent, kernel width.

1 Introduction

Reinforcement learning (RL) refers to a set of trial-and-error-based machine learning methods where an agent can potentially learn optimal policy in an uncertain dynamic environment [1]. It provides a knowledge-free methodology and is very promising to solve the optimization in complex sequential decision-making problems. However, the generalization problem is still an open issue in RL. The traditional RL methods, such as the Q-learning and Sarsa learning algorithms, find difficulties in solving the Markov decision problems (MDPs) with continuous state-action spaces.

In [2], Lagoudakis and Parr proposed the Least-Squares Policy Iteration (LSPI) algorithm. In LSPI, a set of basis functions are combined linearly to approximate the value functions, and it has been shown that the generalization and stability of LSPI are illustrated in MDPs with continuous states. Moreover, the Kernel-based LSPI (KLSPI) algorithm presented in [3] adopts kernel functions as approximators and provides an efficient method for solving RL problems with large-scale and continuous

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 611–619, 2011.

state spaces. In KLSPI, the Radial Basis Function (RBF) kernel is usually used to approximate the optimal value-function with high precision. However, how to select an appropriate kernel-width for a RBF kernel is crucial for the successful implementation of the KLSPI method. An improper kernel-width setting will prevent the learned value-function from approximating the true value function, and even prevents the learning process from converging to the optimal policy. In the past, the kernel-width selection in KLSPI was usually done manually and depended on human experiences and model information. In this paper, to overcome such difficulty, an adaptive kernel-width optimization method based on a Bellman Residual Error (BRE) gradient principle is proposed. Finally, simulations on the Inverted Pendulum problem are carried out to evaluate the new method.

The remainder of this paper is organized as follows. Section 2 introduces the kernel-width selection problem in KLSPI and gives an overview of the related works. Section 3 describes the key idea of the proposed method. Firstly, it adopts a sparsification procedure unrelated to kernel-width, and then introduces a gradient descent method based on Bellman residual error to optimize the kernel-width selection. Section 4 describes the simulation results. Finally, conclusions are drawn in Section 5.

2 Problem Statement and Related Works

Let S denote the original state space in an MDP. A kernel function is a mapping from $S \times S$ to R, which is usually assumed to be continuous. A Mercer kernel is a kernel function that is positive definite. According to the Mercer Theorem [4], there exists a Hilbert space H and a mapping φ from S to H such that

$$k(x_i, x_j) = < \varphi(x_i), \varphi(x_j) > , \tag{1}$$

where <.,.> is the inner product in H. Although the dimension of H may be infinite and the nonlinear mapping φ is usually unknown, all the computation in the feature space H can still be performed as the form of inner products in S. Due to the above properties of kernel functions, kernel methods have attracted many research interests. Researchers manage to kernelize the existing RL algorithms in linear spaces by selecting appropriate kernel functions in order to achieve better learning performance.

For instance, a kernel-based LS-TD learning method was proposed in [5] and the state-action value function can be represented by:

$$\tilde{Q}(s,a) = \sum_{i=1}^{t} \alpha_i k(x(s,a), x(s_i, a_i)) , \tag{2}$$

where $x(.,.)$ is the combined features of state-action pairs, α_i (i =1,2,…, t) are coefficients, and (s_i, a_i) (i =1,2,…, t) are state-action pairs for the sampled data.

After the ALD-based sparsification procedure [3], a data dictionary set D_n with reduced dimension will be obtained, and the state-action value function can be represented as follows:

$$\tilde{Q}(s,a) = \sum_{j=1}^{d_n} \alpha_j k(x(s,a), x(s_j, a_j)) , \tag{3}$$

where d_n is the length of the dictionary D_n and usually is much smaller than the original sample size t. By sparsification, the computational complexity as well as the memory cost of kernel methods can be greatly reduced and more benefits of generalization ability will also be obtained.

As mentioned above, kernel-based reinforcement learning algorithms adopt kernel functions to obtain nonlinear mapping abilities, and then improve the approximation ability and generalization ability remarkably. By introducing Mercer kernels in function approximation, KLSPI can be viewed as a kernelized LSPI algorithm [3]. The selection of kernel functions includes two parts: selecting appropriate kernel function type and selecting appropriate configuration parameters [6]. The Radial Basis Function (RBF) kernel is a kernel function in common use and can approximate any continuous function precisely by adopting appropriate parameters.

As a very important parameter, the width of the RBF kernel is always selected by two means: firstly, all the RBF functions' kernel-width parameters can be set as the same constant [7, 8]. In [8], the kernel-width was defined as $\sigma = d_{\max} / \sqrt{2M}$, where $d_{\max}$ is the maximal distance between the function centers, and M is the number of centers. Secondly, each RBF's width can be regulated individually. For instance, the kernel-width can be calculated according to the distance-deviation to corresponding centers. In [9], the kernel width is selected as the Euclid distance between the *ith* RBF's center and the nearest *jth* RBF's center. A *r*–nearest neighbor method was proposed in [10] to determine the kernel width as $\sigma_j = \sqrt{\sum_{i=1}^{r} \left\| c_i - c_j \right\|^2} \Big/ r$, where c_i is the *r*-nearest neighbor of the RBF's center c_j, r is the number of the neighbors, which is always set as 2. This method takes advantage of the data distribution and gains an advantage over the fixed width selection methods. Moreover, an adaptive clustering-based method is proposed in [11] and [12] and a supervised learning method is proposed in [13] and [14]. Obviously, all the above methods select the kernel-width according the data's distribution in the input space and can't optimize the width parameter after selection. In [15], a kernel-width selection method was proposed based on the mean prediction error (MPE) formula for the faulty RBF neural networks. However, it needs a kernel-width candidate set as well as the corresponding prior knowledge, too. Therefore, in the following, an adaptive kernel-width optimization method will be proposed for KLSPI using RBF kernels.

3 Adaptive Optimization for Kernel-Width

3.1 Basic Ideas of the Adaptive Kernel-Width Optimization

In KLSPI, after the samples were collected, the centers and the kernel-widths of the RBF kernel functions both affect the learning performance. To select proper centers, KLSPI adopts a kernel sparsification method based on the approximate linear dependence (ALD) analysis [3], which depends on the presetting kernel-width. If the kernel-width is optimized and changed, the sparsification process has to be executed again. To simplify the optimization procedure, in this paper, the kernel sparsification and kernel-width optimization are decoupled, which is shown in the following Fig. 1.

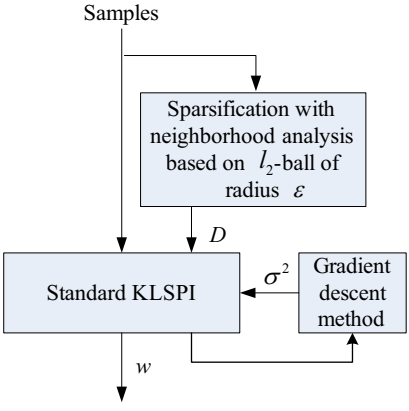


Fig. 1. The schematic diagram for kernel-width optimization in KLSPI

As depicted in Fig. 1, after the samples were obtained, a sparsification procedure is carried out to obtain a reduced dictionary set D. To remove the influence of the kernel-width from sparsification, a neighborhood analysis based on a l_2-ball of radius ε is done instead of the ALD analysis. After the revised sparsification procedure is completed, a standard KLSPI is executed based on a kernel-width σ initialized randomly. At the same time, a kernel-width optimization process by minimizing the Bellman Residual Error (BRE) is performed. The implementation of the new method will be illustrated in more detail in the following context.

3.2 Sparsification with the Neighborhood Analysis Based on a l_2-Ball of Radius ε

In KLSPI, an ALD-based sparsification procedure is executed as follows. Given a dictionary set $D_{t-1}=\{s_j\}$ (j=1,2,…d_{t-1}), for a new sample s_t, the ALD condition is calculated as the following inequality:

$$\delta_t = \min_c \left\| \sum_j c_j \varphi(s_j) - \varphi(s_t) \right\|^2 \le \mu \ , \tag{4}$$

where $c=[c_j]^T$, μ is the threshold parameter for controlling sparsification. Due to the kernel trick, we can obtain

$$\delta_t = \min_c \{ C^T K_{t-1} c - 2c^T K_{t-1}(s_t) + k_{tt} \} \ , \tag{5}$$

where $[K_{t-1}]_{i,j}=k(s_i, s_j)$, s_i and s_j (i,j=1,2,…,d_{t-1}) are the elements in the dictionary, d_{t-1} is the size of the dictionary, $k_{t-1}(s_t)=[k(s_1,s_t), k(s_2, s_t),…,k(s_{d(t-1)},s_t)]^T$, $c=[c_1, c_2,…,c_{d(t-1)}]^T$ and $k_{tt}=k(s_t, s_t)$. The least-squares solution for equation (5) is

$$\delta_t = k_{tt} - k_{t-1}^{-1}(s_t) K_{t-1}^{-1} k_{t-1}(s_t) \ . \tag{6}$$

Obviously, δ_t is influenced by the kernel-width. To remove its effect, in this paper, a sparsification method with the neighborhood analysis based on a l_2-ball of radius ε is proposed. Given an obtained dictionary as $D_{t-1}=\{s_j\}$ (j=1,2,…d_{t-1}) , for a new sample s_t, the linear dependence is calculated as:

$$\delta_t = \min_{j \in D_{t-1}} \left\| s_j - s_t \right\|^2 \le \mu \ , \tag{7}$$

where μ is the threshold parameter, too. If the above inequality holds, then the dictionary D keeps unchanged, if not, update D as follows:

$$\mathrm{D}_t = \mathrm{D}_{t-1} \bigcup [\bigcup_{a \in A} x(s_t, a)] \ . \tag{8}$$

3.3 Optimization of Kernel-width in KLSPI

With the new sparsification method, a dictionary without the dependence on the kernel width is obtained. So an optimization procedure for kernel-width selection can be executed. In KLSPI, a least-squares method is adopted to approximate the Q value function. Essentially, the least-squares method is to minimize the sum of the approximation error, so, in this paper, a method based on the minimization of the BRE (Bellman Residual Error) is proposed similarly.

Given the samples set $S = \{(s_i, a_i, s_i', a_i', r_i) \mid i = 1, 2,\ldots, N\}$, the BRE of the sample (s, a, s', a', r) is defined as:

$$e = \hat{Q}(s,a) - r - \gamma \hat{Q}(s',a') = \sum_{j=1}^{L} \phi_j(s,a) w_j - r - \gamma \sum_{j=1}^{L} \phi_j(s',a') w_j \ . \tag{9}$$

The objective function is defined as:

$$E = \frac{1}{2} \sum_{i=1}^{N} e_i^2 = \frac{1}{2} \sum_{i=1}^{N} \left(\sum_{j=1}^{L} \phi_j(s_i,a_i) w_j - r_i - \gamma \sum_{j=1}^{L} \phi_j(s_i',a_i') w_j \right)^2 \ . \tag{10}$$

The objective function E is decided by the kernel-width σ^2 (for the convenience of representation, a variable σ^2 is used instead of σ in the following context) and the weight vector ω jointly. Given σ^2, ω can be calculated with the LSPI method; on the other hand, given the weight vector ω, the objective function E can be optimized with a gradient descent method to obtain the best σ^2.

To minimize the objective function E, in this paper, a kernel-width optimization method, which is an iterative operation between the calculation of the weight vector and the calculation of the kernel-width, is proposed. When the extreme value of the E is obtained, the iterative operation stops, and then the final kernel-width can be obtained.

To calculate the weight vector, a standard least-squares method is adopted to optimize the objective function:

$$\omega = \left[\left(\phi(s,a) - \gamma \phi(s',a') \right)^T \left(\phi(s,a) - \gamma \phi(s',a') \right) \right]^{-1} \times \left(\phi(s,a) - \gamma \phi(s',a') \right) R \ . \tag{11}$$

To optimize the kernel width, a gradient descent method is adopted as follows. Firstly, the partial derivative for kernel-width σ^2 to the *j-th* kernel function in *D* is defined as:

$$\frac{\partial \phi_j(s,a)}{\partial(\sigma^2)} = \begin{cases} 0 & if\ a \neq a_j \\ e^{-\|s-s_j\|^2/\sigma^2}\left(\|s-s_j\|^2/(\sigma^2)^2\right) & else \end{cases} . \tag{12}$$

Then, the partial derivative for objective function can be obtained as follows:

$$\frac{\partial E}{\partial(\sigma^2)} = \frac{1}{2}\sum_{i=1}^{N}\{e_i^2[\sum_{j=1}^{L}(\frac{\partial \phi_j(s_i,a_i)}{\partial(\sigma^2)}w_j) - r_i - \gamma\sum_{j=1}^{L}(\frac{\partial \phi_j(s_i',a_i')}{\partial(\sigma^2)}w_j)]\} . \tag{13}$$

The optimization operation for the kernel-width selection can be done as:

$$\sigma^2 = \begin{cases} \sigma^2 - l & h\partial E/\partial(\sigma^2) > l \\ \sigma^2 + l & h\partial E/\partial(\sigma^2) < -l \\ \sigma^2 - h\cdot\partial E/\partial(\sigma^2) & others \end{cases} , \tag{14}$$

where *h* and *l* are positive constants respectively, *h* is the regulation step, *l* is the threshold parameter for controlling optimization in one step.

4 Experiments and Evaluations

In this paper, a stochastic Inverted Pendulum problem is used to evaluate the proposed method, which requires balancing an inverted pendulum of unknown length and mass at the upright position by applying forces to the cart which is attached to. This problem is depicted as Fig. 2.

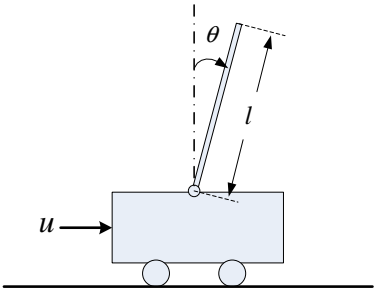


Fig. 2. This shows a figure illustrating the Inverted Pendulum problem

There are three actions: {-50, 0, 50}(*N*), and an uniform noise in [-10,10] (*N*) is added to the chosen action. The state space of the problem consists of the vertical angle θ and the angular velocity $\dot{\theta}$ of the pendulum. The transitions are governed by

the nonlinear dynamics of the system and depend on the current state and the current noisy control u [2]:

$$\ddot{\theta} = \frac{g\sin(\theta) - \alpha ml\dot{\theta}^2 \sin(2\theta)/2 - \alpha\cos(\theta)u}{4l/3 - \alpha ml\cos^2(\theta)} , \tag{15}$$

where g is the gravity constant (g = 9.8m/s^2), m is the mass of the pendulum (m = 2.0kg), l is the length of the pendulum(l = 0.5m), and $\alpha = 1.0/(\text{m} + M_{cart})$, M_{cart} is the mass of the cart (M_{cart} = 8.0kg). The simulation step is set to be 0.1 second. A reward 0 is given as long as the angle of the pendulum does not exceed $\pi/2$ in absolute value, and an angle greater than $\pi/2$ signals the end of the episode and a penalty of -1 is given. The discount factor of the process is set to be 0.90. The kernel function is selected as:

$$k(s_i, s_j) = exp(-\left[(\theta_i - \theta_j)^2 + (\dot{\theta}_i - \dot{\theta}_j)^2\right]/\sigma^2) , \tag{16}$$

where points s_i and s_j in S are$(\theta_i, \dot{\theta}_i)$and $(\theta_j, \dot{\theta}_j)$ respectively.

To evaluate the proposed method, the experiments were done as follows: different numbers of sampling episodes were chosen as 10, 20, 30, …, 200. The sampling strategy is random absolutely, i.e. choosing the action among three given actions randomly. Each episode starts from the state initialized as (0, 0) and ends once the $|\theta| > \pi/2$ or the simulation steps are greater than 3000. Furthermore, to remove randomness in the simulation results, each experiment was repeated 100 times with different random seeds and its data was collected and statistically processed. The experimental results are shown in Fig.3 and Fig.4 respectively.

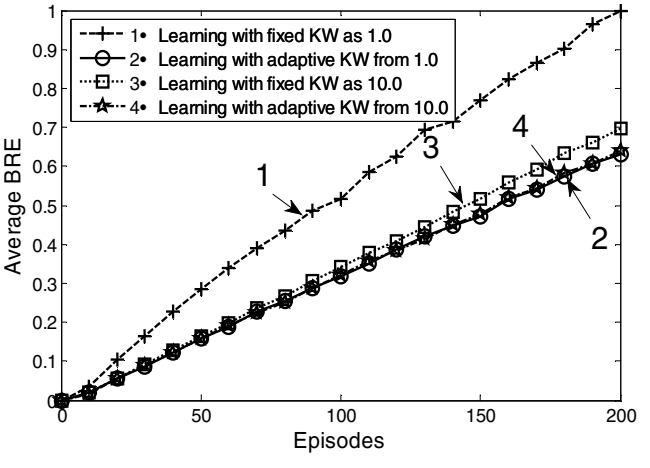


Fig. 3. Performance comparisons of BRE results with/without kernel-width optimization

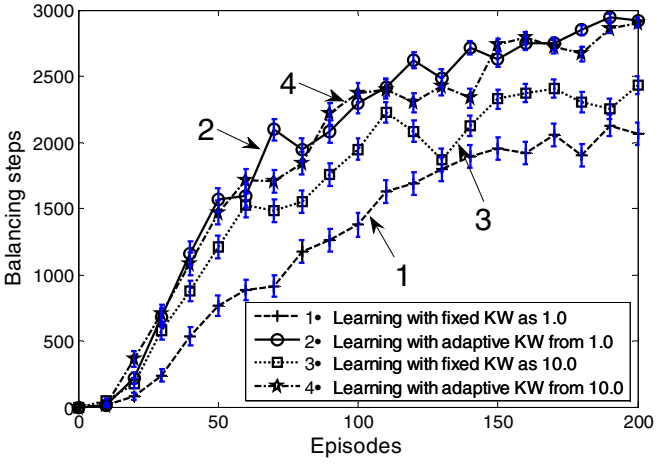


Fig. 4. Balancing steps with/without kernel-width optimization

Fig.3 shows how the number of the episodes and the new method impact the results of average Bellman Residual Error. The curve 1 and curve 3 denote the average BRE results when the kernel-width (KW) is initialized as 1.0 and 10.0 respectively. The curve 2 and curve 4 denote the average BRE results when the kernel-width is set according to the adaptive optimization method. Obviously, the results with the kernel-width optimization are better than the ones without optimization. So, it is evident that the proposed method can optimize the kernel-width effectively, and help the learned function approximate the true Q function in more precise representation.

Fig.4 shows that how the kernel-width optimization method improves the control performance. The average balancing steps resulting from the new method (as depicted by curve 2 and curve 4) are greater than the ones resulting from the original KLSPI (as depicted by curve 1 and curve 3). Therefore, it is shown that the new method helps obtain better control policy.

Additionally, while the number of episodes comes up to 200, although the initial kernel-widths are set differently, such as 1.0 or 10.0, all the optimized kernel-widths can converge to the same value, i.e.14.3, approximatively. So the proposed method is robust to different initial values and can achieve a local optimum stably.

According to the above experiments, the feasibility and validity of the proposed method are shown.

5 Conclusion

In this paper, an adaptive kernel-width selection method is proposed to ease the use of the RBF kernel in KLSPI. A neighborhood analysis method based on a l_2-ball of radius ε is proposed and a gradient descent method based on the Bellman residual errors is proposed for optimizing the kernel-width. The experiments on the Inverted Pendulum problem show the new method can improve the precision of the

approximation and the quality of the learned control policy. Although the results in this paper are very encouraging, more experiments on real-world problems need to be carried out extensively. Moreover, some new kernel functions and corresponding parameter-optimization methods need to be developed in the future.

Acknowledgments. This work is supported in part by the National Natural Science Foundation of China under Grant 60774076, 61075072 and 90820302, the Fork Ying Tung Education Foundation under Grant 114005, and the Natural Science Foundation of Hunan Province under Grant 2007JJ3122.

References

1. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
2. Michai, G.L., Parr, R.: Least-Squares Policy Iteration. Journal of Machine Learning Research 4, 1107–1149 (2003)
3. Xu, X., Hu, D.W., Lu, X.C.: Kernel-based Least Squares Policy Iteration for Reinforcement Learning. IEEE Transactions on Neural Networks 18(4), 973–992 (2007)
4. Vapnik, V.: Statistical Learning Theory. Wiley Interscience, NewYork (1998)
5. Xu, X., Xie, T., Hu, D.W., et al.: Kernel Least-Squares Temporal Difference Learning. Int. J. Inf. Technol. 11(9), 54–63 (2005)
6. Wu, T.: Kernels' Properties, Tricks and Its Applications on Obstacle Detection. National University of Defense Technology, Doctor Thesis (2003)
7. Orr, M.J.L.: Introduction to Radial Basis Functions. Networks (1996)
8. Haykin, S.: Neural Networks-a Comprehensive Foundation. Prentice-Hall, Englewood Cliffs (1999)
9. Moody, J., Darken, C.J.: Fast Learning In Networks of Locally-Tuned Processing Units. Neural Computation 1(2), 281–294 (1989)
10. Archambeau, C., Lendasse, A., Trullemans, C., et al.: Phosphene Evaluation in a Visual Prosthesis with Artificial Neural Networks. In: Proceedings of the European Symposium on Intelligent Technologies, Hybrid Systems and their Implementation on Smart Adaptive Systems, Tenerife, Spain, pp. 509–515 (2001)
11. Wang, Y., Huang, G., Saratchandran, P., et al.: Self- Adjustment of Neuron Impact Width in Growing and Pruning RBF (GAP-RBF) Neuron Networks. In: Proceedings of ICS 2005, vol. 2, pp. 1014–1017 (2003)
12. Gao, D.Q.: Adaptive Structure and Parameter Optimizations of Cascade RBF-LBF Neural Networks. Chinese Journal of Computers 26(5), 575–586 (2003)
13. Chang, Q., Chen, Q., Wang, X.: Scaling Gaussian RBF Kernel Width to Improve SVM Classification. In: International Conference on Neural Networks and Brain, pp. 19–22 (2005)
14. Liu, J.H., Lampinen, J.: A Differential Evolution Based Incremental Training Method for RBF Networks. In: Proceedings of GECCO 2005, Washington, DC, USA, pp. 881–888 (2005)
15. Wang, H.J., Leung, C.S., Sum, P.F., et al.: Kernel Width Optimization for Faulty RBF Neural Networks with Multi-node Open Fault. Neural Processing Letters 32(1), 97–107 (2010)

Finite Horizon Optimal Tracking Control for a Class of Discrete-Time Nonlinear Systems

Qinglai Wei, Ding Wang, and Derong Liu*

Institute of Automation, Chinese Academy of Sciences, 100190, China
{qinglai.wei,ding.wang,derong.liu}@ia.ac.cn

Abstract. In this paper, a new iterative ADP algorithm is proposed to solve the finite horizon optimal tracking control problem for a class of discrete-time nonlinear systems. The idea is that using system transformation, the optimal tracking problem is transformed into optimal regulation problem, and then the iterative ADP algorithm is introduced to deal with the regulation problem with convergence guarantee. Three neural networks are used to approximate the performance index function, compute the optimal control policy and model the unknown system dynamics, respectively, for facilitating the implementation of iterative ADP algorithm. An example is given to demonstrate the validity of the proposed optimal tracking control scheme.

Keywords: Adaptive dynamic programming, approximate dynamic programming, optimal tracking control, neural networks, finite horizon.

1 Introduction

The optimal tracking problem of nonlinear systems has always been the key focus in the control field in the latest several decades. Traditional optimal tracking control is mostly implemented by feedback linearization [1]. However, the controller designed by feedback linearization technique is only effective in the neighborhood of the equilibrium point. When the required operating range is large, the nonlinearities in the system cannot be properly compensated by using a linear model. Therefore, it is necessary to study the direct optimal tracking control approach for the original nonlinear system. The difficulty for nonlinear optimal feedback control lies in solving the time-varying HJB equation which is usually too hard to solve analytically. In order to overcome the difficulty, in [2], the finite-time optimal tracking control problem was solved via transforming the system model into a sequence of "pseudo-linear" systems. In [3], an infinite horizon approximate optimal tracking controller based on the successive approximation approach was proposed. However, the literature mentioned above is all restricted in the continuous-time domain. There are few results discussing the optimal tracking control problem for discrete-time systems. To the best of our knowledge, only [4] has presented the optimal tracking control scheme in infinite horizon domain. There are no results on the finite horizon optimal tracking control for discrete-time nonlinear systems. This motivates our research.

* This work was supported in part by the NSFC under grants 60904037, 60921061, 61034002, and by Beijing Natural Science Foundation under grant 4102061.

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 620–629, 2011.

As is known, dynamic programming is very useful in solving the optimal control problems. However, due to the "curse of dimensionality", it is often computationally untenable to run dynamic programming to obtain the optimal solution. The approximate dynamic programming (ADP) algorithm was proposed by Werbos [5] as a way to solve optimal control problems forward-in-time. ADP combines adaptive critic design, reinforcement learning technique with dynamic programming. In [5] adaptive dynamic programming approaches were classified into four main schemes: Heuristic Dynamic Programming (HDP), Dual Heuristic Dynamic Programming (DHP), Action Dependent Heuristic Dynamic Programming (ADHDP), also known as Q-learning, and Action Dependent Dual Heuristic Dynamic Programming (ADDHP). Though in recent years, ADP has been further studied by many researchers [6, 7, 8, 9, 10, 12, 11], most results are focus on the optimal regulation problem. In [13], a greedy HDP iteration algorithm to solve the discrete-time Hamilton-Jacobi-Bellman (DT HJB) equation of the optimal regulation control problem for general nonlinear discrete-time systems is proposed, which does not require an initially stable policy. It has been rigorously proved in [13] that the greedy HDP iteration algorithm is convergent. To the best of our knowledge, till now only in [4], ADP was used to solve the infinite-time optimal tracking control problem. There have been no results discussing how to use ADP to solve the finite-time optimal tracking control problem for nonlinear systems.

In this paper, it is the first time to solve finite horizon optimal tracking control problem for a class of discrete-time nonlinear systems using ADP. We firstly transform the tracking problem into an optimal regulation problem, and then a new iterative ADP algorithm can be properly introduced to deal with this regulation problem.

2 Paper Preparation

Consider the following discrete-time nonlinear system

$$x_{k+1} = f(x_k) + g(x_k)u_k \tag{1}$$

where $x_k \in \Re^n$ and the input $u_k \in \Re^m$. Here assume that the system is controllable. In this paper, the reference orbit η_k is generated by the n-dimensional autonomous system as $\eta_{k+1} = S(\eta_k)$, where $\eta_k \in \Re^n$, $S(\eta_k) \in \Re^n$. Therefore we define the tracking error as:

$$z_k = x_k - \eta_k. \tag{2}$$

Let $\underline{v}_k$ be an arbitrary finite-horizon tracking control sequence starting at k and let $\mathcal{U}_{z_k} = \left\{\underline{v}_k \colon z^{(f)}\left(z_k, \underline{v}_k\right) = 0\right\}$ be the set of all finite-horizon tracking control sequences of x_k. Let $\mathcal{U}_{z_k}^{(i)} = \left\{\underline{v}_k^{k+i-1} \colon z^{(f)}\left(z_k, \underline{v}_k^{k+i-1}\right) = 0,\ \left|\underline{v}_k^{k+i-1}\right| = i\right\}$ be the set of all finite-horizon admissible control sequences of z_k with length i, where the final state error can be written as $z^{(f)}\left(z_k, \underline{v}_k^{k+i-1}\right) = z_{k+i}$. Then, $\mathcal{U}_{z_k} = \cup_{1 \le i < \infty} \mathcal{U}_{z_k}^{(i)}$. By this notation, a state error z_k is controllable if and only if $\mathcal{U}_{z_k} \neq \emptyset$.

Noticing that the objective in this paper is to design an optimal feedback control policy v_k, which not only renders the state error z_k asymptotically tracking the reference orbit, $i.e.$, z_k asymptotically approaches zero, but also minimizes the performance index function as follow

$$J(z_k, \underline{v}_k^{N-1}) = \sum_{i=k}^{N-1} \left\{ z_i^T Q z_i + v_i^T R v_i \right\}, \tag{3}$$

where Q and R are positive-definite matrices. $U(k) = z_k^T Q z_k + v_k^T R v_k$ is the utility function. In addition, we define

$$v_k = u_k - u_{ek}, \tag{4}$$

where u_{ek} is the steady control input expressed as

$$u_{ek} = g^{-1}(\eta_k)(\eta_{k+1} - f(\eta_k)) \tag{5}$$

Combining (2) with (5) , we can get

$$\begin{aligned} z_{k+1} = F(z_k, v_k) = &-S(\eta_k) + f(z_k + \eta_k) + g(z_k + \eta_k)\, v_k \\ &-g(z_k + \eta_k) g^{-1}(\eta_k)(f(\eta_k) - S(\eta_k)). \end{aligned} \tag{6}$$

For any given system state error z_k, the objective of the present finite-horizon optimal control problem is to find a finite-horizon admissible control sequence $\underline{v}_k^{N-1} \in \mathcal{U}_{z_k}^{(N-k)} \subseteq \mathcal{U}_{x_k}$ to minimize the performance index $J\left(z_k, \underline{v}_k^{N-1}\right)$. The control sequence $\underline{v}_k^{N-1}$ has finite length. However, before it is determined, we do not know its length which means that the length of the control sequence $\left|\underline{v}_k^{N-1}\right| = N - k$ is unspecified. This kind of optimal control problems has been called finite-horizon problems with unspecified terminal time.

3 Properties of the Iterative Adaptive Dynamic Programming Algorithm

In this section, a new iterative ADP algorithm is proposed to obtain the finite horizon optimal tracking control for nonlinear systems. The goal of the proposed iterative ADP algorithm is to construct an optimal control policy $v^*(z_k)$, $k = 0, 1, \ldots$, which makes an arbitrary initial state error z_0 to the singularity 0 within finite time, simultaneously makes the performance index function reach the optimum $V^*(z_k)$. Convergence proofs will also be given.

3.1 Derivation of the Iterative ADP Algorithm

In the iterative ADP algorithm, the performance index function and control policy are updated by recurrent iteration, with the iteration number i increasing from 0. Let the initial performance index function $V_0(z_k) = 0$ and there exists a control v_k that makes $F(z_k, v_k) = 0$, where z_k is any initial state error. Then, the iterative control $v_0(z_k)$ can be computed as follows:

$$\begin{aligned} v_0(z_k) = &\arg\min_{v_k} \left\{ U(z_k, v_k) + V_0(z_{k+1}) \right\}, \\ &\text{s.t. } z_{k+1} = F(z_k, v_k) = 0 \end{aligned} \tag{7}$$

where $V_0(z_{k+1}) = 0$. The performance index function can be updated as

$$V_1(z_k) = U(z_k, v_0(z_k)) + V_0(F(z_k, v_0(z_k)). \tag{8}$$

For $i = 1, 2, \ldots$, the iterative ADP algorithm will iterate between

$$\begin{aligned} v_i(z_k) &= \arg\min_{v_k} \{U(z_k, v_k) + V_i(z_{k+1})\} \\ &= \arg\min_{v_k} \{U(z_k, v_k) + V_i(F(z_k, v_k))\} \end{aligned} \tag{9}$$

and performance index function

$$\begin{aligned} V_{i+1}(z_k) &= \min_{v_k} \{U(z_k, v_k) + V_i(z_{k+1})\} \\ &= U(z_k, v_i(z_k)) + V_i(F(z_k, v_i(z_k)). \end{aligned} \tag{10}$$

3.2 Properties of the Iterative ADP Algorithm

In the above, we can see that the performance index function $V^*(z_k)$ is replaced by a sequence of iterative performance index functions $V_i(z_k)$ and the optimal control law $v^*(z_k)$ is replaced by a sequence of iterative control law $v_i(z_k)$, where $i \geq 0$ is the iterative index. As (10) is not an HJB equation for $\forall i \geq 0$, generally, the iterative performance index function $V_i(z_k)$ is not optimal. However, we can prove that $V^*(z_k)$ is the limit of $V_i(z_k)$ as $i \to \infty$.

Theorem 1. *Let z_k be an arbitrary state error vector. Suppose that there is a positive integer i such that $\mathcal{U}_{z_k}^{(i)} \neq \emptyset$. Then, for $\mathcal{U}_{z_k}^{(i+1)} \neq \emptyset$, the performance index function $V_i(z_k)$ obtained by (7)–(10) is a nonincreasing convergent sequence for $\forall\, i \geq 1$, i.e., $V_{i+1}(z_k) \leq V_i(z_k)$.*

Proof. We prove this by mathematical induction. First, we let $i = 1$. Then, We have

$$\begin{aligned} V_1(z_k) &= \min_{v_k} \{U(z_k, v_k) + V_0(F(z_k, v_k))\} \\ &= \min_{v_k} \{U(z_k, v_k)\} = U(z_k, v_0(z_k)) \end{aligned} \tag{11}$$

where $V_0(F(z_k, v_0(z_k)) = 0$. The finite horizon admissible control sequence $\underline{v}_k^k = (v_0(z_k))$.

Next, let us show that there exists a finite horizon admissible control sequence $\hat{\underline{v}}_k^{k+1}$ with length 2 such that $V_1(z_k, \underline{v}_k^k) = \hat{V}_2(z_k, \hat{\underline{v}}_k^{k+1})$. Obviously, $v_0(z_k) \in \mathcal{U}_{z_k}^{(1)}$. The trajectory starting from z_k under the control of $\underline{v}_k^k$ is $z_{k+1} = F(z_k, v_0(z_k)) = 0$. Then, we create a new control sequence $\hat{\underline{v}}_k^{k+1}$ by adding a 0 at the end of sequence $\underline{v}_k^k$ to obtain the control sequence $\hat{\underline{v}}_k^{k+1} = (\underline{v}_k^k, 0)$. Obviously, $|\hat{\underline{v}}_k^{k+1}| = 2$. The state error trajectory under the control of $\hat{\underline{v}}_k^{k+1}$ is $z_{k+1} = F(z_k, v_0(z_k))$, $z_{k+2} = F(z_{k+1}, v_{k+1})$ where $v_{k+1} = 0$. As $z_{k+1} = 0$ and $F(0, 0) = 0$, we have $z_{k+2} = F(z_{k+1}, v_{k+1}) = 0$. So, $\hat{\underline{v}}_k^{k+1}$ is a finite horizon admissible control. Furthermore,

$$\begin{aligned} V_1(z_k, \hat{\underline{v}}_k^k) &= U(z_k, v_k) \\ &= U(z_k, v_k) + U(z_{k+1}, v_{k+1}) = \hat{V}_2(z_k, \hat{\underline{v}}_k^{k+1}). \end{aligned} \tag{12}$$

On the other hand, we have

$$V_2(z_k) = \min_{v_k} \{U(z_k, v_k) + V_1(F(z_k, v_k)\}. \tag{13}$$

According to (11), we have

$$V_2(z_k) = \min_{\underline{v}_k^{k+1}} \{U(z_k, v_k) + U(z_{k+1}, v_{k+1})\} \tag{14}$$

where $z_{k+2} = F(z_{k+1}, v_{k+1}) = 0$. Then we have

$$V_2(z_k) \le \hat{V}_2(z_k, \underline{\hat{v}}_k^{k+1}). \tag{15}$$

So the theorem holds for $i = 1$. Assume that the theorem holds for any $i = l - 1$, where $l \ge 1$. We have

$$V_l(z_k) = \min_{v_k} \{U(z_k, v_k) + V_{l-1}(F(z_k, v_k))\} \tag{16}$$

where the corresponding finite horizon admissible control sequence is $\underline{v}_k^{k+l-1}$.

Then for $i = l$, we create a control sequence $\underline{\hat{v}}_k^{k+l} = \{\underline{v}_k^{k+l-1}, 0\}$ with length $l + 1$. Then the state error trajectory under the control of $\underline{\hat{v}}$ is $z_{k+1} = F(z_k, v_l(z_k))$, $z_{k+2} = F(z_{k+1}, v_{l-1}(z_{k+1})), \ldots, z_{k+l} = F(z_{k+l}, v_0(z_{k+l})) = 0$, $z_{k+l+1} = 0$. So $\underline{\hat{v}}_k^{k+l}$ is finite horizon admissible control. The performance under the control sequence is

$$\begin{aligned} V_{l+1}(z_k, \underline{\hat{v}}_k^{k+l}) =& U(z_k, v_l(z_k)) + U(z_{k+1}, v_{l-1}(z_{k+1})) \\ &+ \ldots + U(z_{k+l}, v_0(z_{k+l})) + U(z_{k+l+1}, 0) \\ =& \sum_{j=0}^{l+1} U(z_{k+j}, v_{i-j}(z_{k+j})) \end{aligned} \tag{17}$$

where $v_{l-j} = 0$ for all $l < j$.

On the other hand, we have

$$V_{i+1}(z_k) = \min_{v_k} \{U(z_k, v_k) + V_i(F(z_k, v_k)\} = \min_{\underline{v}_k^{k+i}} \left\{ \sum_{j=0}^{i+1} U(z_{k+j}, v_{i-j}(z_{k+j})) \right\}. \tag{18}$$

Then, we have

$$V_{l+1}(z_k) \le V_{l+1}(z_k, \underline{\hat{v}}_k^{k+l}) = V_l(z_k) \tag{19}$$

The proof is completed.

Lemma 1. *Let $\mu_i(z_k), i = 0, 1 \ldots$ be any sequence of tracking control, and $v_i(z_k)$ is expressed as (9). Define $V_{i+1}(z_k)$ as (10) and $\Lambda_{i+1}(z_k)$ as*

$$\Lambda_{i+1}(z_k) = U(z_k, \mu_i(z_k)) + \Lambda_i(z_{k+1}). \tag{20}$$

Then if $V_0(z_k) = \Lambda_0(z_k) = 0$, we have $V_i(z_k) \le \Lambda_i(z_k)$, $\forall i$.

According to Theorem 1, we know that the performance index function $V_i(z_k) \geq 0$ is a nonincreasing bounded sequence for iteration index $i = 1, 2, \ldots$. Then we can derive the following theorem.

Theorem 2. *Let z_k be an arbitrary state error vector. Define the performance index function $V_\infty(z_k)$ as the limit of the iterative function $V_i(z_k)$, i.e.,*

$$V_\infty(z_k) = \lim_{i \to \infty} V_i(z_k). \tag{21}$$

Then, we have the following HJB equation

$$V_\infty(z_k) = \min_{v_k} \{U(z_k, v_k) + V_\infty(z_{k+1})\} \tag{22}$$

holds.

Proof. Let $\eta_k = \eta(z_k)$ be any admissible control. According to Theorem 1, for $\forall i$, we have

$$V_\infty(z_k) \leq V_{i+1}(z_k) \leq U(z_k, \eta_k) + V_i(z_{k+1}). \tag{23}$$

Let $i \to \infty$, we have

$$V_\infty(z_k) \leq U(z_k, \eta_k) + V_\infty(z_{k+1}). \tag{24}$$

So

$$V_\infty(z_k) \leq \min_{v_k} \{U(z_k, \eta_k) + V_\infty(z_{k+1})\}. \tag{25}$$

Let $\epsilon > 0$ be an arbitrary positive number. Since $V_i(z_k)$ is nonincreasing for $\forall\, i$ and $\lim_{i \to \infty} V_i(z_k) = V_\infty(z_k)$, there exists a positive integer p such that

$$V_p(z_k) - \epsilon \leq V_\infty(z_k) \leq V_p(z_k). \tag{26}$$

Then, we let

$$\begin{aligned} V_p(z_k) &= \min_{v_k} \{U(z_k, v_k) + V_p(z_{k+1})\} \\ &= U(z_k, v_{p-1}(z_k)) + V_{p-1}(z_{k+1}). \end{aligned} \tag{27}$$

Hence

$$\begin{aligned} V_\infty(z_k) &\geq U(z_k, v_{p-1}(z_k)) + V_{p-1}(z_{k+1}) - \epsilon \\ &\geq U(z_k, v_{p-1}(z_k)) + V_\infty(z_{k+1}) - \epsilon \\ &\geq \min_{v_k} \{U(z_k, v_k) + V_\infty(z_{k+1})\} - \epsilon. \end{aligned} \tag{28}$$

Since ϵ is arbitrary, we have

$$V_\infty(z_k) \geq \min_{v_k} \{U(z_k, v_k) + V_\infty(z_{k+1})\}. \tag{29}$$

Combining (25) and (29) we have

$$V_\infty(z_k) = \min_{v_k} \{U(z_k, v_k) + V_\infty(z_{k+1})\} \tag{30}$$

which proves the theorem.

Next, we will prove that the iterative performance index function $V_i(z_k)$ converges to the optimal performance index function $V^*(z_k)$ as $i \to \infty$.

Theorem 3. *Let the performance index function $V_i(z_k)$ be defined by (10). If the system state error z_k is controllable, then the performance index function $V_i(z_k)$ converges to the optimal performance index function $V^*(z_k)$ as $i \to \infty$, i.e.,*

$$V_i(z_k) \to V^*(z_k). \tag{31}$$

Proof. As

$$V^*(z_k) = \min\left\{V(z_k, \underline{v}_k) \colon \underline{v}_k \in \mathcal{U}_{z_k}^{(i)}\right\}, \quad i = 1, 2, \ldots. \tag{32}$$

we have

$$V^*(z_k) \le V_i(z_k). \tag{33}$$

Then, let $i \to \infty$, we have

$$V^*(z_k) \le V_\infty(z_k). \tag{34}$$

Let $\epsilon > 0$ be an arbitrary positive number. Then there exists a finite horizon admissible control sequence η_q such that

$$V_q(z_k) \le V^*(z_k) + \epsilon. \tag{35}$$

On the other side, according to Lemma 1, for any finite horizon admissible control η_q, we have

$$V_\infty(z_k) \le V_q(z_k) \tag{36}$$

holds.

Combining (35) and (36), we have

$$V_\infty(z_k) \le V^*(z_k) + \epsilon. \tag{37}$$

As ϵ is arbitrary positive number, we have

$$V_\infty(z_k) \le V^*(z_k). \tag{38}$$

According to (34) and (38), we have

$$V_\infty(z_k) = V^*(z_k). \tag{39}$$

The proof is completed.

Then we can derive the following corollary.

Corollary 1. *Let the performance index function $V_i(z_k)$ be defined by (10). If the system state error z_k is controllable and Theorem 3 holds, then the iterative control law $v_i(z_k)$ converges to the optimal control law $v^*(z_k)$.*

3.3 The Procedure of the Algorithm

Now we summarize the iterative ADP algorithm for the time-variant optimal tracking control problem as:

Step 1. Give $x(0)$, $i_{\max}$, ε, desired trajectory η_k.
Step 2. Set $i = 0$, $V_0(z_k) = 0$.
Step 3. Compute $v_0(z_k)$ by (7) and $V_1(z_k)$ by (8).
Step 4. Set $i = i + 1$.
Step 5. Compute $v_i(z_k)$ by (9) and $V_{i+1}(z_k)$ by (10).
Step 6. If $|V_{i+1}(z_k) - V_i(z_k)| < \varepsilon$ then go to step 8, else go to step 7.
Step 7. If $i > i_{\max}$ then go to step 8, otherwise go to step 6.
Step 8. Stop.

4 Simulation Study

Consider the following affine nonlinear system

$$x_{k+1} = f(x_k) + g(x_k)u_k \tag{40}$$

where $x_k = \begin{bmatrix} x_{1k} & x_{2k} \end{bmatrix}^T$, $u_k = \begin{bmatrix} u_1(k) & u_2(k) \end{bmatrix}^T$,

$$f(x_k) = \begin{bmatrix} 0.2x_{1k}\exp(x_{2k}^2) \\ 0.3x_{2k}^3 \end{bmatrix}, \; g(x_k) = \begin{bmatrix} -0.2 & 0 \\ 0 & -0.2 \end{bmatrix}.$$

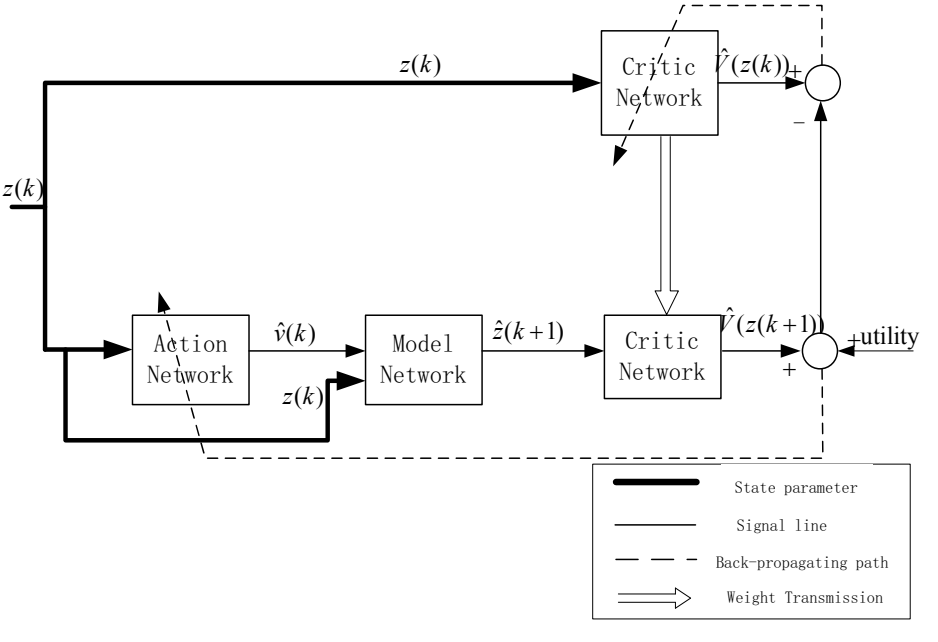


Fig. 1. The structure diagram of the algorithm

The desired trajectory is set to $\eta(k) = [\sin(k + \frac{\pi}{2}) \quad 0.5\cos(k)]^T$. We use neural network to implement the iterative ADP algorithm. We choose three-layer neural networks as the critic network, the action network and the model network with the structure 2-8-1, 2-8-2 and 6-8-2 respectively. The initial weights of action network, critic network and model network are all set to be random in $[-1, 1]$. It should be mentioned that the model network should be trained first. For the given initial state $x(0) = [1.5 \quad 1]^T$, we train the model network for 10000 steps under the learning rate $\alpha_m = 0.05$. After the training of the model network completed, the weights keep unchanged. Then the critic network and the action network are trained for 5000 steps so that the given accuracy $\varepsilon = 10^{-6}$ is reached. In the training process, the learning rate $\beta_a = \alpha_c = 0.05$. The structure diagram of the algorithm is shown in Fig. 1.

The convergence curve of the performance index function is shown in Fig.2(a). The state trajectories are given as Fig. 2(b) and Fig. 2(c). The corresponding control curves are given as Fig. 2(d).

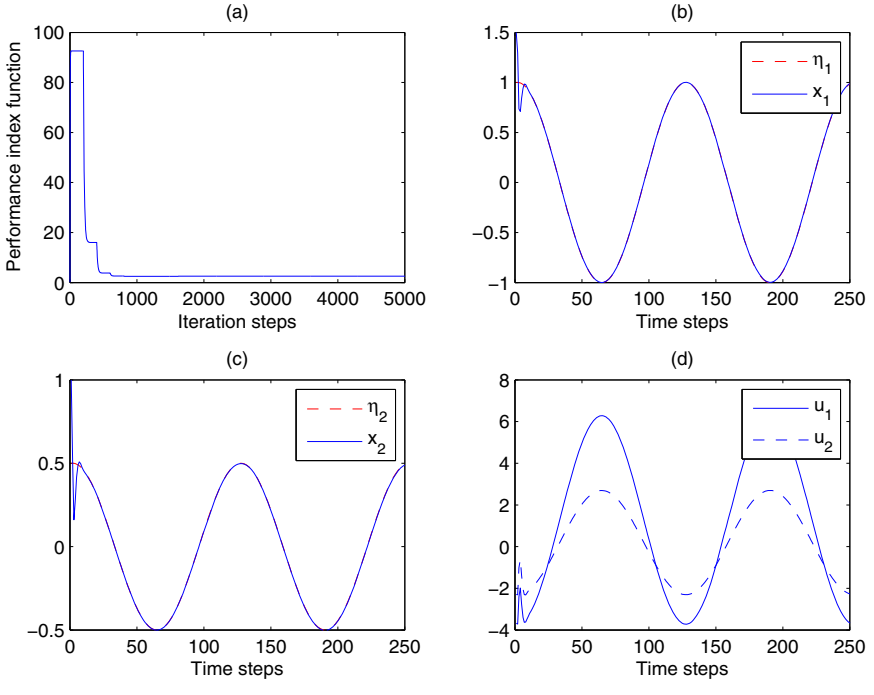


Fig. 2. The results of the algorithm

5 Conclusions

In this paper we propose an effective algorithm to solve the optimal finite horizon tracking control problem for a class of discrete-time systems. First, the tracking problems are transformed as regulation problem. Then the iterative ADP algorithm is introduced to

deal with the regulation problem with rigorous convergence analysis. Three neural networks are used as parametric structures to approximate the performance index function, compute the optimal control policy and model the unknown system respectively, i.e. the critic network, the action network and the model network. The construction of model network make the scheme can be use to control the plant with unknown dynamics. The simulation study have successfully demonstrated the upstanding performance of the proposed tracking control scheme for various discrete-time nonlinear systems.

References

1. Ha, I.J., Gilbert, E.G.: Robust tracking in nonlinear systems. IEEE Transactions Automatic Control 32, 763–771 (1987)
2. Cimen, T., Banks, S.P.: Nonlinear optimal tracking control with application to super-tankers for autopilot design. Automatica 40, 1845–1863 (2004)
3. Gao, D., Tang, G., Zhang, B.: Approximate optimal tracking control for a class of nonlinear systems with disturbances. In: Proceedings of 6th World Congress on Intelligent Control and Automation, Dalian, China, vol. 1, pp. 521–525 (2006)
4. Zhang, H.G., Wei, Q.L., Luo, Y.H.: A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear systems via the greedy HDP iteration algorithm. IEEE Transactions on System, Man, and cybernetics-Part B: Cybernetics 38, 937–942 (2008)
5. Werbos, P.J.: A menu of designs for reinforcement learning over time. In: Miller, W.T., Sutton, R.S., Werbos, P.J. (eds.) Neural Networks for Control, pp. 67–95. MIT Press, Cambridge (1991)
6. Liu, D.R., Zhang, Y., Zhang, H.: A self-learning call admission control scheme for CDMA cellular networks. IEEE Trans. Neural Networks 16, 1219–1228 (2005)
7. Wei, Q.L., Zhang, H.G., Dai, J.: Model-free multiobjective approximate dynamic programming for discrete-time nonlinear systems with general performance index functions. Neurocomputing 72, 1839–1848 (2009)
8. Zhang, H.G., Wei, Q.L., Liu, D.R.: An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games. Automatica 47(1), 207–214 (2011)
9. Zhang, H.G., Wei, Q.L., Liu, D.R.: On-Line Learning Control for Discrete Nonlinear Systems Via an Improved ADDHP Method. In: Liu, D., Fei, S., Hou, Z.-G., Zhang, H., Sun, C. (eds.) ISNN 2007. LNCS, vol. 4491, pp. 387–396. Springer, Heidelberg (2007)
10. Wei, Q.L., Liu, D.R., Zhang, H.G.: Adaptive Dynamic Programming for a Class of Nonlinear Control Systems with General Separable Performance Index. In: Sun, F., Zhang, J., Tan, Y., Cao, J., Yu, W. (eds.) ISNN 2008, Part II. LNCS, vol. 5264, pp. 128–137. Springer, Heidelberg (2008)
11. Zhang, H., Wei, Q., Liu, D.: An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games. Automatica 47, 207–214 (2011)
12. Wang, F., Jin, N., Liu, D., Wei, Q.: Adaptive dynamic programming for finite-horizon optimal control of discrete-time nonlinear systems with ϵ-error bound. IEEE Transactions on Neural Networks 22, 24–36 (2011)
13. Al-Tamimi, A., Abu-Khalaf, M., Lewis, F.L.: Adaptive critic designs for discrete-time zero-sum games with application to H_∞ control. IEEE Trans. Systems, Man, and Cybernetics-Part B: Cybernetics 37, 240–247 (2007)

Optimal Control for a Class of Unknown Nonlinear Systems via the Iterative GDHP Algorithm

Ding Wang and Derong Liu

Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
{ding.wang,derong.liu}@ia.ac.cn

Abstract. Using the neural-network-based iterative adaptive dynamic programming (ADP) algorithm, an optimal control scheme for a class of unknown discrete-time nonlinear systems with discount factor in the cost function is proposed in this paper. The optimal controller is designed with convergence analysis in terms of cost function and control law. In order to implement the algorithm via globalized dual heuristic programming (GDHP) technique, a neural network is constructed first to identify the unknown nonlinear system, and then two other neural networks are used to approximate the cost function and the control law, respectively. An example is provided to verify the effectiveness of the present approach.

Keywords: Adaptive critic designs, adaptive dynamic programming, approximate dynamic programming, intelligent control, neural networks, optimal control, reinforcement learning.

1 Introduction

The optimal control of nonlinear systems is a challenging area because it often requires solving the nonlinear Hamilton-Jacobi-Bellman (HJB) equation instead of the Riccati equation. Though dynamic programming (DP) has been an useful computational technique in solving optimal control problems for many years, it is often computationally untenable to run it to obtain the optimal solution due to the "curse of dimensionality". With strong capabilities of self-learning and adaptivity, artificial neural networks (ANN or NN) are an effective tool to implement intelligent control [2, 1, 3]. Besides, it has been used for universal function approximation in adaptive/approximate dynamic programming (ADP) algorithms, which were proposed in [4, 3] as a method to solve optimal control problems forward-in-time. There are several synonyms used for ADP including "adaptive dynamic programming", "approximate dynamic programming", "neuro-dynamic programming", "neural dynamic programming", "adaptive critic designs", and "reinforcement learning".

In recent years, ADP and the related research have gained much attention from researchers [15, 12, 11, 9, 14, 18, 17, 19, 7, 10, 8, 13, 6, 16, 20, 3, 21]. According to [3] and [6], ADP approaches were classified into several main schemes:

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 630–639, 2011.

heuristic dynamic programming (HDP), action-dependent HDP (ADHDP), dual heuristic dynamic programming (DHP), ADDHP, globalized DHP (GDHP), and ADGDHP. Al-Tamimi et al. [16] proposed a greedy HDP algorithm to solve the discrete-time HJB (DTHJB) equation for optimal control of nonlinear systems. Liu and Jin [13] developed an ε-ADP algorithm for studying finite-horizon optimal control of discrete-time nonlinear systems. Abu-Khalaf and Lewis [15], Vrabie et al. [20] studied the continuous-time optimal control problem using ADP. Though great progress has been made for ADP in optimal control field, there is still no results to solve the optimal control problem for unknown discrete-time nonlinear systems with discount factor in the cost function based on iterative ADP algorithm using GDHP technique (iterative GDHP algorithm for brief). In this paper, we will give an iterative GDHP algorithm to find the optimal controller for a class of unknown discrete-time nonlinear systems.

This paper is organized as follows. In Section 2, the DTHJB equation is introduced for nonlinear systems. In Section 3, we first design an NN identifier for the unknown system with stability proof. Then, the optimal control scheme based on the learned system dynamics and iterative ADP algorithm is developed with convergence analysis. At last, the NN implementation of the iterative algorithm is presented. In Section 4, an example is given to substantiate the theoretical results. Section 5 contains concluding remarks.

2 Problem Statement

Consider the discrete-time nonlinear system given by

$$x_{k+1} = f(x_k) + g(x_k)u(x_k), \tag{1}$$

where $x_k \in \mathbb{R}^n$ is the state and $u(x_k) \in \mathbb{R}^m$ is the control vector, $f(\cdot)$ and $g(\cdot)$ are differentiable in their argument with $f(0) = 0$ and $g(0) = 0$. Assume that $f + gu$ is Lipschitz continuous on a set Ω in $\mathbb{R}^n$ containing the origin, and that the system (1) is controllable in the sense that there exists a continuous control on Ω that asymptotically stabilizes the system.

Let x_0 be an initial state and define $\underline{u}_0^{N-1} = (u_0, u_1, \cdots, u_{N-1})$ be a control sequence with which the system (1) gives a trajectory starting from x_0: $x_1 = f(x_0) + g(x_0)u(x_0)$, $x_2 = f(x_1) + g(x_1)u(x_1)$, $\cdots$, $x_N = f(x_{N-1}) + g(x_{N-1})u(x_{N-1})$. We call the number of elements in the control sequence $\underline{u}_0^{N-1}$ the length of $\underline{u}_0^{N-1}$ and denote it as $\left|\underline{u}_0^{N-1}\right|$. Then, $\left|\underline{u}_0^{N-1}\right| = N$. The final state under the control sequence $\underline{u}_0^{N-1}$ can be denoted as $x^{(f)}\left(x_0, \underline{u}_0^{N-1}\right) = x_N$. When the control sequence starting from u_0 has infinite length, we denote it as $\underline{u}_0^{\infty} = (u_0, u_1, \cdots)$ and then the correspondingly final state can be written as $x^{(f)}(x_0, \underline{u}_0^{\infty}) = \lim_{k\to\infty} x_k$.

Let $\underline{u}_k^{\infty} = (u_k, u_{k+1}, \cdots)$ be the control sequence starting at k. It is desired to find the control sequence $\underline{u}_k^{\infty}$ which minimizes the infinite horizon cost function given by

$$J(x_k, \underline{u}_k^{\infty}) = \sum_{i=k}^{\infty} \gamma^{i-k} U(x_i, u_i), \tag{2}$$

where U is the utility function, $U(0,0)=0$, $U(x_i,u_i)\geq 0$ for $\forall x_i,u_i$, and γ is the discount factor with $0<\gamma\leq 1$. Generally speaking, the utility function can be chosen as the quadratic form as $U(x_i,u_i)=x_i^TQx_i+u_i^TRu_i$.

For optimal control problems, the designed feedback control must not only stabilize the system on Ω but also guarantee that (2) is finite, i.e., the control must be admissible.

Definition 1. *A control sequence $\underline{u}_k^\infty$ is said to be admissible for a state $x_k\in\mathbb{R}^n$ with respect to (2) on Ω if $\underline{u}_k^\infty$ is continuous on a compact set $\Omega\in\mathbb{R}^m$, $u(0)=0$, $x^{(f)}(x_k,\underline{u}_k^\infty)=0$ and $J(x_k,\underline{u}_k^\infty)$ is finite.*

Let $\mathfrak{A}_{x_k}=\left\{\underline{u}_k^\infty\colon x^{(f)}(x_k,\underline{u}_k^\infty)=0\right\}$ be the set of all infinite horizon admissible control sequences of x_k. The optimal cost function is defined as

$$J^*(x_k)=\inf_{\underline{u}_k^\infty}\left\{J(x_k,\underline{u}_k^\infty)\colon \underline{u}_k^\infty\in\mathfrak{A}_{x_k}\right\}. \tag{3}$$

According to Bellman's optimality principle, it is known that the optimal cost function $J^*(x_k)$ satisfies the DTHJB equation

$$J^*(x_k)=\min_{u_k}\left\{x_k^TQx_k+u_k^TRu_k+\gamma J^*(x_{k+1})\right\}. \tag{4}$$

The optimal control u^* is given by the gradient of the right-hand side of (4) with respect to u_k, i.e.,

$$u^*(x_k)=-\frac{\gamma}{2}R^{-1}g^T(x_k)\frac{\partial J^*(x_{k+1})}{\partial x_{k+1}}. \tag{5}$$

By substituting (5) into (4), the DTHJB equation becomes

$$J^*(x_k)=x_k^TQx_k+\frac{\gamma^2}{4}\left(\frac{\partial J^*(x_{k+1})}{\partial x_{k+1}}\right)^T g(x_k)R^{-1}g^T(x_k)\frac{\partial J^*(x_{k+1})}{\partial x_{k+1}}+\gamma J^*(x_{k+1}),$$

where $J^*(x_k)$ is the optimal cost function corresponding to the optimal control law $u^*(x_k)$. Since the above DTHJB equation is difficult to solve, we will present a novel algorithm to approximate the cost function iteratively in next section.

3 Adaptive Dynamic Programming for Neuro-Optimal Control of the Unknown Nonlinear Systems

3.1 NN System Identification of the Unknown Nonlinear Systems

In this paper, we consider a three-layer feedforward NN as the function approximation structure. Let the number of hidden layer neurons be denoted by l, the ideal weight matrix between the input layer and hidden layer be denoted by ν_m^*, and the ideal weight matrix between the hidden layer and output layer be denoted by ω_m^*. According to the universal approximation property of NN, the

system dynamics (1) has a NN representation on a compact set S, which can be written as

$$x_{k+1} = \omega_m^{*T}\sigma\left(\nu_m^{*T} z_k\right) + \theta_k. \tag{6}$$

In (6), $z_k = [x_k^T \ u_k^T]^T$ is the NN input, θ_k is the bounded NN functional approximation error according to the universal approximation property, and $[\sigma(\bar{z})]_i = (e^{\bar{z}_i} - e^{-\bar{z}_i})/(e^{\bar{z}_i} + e^{-\bar{z}_i})$, $i = 1, 2, \cdots, l$, are the activation functions selected in this work, where $\bar{z}_k = \nu_m^{*T} z_k$, $\bar{z}_k \in \mathbb{R}^l$. Additionally, the NN activation functions are bounded such that $\|\sigma(\bar{z}_k)\| \leq \sigma_M$ for a constant σ_M.

In the system identification process, we keep the weight matrix between the input layer and the hidden layer as constant while only tune the weight matrix between the hidden layer and the output layer. So, we define the NN system identification scheme as

$$\hat{x}_{k+1} = \omega_m^T(k)\sigma(\bar{z}_k), \tag{7}$$

where $\hat{x}_k$ is the estimated system state vector, and $\omega_m(k)$ is the estimation of the constant ideal weight matrix ω_m^*.

Denote $\tilde{x}_k = \hat{x}_k - x_k$ as the system identification error. Combining (6) and (7), we can obtain the identification error dynamics as $\tilde{x}_{k+1} = \tilde{\omega}_m^T(k)\sigma(\bar{z}_k) - \theta_k$, where $\tilde{\omega}_m(k) = \omega_m(k) - \omega_m^*$. Let $\psi_k = \tilde{\omega}_m^T(k)\sigma(\bar{z}_k)$, then, we have $\tilde{x}_{k+1} = \psi_k - \theta_k$. Using the gradient-based adaptation rule, the weights are updated as

$$\omega_m(k+1) = \omega_m(k) - \alpha_m\left[\frac{\partial E_{k+1}}{\partial \omega_m(k)}\right] = \omega_m(k) - \alpha_m\sigma(\bar{z}_k)\tilde{x}_{k+1}^T, \tag{8}$$

where $E_{k+1} = 0.5\tilde{x}_{k+1}^T\tilde{x}_{k+1}$ is the performance measure to be minimized and $\alpha_m > 0$ is the NN learning rate.

Assumption 1. *The NN approximation error term θ_k is assumed to be upper bounded by a function of the state estimation error $\tilde{x}_k$, i.e., $\theta_k^T\theta_k \leq \delta_M\tilde{x}_k^T\tilde{x}_k$, where δ_M is a constant value.*

Theorem 1. *Let the identification scheme (7) be used to identify the nonlinear system (1), and let the parameter update law given in (8) be used for tuning the NN weights. Then, the state estimation error dynamics $\tilde{x}_k$ is asymptotically stable while the parameter estimation error $\tilde{\omega}_m(k)$ is bounded.*

Proof. Basing on the Lyapunov theory, this theorem can be proved by choosing the Lyapunov function candidate as $L_k = \tilde{x}_k^T\tilde{x}_k + \mathrm{tr}\{\tilde{\omega}_m^T(k)\tilde{\omega}_m(k)\}/\alpha_m$. The details is omitted here due to the length constraints.

According to Theorem 1, after a sufficiently long learning session, the NN system identification error converges to zero, i.e., we have $f(x_k) + \hat{g}(x_k)u(x_k) = \omega_m^T(k)\sigma(\bar{z}_k)$, where $\hat{g}(x_k)$ denotes the estimated value of the control coefficient matrix $g(x_k)$. Taking partial derivative of it with respect to u_k yields

$$\hat{g}(x_k) = \frac{\partial\left(\omega_m^T(k)\sigma(\bar{z}_k)\right)}{\partial u_k} = \omega_m^T(k)\frac{\partial\sigma(\bar{z}_k)}{\partial\bar{z}_k}\nu_m^{*T}\left[\begin{array}{c}0_{n\times m}\\ \hdashline I_m\end{array}\right], \tag{9}$$

where $0_{n\times m}$ is an $n \times m$ zero matrix and I_m is an $m \times m$ identity matrix. This result will be used in the derivation and implementation of the iterative ADP algorithm for the optimal control of unknown discrete-time nonlinear systems.

3.2 Derivation of the Iterative ADP Algorithm

Now, we present the iterative ADP algorithm. First, we start with the initial cost function $V_0(\cdot) = 0$, and then solve for the law of single control vector

$$v_0(x_k) = \arg\min_{u_k} \left\{ x_k^T Q x_k + u_k^T R u_k + \gamma V_0(x_{k+1}) \right\}. \tag{10}$$

Once the control law $v_0(x_k)$ is determined, we update the cost function as

$$\begin{aligned} V_1(x_k) &= \min_{u_k} \left\{ x_k^T Q x_k + u_k^T R u_k + \gamma V_0(x_{k+1}) \right\} \\ &= x_k^T Q x_k + v_0^T(x_k) R v_0(x_k). \end{aligned} \tag{11}$$

Then, for $i = 1, 2, \cdots$, the iterative ADP algorithm can be used to implement the iteration between the control law

$$\begin{aligned} v_i(x_k) &= \arg\min_{u_k} \left\{ x_k^T Q x_k + u_k^T R u_k + \gamma V_i(x_{k+1}) \right\} \\ &= -\frac{\gamma}{2} R^{-1} \hat{g}^T(x_k) \frac{\partial V_i(x_{k+1})}{\partial x_{k+1}} \end{aligned} \tag{12}$$

and the cost function

$$\begin{aligned} V_{i+1}(x_k) &= \min_{u_k} \left\{ x_k^T Q x_k + u_k^T R u_k + \gamma V_i(x_{k+1}) \right\} \\ &= x_k^T Q x_k + v_i^T(x_k) R v_i(x_k) + \gamma V_i(f(x_k) + \hat{g}(x_k) v_i(x_k)). \end{aligned} \tag{13}$$

In the following part, we will present a proof of convergence of the iteration between (12) and (13) with $V_i \to J^*$ and $v_i \to u^*$ as $i \to \infty$.

3.3 Convergence Analysis of the Iterative ADP Algorithm

Lemma 1. *Let $\{\mu_i\}$ be an arbitrary sequence of control laws and $\{v_i\}$ be the control law sequence described in (12). Define V_i as in (13) and Λ_i as*

$$\Lambda_{i+1}(x_k) = x_k^T Q x_k + \mu_i^T(x_k) R \mu_i(x_k) + \gamma \Lambda_i(f(x_k) + \hat{g}(x_k)\mu_i(x_k)). \tag{14}$$

If $V_0(x_k) = \Lambda_0(x_k) = 0$, then $V_i(x_k) \le \Lambda_i(x_k)$, $\forall i$.

Proof. It can easily be derived noticing that V_{i+1} is the result of minimizing the right-hand side of (13) with respect to the control input u_k, while Λ_{i+1} is a result of arbitrary control input.

Lemma 2. *Let the cost function sequence $\{V_i\}$ be defined as in (13). If the system is controllable, there is an upper bound Y such that $0 \le V_i(x_k) \le Y$, $\forall i$.*

Based on Lemmas 1 and 2, we now present our main theorems.

Theorem 2. *Define the cost function sequence $\{V_i\}$ as in (13) with $V_0(\cdot) = 0$, and the control law sequence $\{v_i\}$ as in (12). Then, $\{V_i\}$ is a monotonically nondecreasing sequence satisfying $V_{i+1} \ge V_i$, $\forall i$.*

We have acquired the conclusion that the cost function sequence $\{V_i\}$ is a monotonically nondecreasing sequence with an upper bound, and therefore, its limit exists. Now, we can derive the following theorem.

Theorem 3. *For any state vector x_k, define $\lim_{i\to\infty} V_i(x_k) = V_\infty(x_k)$ as the limit of the cost function sequence $\{V_i(x_k)\}$. Then, the following equation holds:*

$$V_\infty(x_k) = \min_{u_k} \left\{ x_k^T Q x_k + u_k^T R u_k + \gamma V_\infty(x_{k+1}) \right\}.$$

Next, we will prove that the cost function sequence $\{V_i(x_k)\}$ converges to the optimal cost function $J^*(x_k)$ as $i \to \infty$.

Theorem 4. *Define the cost function sequence $\{V_i\}$ as in (13) with $V_0(\cdot) = 0$. If the system state x_k is controllable, then J^* is the limit of the cost function sequence $\{V_i\}$, i.e., $\lim_{i\to\infty} V_i(x_k) = J^*(x_k)$.*

For space reasons, we will present the details of the proof of Lemma 2 and Theorems 2–4 in a future paper.

From Theorems 2–4, we can obtain that the cost function sequence $\{V_i(x_k)\}$ converges to the optimal cost function $J^*(x_k)$ of the DTHJB equation. Then, according to (5) and (12), we can conclude that the control law sequence also converges to the optimal control law (5), i.e., $v_i \to u^*$ as $i \to \infty$.

3.4 NN Implementation of the Iterative ADP Algorithm Using GDHP Technique

Now, we implement the iterative GDHP algorithm in (12) and (13). In the iterative GDHP algorithm, there are three NNs, which are model network, critic network and action network. All the networks are chosen as three-layer feedforward NNs. The inputs of the critic network and action network are x_k, while the inputs of the model network are x_k and $\hat{v}_i(x_k)$. The training of the model network is completed after the system identification process and its weights are kept unchanged. The learned NN system model will be used in the process of training the critic network and action network.

The critic network is used to approximate both $V_i(x_k)$ and its derivative $\partial V_i(x_k)/\partial x_k$, which is named costate function and denoted as $\lambda_i(x_k)$. The output of the critic network is denoted as

$$\begin{bmatrix} \hat{V}_i(x_k) \\ \hat{\lambda}_i(x_k) \end{bmatrix} = \begin{bmatrix} \omega_{ci}^{1T} \\ \omega_{ci}^{2T} \end{bmatrix} \sigma\left(\nu_{ci}^T x_k\right) = \omega_{ci}^T \sigma\left(\nu_{ci}^T x_k\right), \tag{15}$$

where $\omega_{ci} = \begin{bmatrix} \omega_{ci}^1 & \omega_{ci}^2 \end{bmatrix}$. So, $\hat{V}_i(x_k) = \omega_{ci}^{1T} \sigma\left(\nu_{ci}^T x_k\right)$ and $\hat{\lambda}_i(x_k) = \omega_{ci}^{2T} \sigma\left(\nu_{ci}^T x_k\right)$. The target function can be written as

$$V_{i+1}(x_k) = x_k^T Q x_k + v_i^T(x_k) R v_i(x_k) + \gamma \hat{V}_i(\hat{x}_{k+1}) \tag{16}$$

and

$$\lambda_{i+1}(x_k) = \frac{\partial\left(x_k^T Q x_k + v_i^T(x_k) R v_i(x_k)\right)}{\partial x_k} + \gamma \frac{\partial \hat{V}_i(\hat{x}_{k+1})}{\partial x_k}$$
$$= 2Qx_k + 2\left(\frac{\partial v_i(x_k)}{\partial x_k}\right)^T R v_i(x_k)$$
$$+ \gamma\left(\frac{\partial \hat{x}_{k+1}}{\partial x_k} + \frac{\partial \hat{x}_{k+1}}{\partial \hat{v}_i(x_k)} \frac{\partial \hat{v}_i(x_k)}{\partial x_k}\right)^T \hat{\lambda}_i(\hat{x}_{k+1}). \tag{17}$$

Then we define the error function for training the critic network as $e_{cik}^1 = \hat{V}_{i+1}(x_k) - V_{i+1}(x_k)$ and $e_{cik}^2 = \hat{\lambda}_{i+1}(x_k) - \lambda_{i+1}(x_k)$. The objective function to be minimized in the critic network training is $E_{cik} = (1-\beta)E_{cik}^1 + \beta E_{cik}^2$, where $E_{cik}^1 = 0.5 e_{cik}^{1T} e_{cik}^1$ and $E_{cik}^2 = 0.5 e_{cik}^{2T} e_{cik}^2$. The weight updating rule for training the critic network is also gradient-based adaptation given by

$$\omega_{ci}(j+1) = \omega_{ci}(j) - \alpha_c \left[(1-\beta)\frac{\partial E_{cik}^1}{\partial \omega_{ci}(j)} + \beta \frac{\partial E_{cik}^2}{\partial \omega_{ci}(j)}\right], \tag{18}$$

$$\nu_{ci}(j+1) = \nu_{ci}(j) - \alpha_c \left[(1-\beta)\frac{\partial E_{cik}^1}{\partial \nu_{ci}(j)} + \beta \frac{\partial E_{cik}^2}{\partial \nu_{ci}(j)}\right], \tag{19}$$

where $\alpha_c > 0$ is the learning rate of the critic network, j is the inner-loop iterative step for updating the weight parameters, and $0 \le \beta \le 1$ is a parameter that adjusts how HDP and DHP are combined in GDHP. For $\beta = 0$, the training of the critic network reduces to a pure HDP, while $\beta = 1$ does the same for DHP.

The output of the action network is expressed as $\hat{v}_i(x_k) = \omega_{ai}^T \sigma\left(\nu_{ai}^T x_k\right)$ and its training target is

$$v_i(x_k) = -\frac{\gamma}{2} R^{-1} \hat{g}^T(x_k) \frac{\partial \hat{V}_i(\hat{x}_{k+1})}{\partial \hat{x}_{k+1}}. \tag{20}$$

The error function of the action network can be defined as $e_{aik} = \hat{v}_i(x_k) - v_i(x_k)$. The weights of the action network are updated to minimize the error measure $E_{aik} = 0.5 e_{aik}^T e_{aik}$. Similarly, the weight updating algorithm is

$$\omega_{ai}(j+1) = \omega_{ai}(j) - \alpha_a \left[\frac{\partial E_{aik}}{\partial \omega_{ai}(j)}\right], \tag{21}$$

$$\nu_{ai}(j+1) = \nu_{ai}(j) - \alpha_a \left[\frac{\partial E_{aik}}{\partial \nu_{ai}(j)}\right], \tag{22}$$

where $\alpha_a > 0$ is the learning rate of the action network, and j is the inner-loop iterative step for updating the weight parameters.

Remark 1. *According to Theorem 4, $V_i(x_k) \to J^*(x_k)$ as $i \to \infty$. Since $\lambda_i(x_k) = \partial V_i(x_k)/\partial x_k$, we can conclude that the costate function sequence $\{\lambda_i(x_k)\}$ is also convergent with $\lambda_i(x_k) \to \lambda^*(x_k)$ as $i \to \infty$.*

4 Simulation Study

Consider the nonlinear system derived from [5]:

$$x_{k+1} = \begin{bmatrix} \dfrac{x_{2k}^3 + x_{2k}}{1 + 2x_{2k}^2} \\ \dfrac{x_{1k} + x_{2k}}{1 + x_{1k}^2} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(x_k),$$

where $x_k = [x_{1k}\ x_{2k}]^T \in \mathbb{R}^2$ and $u_k \in \mathbb{R}$ are the state and control variables, respectively. The cost function is chosen as $U(x_k, u_k) = x_k^T x_k + u_k^T u_k$.

We choose three-layer feedforward NNs as model network, critic network and action network with the structures 3–8–2, 2–8–3, 2–8–1, respectively. The initial weights of them are all set to be random in $[-0.1, 0.1]$. Let the discount factor $\gamma = 1$ and the adjusting parameter $\beta = 0.5$, we train the critic network and action network for 38 training cycles with each cycle of 2000 steps. In the training process, the learning rate $\alpha_c = \alpha_a = 0.05$. The convergence process of the cost function and its derivative of the iterative GDHP algorithm at time instant $k = 0$ are shown in Fig. 1. We can see that the iterative cost function sequence does converge to the optimal cost function quite rapidly, which indicates the effectiveness of the iterative GDHP algorithm. Besides, the costate function sequence is also convergent as Remark 1 stated.

Then, for the given initial state $x_{10} = 1$ and $x_{20} = 1$, we apply the optimal control law derived by the iterative GDHP algorithm to the controlled system for 15 time steps, and obtain the simulation results are shown in Fig. 2. We can see that the controller designed by the iterative GDHP algorithm has excellent performance. Moreover, the most important property that the iterative GDHP algorithm superior to the iterative HDP and DHP algorithms is that the former can show us the convergence process of the cost function and costate function sequence simultaneously.

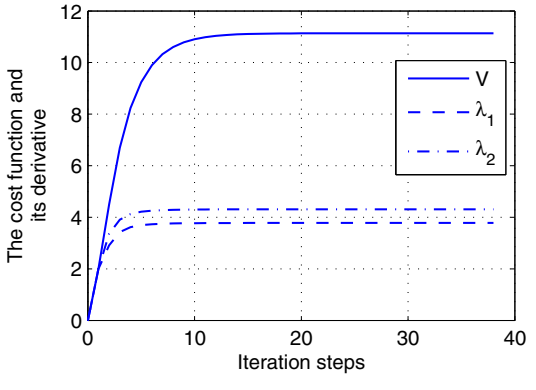


Fig. 1. The convergence process of the cost function and costate function sequence

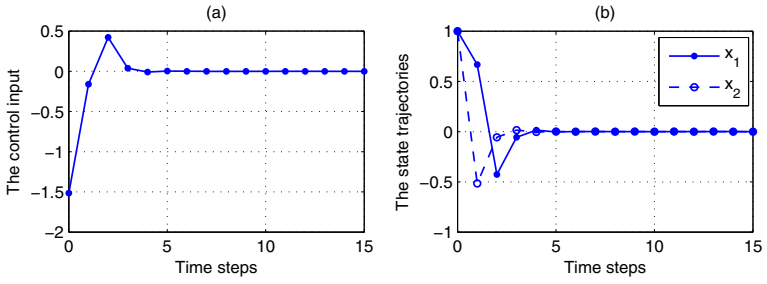


Fig. 2. The simulation results. (a) The control input u. (b) The corresponding state trajectories x_1 and x_2.

5 Conclusion

In this paper, a novel NN-based approach is proposed to design the near optimal controller for a class of unknown discrete-time nonlinear systems with discount factor in the cost function. The iterative GDHP algorithm is introduced to solve the cost function of the DTHJB equation with convergence analysis. Three NNs are used to implement the algorithm. The simulation example demonstrated the validity of the derived optimal control strategy.

Acknowledgments. This work was supported in part by the NSFC under grants 60904037, 60921061, and 61034002, and by Beijing Natural Science Foundation under grant 4102061.

References

1. Jagannathan, S.: Neural Network Control of Nonlinear Discrete-time Systems. CRC Press, Boca Raton (2006)
2. Yu, W.: Recent Advances in Intelligent Control Systems. Springer, London (2009)
3. Werbos, P.J.: Approximate Dynamic Programming for Real-time Control and Neural Modeling. In: White, D.A., Sofge, D.A. (eds.) Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approach, ch. 13. Van Nostrand Reinhold, New York (1992)
4. Werbos, P.J.: Advanced Forecasting Methods for Global Crisis Warning and Models of Intelligence. General Systems Yearbook 22, 25–38 (1977)
5. Canelon, J.I., Shieh, L.S., Karayiannis, N.B.: A New Approach for Neural Control of Nonlinear Discrete Dynamic Systems. Information Sciences 174(3-4), 177–196 (2005)
6. Prokhorov, D.V., Wunsch, D.C.: Adaptive Critic Designs. IEEE Transactions on Neural Networks 8(5), 997–1007 (1997)
7. Si, J., Wang, Y.T.: On-line Learning Control by Association and Reinforcement. IEEE Transactions on Neural Networks 12(2), 264–276 (2001)
8. Murray, J.J., Cox, C.J., Lendaris, G.G., Saeks, R.: Adaptive Dynamic Programming. IEEE Transactions on Systems, Man, Cybernatics–Part C: Applications and Reviews 32(2), 140–153 (2002)

9. Wang, F.Y., Zhang, H., Liu, D.: Adaptive Dynamic Programming: an Introduction. IEEE Computational Intelligence Magazine 4(2), 39–47 (2009)
10. Lewis, F.L., Vrabie, D.: Reinforcement Learning and Adaptive Dynamic Programming for Feedback Control. IEEE Circuits and Systems Magazine 9(3), 32–50 (2009)
11. Liu, D., Xiong, X., Zhang, Y.: Action-dependent Adaptive Critic Designs. In: Proceedings of the International Joint Conference on Neural Networks, pp. 990–995 (2001)
12. Liu, D., Zhang, Y., Zhang, H.: A Self-learning Call Admission Control Scheme for CDMA Cellular Networks. IEEE Transactions on Neural Networks 16(5), 1219–1228 (2005)
13. Liu, D., Jin, N.: ϵ-adaptive Dynamic Programming for Discrete-time Systems. In: Proceedings of the International Joint Conference on Neural Networks, pp. 1417–1424 (2008)
14. Yen, G.G., Delima, P.G.: Improving the Performance of Globalized Dual Heuristic Programming for Fault Tolerant Control Through an Online Learning Supervisor. IEEE Transactions an Automation Science and Engineering 2(2), 121–131 (2005)
15. Abu-Khalaf, M., Lewis, F.L.: Nearly Optimal Control Laws for Nonlinear Systems with Saturating Actuators Using a Neural Network HJB Approach. Automatica 41(5), 779–791 (2005)
16. Al-Tamimi, A., Lewis, F.L., Abu-Khalaf, M.: Discrete-time Nonlinear HJB Solution Using Approximate Dynamic Programming: Convergence Proof. IEEE Transactions on Systems, Man, Cybernatics–Part B: Cybernatics 38(4), 943–949 (2008)
17. Luo, Y., Zhang, H.: Approximate Optimal Control for a Class of Nonlinear Discrete-time Systems with Saturating Actuators. Progress in Natural Science 18(8), 1023–1029 (2008)
18. Wei, Q., Zhang, H., Liu, D., Zhao, Y.: An Optimal Control Scheme for a Class of Discrete-time Nonlinear Systems with Time Delays Using Adaptive Dynamic Programming. Acta Automatica Sinica 36(1), 121–129 (2010)
19. Dierks, T., Thumati, B.T., Jagannathan, S.: Optimal Control of Unknown Affine Nonlinear Discrete-time Systems Using Offline-trained Neural Networks with Proof of Convergence. Neural Networks 22(5-6), 851–860 (2009)
20. Vrabie, D., Pastravanu, O., Abu-Khalaf, M., Lewis, F.L.: Adaptive Optimal Control for Continuous-time Linear Systems Based on Policy Iteration. Automatica 45(2), 477–484 (2009)
21. Sun, Z., Chen, X., He, Z.: Adaptive Critic Designs for Energy Minimization of Portable Video Communication Devices. IEEE Transactions on Circuits and Systems for Video Technology 20(1), 27–37 (2010)

Author Index